Doctoral theses at NTNU, 2023:59

Ashkan Moradi

Distributed Learning and Estimation with Enhanced Privacy and Security

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Ashkan Moradi

Distributed Learning and Estimation with Enhanced Privacy and Security

Thesis for the Degree of Philosophiae Doctor

Trondheim, March 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

ISBN 978-82-326-6586-0 (printed ver.)
ISBN 978-82-326-5501-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2023:59

Printed by NTNU Grafisk senter

Abstract

This thesis focuses on threat analysis and management in distributed learning scenarios intending to develop algorithms to mitigate the impact of adversaries in the network. The thesis begins with a threat analysis that includes investigating possible adversaries and their attack strategies. It examines the worst-case scenario of an adversarial attack to identify critical agents/links or potential loopholes. Further, the thesis investigates threat management and security algorithms to provide resilience against malicious behaviors in the network, as well as strategies to protect the privacy of network agents.

In the scope of the threat analysis, we mainly focus on distributed learning algorithms that are essentially vulnerable to adversarial attacks. By investigating the network dynamics from an adversarial perspective, we design the optimal coordinated data falsification attack that maximizes the network steady-state mean squared error (MSE). The adversary simultaneously optimizes the subset of Byzantine agents and their attack sequences to maximize the network MSE. The Byzantine agent is a legitimate network agent that injects false data into the system to disrupt the overall performance of the network. Moreover, we propose a distributed filtering algorithm that provides robustness to Byzantine attacks. The proposed Byzantine-resilient consensus-based distributed filter (BR-CDF) also offers communication efficiency by allowing agents to exchange only a fraction of their information at each instant. In addition, we redesign the optimal attack strategy by solving an optimization problem where Byzantine agents cooperate on designing their attack covariances or the sequence of the information fractions they share.

Agents in distributed learning scenarios improve local estimates by exchanging information with neighbors. These local interactions, however, expose private information to adversaries. As an approach to threat management, we propose a privacy-preserving distributed Kalman filter (PP-DKF) that protects local information from being inferred by adversaries. The proposed PP-DKF protects local information by randomly decomposing the state estimates into public and private substates and only sharing a perturbed version of the public substate with neighbors. Moreover, we derive privacy bounds for all agents in the presence of an external eavesdropper (EE) and an honest-but-curious (HBC) adversary. Additionally, we propose partial sharing and privacy-preserving distributed learning (PPDL) algorithms that offer communication efficiency while preserving privacy. The proposed PPDL algorithms utilize noise injection and state decomposition techniques to induce privacy and provide communication efficiency by only sharing a fraction of information at any given instant.

The final part of the thesis aims to further enhance the robustness of the distributed filtering algorithm to coordinated data falsification attacks. To this end, we model a distributed Kalman filtering process as a distributed optimization problem with consensus constraints. We derive a suboptimal solution to the filtering algorithm that provides robustness to Byzantine attacks using a total variation (TV) penalty term for the objective function. The proposed Byzantine-resilient distributed Kalman filter (BR-DKF) restricts the impact of Byzantine perturbations completely, and only the number of Byzantine agents influences the filtering error bound.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for the fulfillment of requirements for the degree of Doctor of Philosophy.

The doctoral work started in September 2018 at the Department of Electronic Systems, NTNU, Trondheim, Norway. The work has been supervised by Professor Stefan Werner and co-supervised by Associate Professor Naveen Kumar Dasanadoddi Venkategowda.

The members of the assessment committee are: Professor Subhrakanti Dey, Uppsala University, Sweden; Assistant Professor Geethu Joseph, Delft University of Technology, The Netherlands; Professor Pierluigi Salvo Rossi, Norwegian University of Science and Technology, Norway.

The doctoral work was funded by the Research Council of Norway.

Acknowledgments

First and foremost I am extremely grateful to my family for their unwavering support and love. They have been my rock through life's challenges and celebrations and I am truly blessed to have them in my life. Their support and encouragement have been a constant source of strength and inspiration in my Ph.D. journey. This journey would not be complete without such wonderful individuals as my loved ones, and I will always cherish their presence in my life.

I would like to extend my sincerest appreciation to my friends who have supported me throughout my Ph.D. journey. I am certain that you are aware of who I am speaking to. Yes you, thank you for always being there to listen to my complaints about my Ph.D. situation. Thanks for all the pep talks that made me stronger and kept me sane during this journey. I am truly grateful to have you amazing individuals as my friends.

I would like to acknowledge and give my warmest thanks to my main supervisor Professor Stefan Werner who made this journey possible. His guidance and advice carried me through all stages of my Ph.D. and helped me refine my scientific skills. I would also like to thank my co-supervisor Associate Professor Naveen Venkategowda for his valuable suggestions and mentor-ship during my Ph.D. journey. I would also like to gratefully acknowledge and appreciate the help and support of my co-authors Dr. Pouria Talebi and Dr. Vinay Gogineni for their valuable scientific collaborations on my academic works.

I would also like to thank the members of the assessment committee for this thesis, Professor Subhrakanti Dey, Assistant Professor Geethu Joseph, and Professor Pierluigi Salvo Rossi. Their insightful feedback and expertise helped immensely in shaping and enhancing the final manuscript.

Last but not least, I would like to thank the Department of Electronic Systems, especially the Signal Processing Group members, who made this experience more enjoyable by providing such a friendly working atmosphere.

Contents

List of Figures

Abbreviations and Symbols

Abbreviations

ACF	Average consensus filter
BCD	Block-coordinate descent
BR-CDF	Byzantine-resilient consensus-based distributed filter
BR-DKF	Byzantine robust distributed Kalman filter
CDF	Consensus-based distributed Kalman filter
CKF	Centralized Kalman filter
CPS	Cyber physical system
D-LMS	Distributed least mean square
DKF	Distributed Kalman filter
DL	Distributed learning
DNI-PPDL	Decomposition and noise injection-based partial-sharing private distributed learning
DoS	Denial-of-service
DP	Differential privacy
EE	External eavesdropper

FL	Federated learning
HBC	Honest-but-curious
IoT	Internet of Things
KL	Kullback-Leibler
LMS	Least mean square
ML	Maximum likelihood
MSE	Mean squared error
NI-PPDL	Noise injection-based partial-sharing private distributed learning
NIP-DKF	Noise-injection based privacy-preserving distributed Kalman filter
NMSE	Network-wide mean squared error
PP-DKF	Privacy-preserving distributed Kalman filter
PPDL	Partial-sharing private distributed learning
SDP	Semidefinite programming
TV	Total variation

Symbols

$(\cdot)^{\dagger}$	Moore–Penrose pseudoinverse operator
$(\cdot)^{\mathrm{T}}$	Transpose operator
$(\cdot)^{-1}$	Inverse operator
$\mathbf{A}$	State transition matrix
$\boldsymbol{\alpha}_{i,n}(k)$	Public substate at agent i, time instant n, and consensus iteration k
$\mathcal{B}$	Set of Byzantine agents
$\boldsymbol{\beta}_{i,n}(k)$	Private substate at agent i, time instant n, and consensus iteration k
$\boldsymbol{\nu}_i(k)$	Average consensus perturbation noise at agent i and consensus iteration k

$\boldsymbol{\omega}_i(k)$	Average consensus perturbation noise at agent i and consensus iteration k
$\boldsymbol{\delta}_{i,n}$	Byzantine agent perturbation noise at agent i and time instant n
$\mathbb{E}\{\cdot\}$	Statistical expectation operator
$\mathbf{E}$	Adjacency matrix
$\mathbf{e}_{i,n}$	Estimation error at agent i and time instant n
ϵ	Consensus gain
$\mathbf{H}_i$	Observation matrix at agent i
$\mathbf{I}_m$	$m \times m$ identity matrix
$\mathcal{I}_{\text{EE}}$	Information set at external eavesdropper
$\mathcal{I}_{\text{HBC}}$	Information set at honest-but-curious agent
λ_{tv}	Total variation penalty parameter
$\mathbf{0}_m$	$m \times m$ zero matrix
$\mathbf{1}_m$	One vector with length m
$\boldsymbol{\Gamma}_{i,n}$	Intermediate error covariance at agent i and time instant n
$\mathbf{K}_{i,n}$	Kalman gain at agent i and time instant n
$\mathbf{r}_{i,n}$	Intermediate state estimate at agent i and time instant n
$\mathcal{E}$	Set of edges
$\mathcal{E}_{i,n}(k)$	Privacy measure at agent i, time instant n, and after k consensus iterations
$\mathcal{G}$	Graph
$\mathcal{N}$	Set of agents
$\mathcal{N}_i$	Set of neighboring agents at agent i
$\mathsf{Blockdiag}(\cdot)$	Block diagonal matrix containing the argument matrices on the main diagonal
$\mathsf{diag}(\cdot)$	Diagonal matrix containing the argument vector on the main diagonal

$\mathbb{N}$	Set of natural numbers
$\odot$	Matrix Hadamard product
$\otimes$	Matrix Kronecker product
$\mathbf{P}_{i,n}$	Estimation error covariance at agent i and time instant n
ϕ	Perturbation noise constant
$\mathbf{Q}$	State noise covariance matrix
$\mathbb{R}$	Set of real numbers
$\mathbf{R}_i$	Observation noise covariance matrix at agent i
$\mathbf{S}_{i,n}(k)$	The partial-sharing selection matrix at agent i, time instant n, and consensus iteration k
σ^2	Perturbation noise variance
$\mathbf{\Sigma}_i$	Perturbation covariance at Byzantine agent i
$\mathrm{sign}(\cdot)$	Sign function
$\mathrm{tr}(\cdot)$	Trace operator
$\mathrm{vec}(\cdot)$	A column vector consisting of elements of the argument matrix
$\mathrm{vec}^{-1}(\cdot)$	The inverse operation of $\mathrm{vec}(\cdot)$ operator
$\mathbf{v}_{i,n}$	Observation noise at agent i and time instant n
$\mathbf{w}_n$	State noise at time instant n
$\mathbf{x}_n$	Actual state of the system at time instant n
$\hat{\mathbf{x}}_{i,n}$	State estimate at agent i and time instant n
$\mathbf{y}_{i,n}$	Agent observation at agent i and time instant n
N	Number of agents in the network

Chapter 1

Introduction

The pervasiveness of the Internet of Things (IoT), which connects numerous buildings, sensors, appliances, and vehicles, is driving the evolution of more intelligent and greener cities, environmental monitoring, and physical health care systems. However, the high connectivity of smart objects and their severe energy and processing limitations cause many security challenges. In a distributed setting, an adversary may hijack and gain complete control over a subset of devices and maliciously alter the reported measurements or inject wrong information into the system. Thus, a vital requirement in a secure IoT network is an intelligent and autonomous infrastructure resilient to such malicious disruptions. Preserving the integrity and trustworthiness of data and their associated analytics calls for new approaches that can securely gather and process the collected data to ensure a reliable inference even in the presence of adversaries.

Sensor devices with networking capabilities have sparked considerable interest in distributed learning, filtering, and estimation algorithms in multi-agent systems. In this study, we focus mainly on distributed learning algorithms with high accuracy, computational efficiency, and ability to model various real-world physical systems. This broad applicability has made distributed learning and estimation techniques a prominent fixture in many signal processing applications [1–6]. Furthermore, with the emergence of consensus- and diffusion-based algorithms, distributed filtering became widely used and significantly impacted the development of dynamic state estimation methods [6–10]. In general, distributed learning algorithms rely on local collaboration among agents. As a result of these algorithms, learning performance is improved by aggregating local data with observation and state information from neighbors [3, 7].

The information exchange among network agents facilitates collaboration but raises privacy concerns because of the exposure of private information to potential adversaries. In distributed algorithms, even a single intruder or malicious agent can jeopardize the trustworthiness of the system, further accentuating concerns about privacy and security [11, 12]. In particular, an adversary may manipulate the data stream of an agent or hijack and control a subset of agents to compromise the overall system performance. Thus, providing a robust procedure to maintain the desired goal of the system even in the presence of adversaries remains a challenge. Developing an algorithm that preserves privacy or performs robustly in the presence of adversaries requires a thorough understanding of the adversarial model and capabilities of the attacker. We can divide network adversaries into three types based on their capabilities and the information they can access. The first is the external eavesdropper (EE), who gets access to all the information exchanged between agents to glean private information. The second type is the honest-but-curious (HBC) adversary, which is a legitimate node of the network that contributes to the overall distributed operations but, at the same time, passively seeks to infer private information from messages shared by its immediate neighbors. Last but not least is the Byzantine adversary, a legitimate network agent that injects false data into systems to impair the overall performance of the network.

Aside from the attackers and their accessible information, the attack strategies also significantly impact the privacy-preserving and secure approaches developed. Attacks on multi-agent networks can be classified as active or passive, with passive attacks occurring when an eavesdropper intercepts a link between agents to obtain information [13]. On the other hand, active attacks include denial-of-service attacks (DoS) and integrity attacks. During DoS attacks, agents cannot exchange information due to link blockages [14], while integrity attacks are caused by external adversaries or malicious agents injecting false information into the network [15]. In this thesis, we also examine distributed algorithms from the perspective of an attacker to investigate security threats and identify their potential loopholes under critical circumstances. Thus, the optimal attack strategy, from the perspective of an attacker, is designed to improve attack protection methods by experiencing the worst-case scenario of a cyber-attack [16–18].

After analyzing potential adversaries and their attacking strategies, the threat management process begins to act by developing mitigation approaches to reduce the negative impact of adversaries on the network performance. One method is to detect the potential adversaries and counteract their actions by implementing correction measures [19–21]. However, studies in [22–24] have shown that relying on attack detection to limit the impact of adversaries has restricted utility in the presence of undetected attacks. Hence, there is a need for a robust algorithm that can operate effectively even when unidentified attacks occur. To that end, numerous studies have been conducted on enhancing resilience to malicious activities in the network using statistical strategies [25–30], homomorphic encryption approaches [31–33], randomization-based methods [34], and redundancy-based schemes [35–38].

Another critical element of the threat management process involves ensuring that agents within the network meet their privacy requirements. A privacy-preserving operation protects the private information of the network agents from being inferred by malicious adversaries. The literature contains various methods that address the privacy issues in distributed processing problems, such as consensus [39–41], optimization [42], and state estimation [43–46]. Differential privacy (DP) is the most widely adopted privacy-preserving approach in the literature on distributed state estimation [39, 47]. The DP technique uses perturbation to protect individual information from being inferred by other agents or eavesdroppers [39, 47]. However, since differentially private approaches come with a performance penalty, more recent consensus methods, such as noise-injection-based methods [48, 49], have gained wide acceptance. These methods improve the privacy-accuracy tradeoffs compared to DP approaches by injecting correlated random sequences into the local information. In the meantime, decomposition-based techniques were proposed to provide privacy by reducing the amount of information exchanged among neighbors. For instance, in [50], the initial state at each agent is randomly decomposed into two substates, one for inter-node interactions and another that remains invisible to other agents.

Overall, this dissertation examines distributed multi-agent scenarios where agents share private information with neighbors to complete a common task. The threat analysis is conducted here by designing the worst-case scenario of a linear data falsification attack from the perspective of a Byzantine adversary, as well as analyzing the impact of EE and HBC adversaries on the network performance. In the scope of threat management, this study proposes different methods to mitigate the impacts of data falsification attacks launched by Byzantine adversaries. In addition, this thesis develops privacy-preserving distributed learning algorithms to ensure the privacy of network agents in the presence of various adversaries. To evaluate the proposed algorithms, this thesis focuses, in particular, on distributed Kalman filtering (DKF) and distributed least mean square (D-LMS) scenarios. We focus on these algorithms since DKF and D-LMS are fundamental distributed estimation approaches that rely on consensus-based operations, exposing private information to potential adversaries in the network. As a result, the integrity of the entire network may be compromised and immediate privacy measures are required.

The developed algorithms in this thesis aim to guarantee privacy for network agents and provide robustness against adversaries. Thus, as a proof of concept, we analyzed the performance of our proposed algorithms on consensus-based distributed algorithms such as DKF and D-LMS scenarios. The DKF is employed as a base model, however, we also evaluate the performance of the proposed algorithms in the D-LMS scenario in order to show that the proposed algorithms are adaptable to other distributed learning strategies. As the literature has shown, the Kalman filter and LMS estimation algorithms are intrinsically related [51]; thus, this adaptation can also be theoretically justified.

1.1 Objectives

The main focus of this research is to address the need for privacy and robustness against potential adversaries in distributed settings. In particular, this thesis concentrates on distributed learning algorithms wherein agents share information with their neighbors and thereby expose private information to potential adversaries, resulting in heightened privacy concerns. This thesis addresses the following questions by analyzing potential threats in networks and proposing algorithms to mitigate their impacts. (i) How can we provide robustness against adversaries without compromising performance? (ii) What are the best ways to reliably complete a distributed inference task with a privacy guarantee for all agents without demanding a high computation load? Is it possible to improve communication efficiency at the same time? (iii) What is the performance of the proposed resilient methods under the worst-case scenario of an attack? In summary, this thesis focuses on the following research objectives:

T1: Analyze a distributed estimation scenario from an adversarial perspective and assess the performance of the network under the worst-case attack scenario (optimal attack strategy designed by an adversary).

T2: Improve the overall privacy-accuracy tradeoffs in the network by developing a privacy-preserving strategy that ensures agent privacy in the presence of various adversaries.

T3: Develop a distributed strategy with low computational complexity that provides robustness to adversaries without compromising performance significantly.

1.2 Methodology

In this thesis, theoretical algorithms are examined to address concerns about privacy and attack resilience in distributed scenarios with agents that are limited in power and computational resources. The methods proposed here are investigated in distributed Kalman filtering and D-LMS scenarios due to their simplicity in implementation and high accuracy. Finally, the thesis includes motivations, technical development of the algorithms proposed, and performance comparisons with contemporary approaches. In particular, each section includes a theoretical analysis to support the concepts proposed, followed by numerical experiments using practical case studies to validate the proposed methodology.

1.3 Thesis Contributions

The contributions of this thesis are devoted to answering the research concerns raised in **T1**-**T3** in Section 1.1.

To answer **T1**, the literature includes several attack designs such as the optimal jamming policy to maximize the estimation error [52] and linear deception attack designs to successfully bypass χ^2 and Kullback-Leibler (KL) false data detectors [53, 54]. Moreover, in a remote state estimation scenario, [55] investigates the impact of Byzantine agents on maximizing MSE by injecting zero-mean Gaussian attack sequences. However, to determine **T1**, the thesis focuses on coordinated data falsification attacks by a group of Byzantine agents in a consensus-based distributed Kalman filtering scenario. In particular, we design a coordinated data falsification attack that maximizes steady-state mean squared error (MSE). The adversary simultaneously optimizes the subset of Byzantine agents and their attack sequences to maximize the MSE of the network. Particularly, our proposed strategy, compared to the literature, considers a fully distributed scenario and optimizes both the attack sequence and the set of Byzantine agents simultaneously.

Furthermore, we propose a Byzantine-resilient consensus-based distributed filter (BR-CDF) that reduces the impact of the coordinated data falsification attack on the network performance. In addition to robustness against Byzantine attacks, the proposed BR-CDF algorithm reduces the communication load among agents by allowing agents to exchange only a fraction of their information at each given instant. Although the idea of sharing only fractions of information to reduce the communication load in distributed settings was originally proposed in [56, 57], to the best of our knowledge, it was not investigated in an adversarial environment. As a result of partial information sharing, Byzantine agents can control the sequence of information fractions they share, to further degrade the MSE of the network. Accordingly, we model the optimal attack strategy as a solution of an optimization problem where Byzantine agents cooperate on designing their attack covariances or the sequence of the information fractions they share.

In response to **T2**, the literature mainly focuses on filtering settings utilizing differential privacy techniques to protect private information where the resulting algorithm respects individual data [11, 58–60]. In contrast to indistinguishability approaches such as the DP technique, we use privacy constraints to protect the value of private information from being estimated by adversaries. Moreover, although the privacy of Kalman filtering algorithms has been investigated in the literature, the privacy framework for distributed solutions is not adequately covered. Thus, to address **T2**, this thesis proposes a privacy-preserving distributed Kalman filter (PP-DKF) that protects local agent information from being inferred by adversaries. The proposed PP-DKF protects local information by decomposing the local estimates into public and private substates and only sharing the public substate with neighboring agents. For further protection, it uses correlated perturbation sequences to obfuscate public substates before sharing. Moreover, the thesis analyzes the first- and second-order convergence properties of the PP-DKF and derives privacy bounds for agents in the presence of an EE and an HBC adversary. A set of numerical simulations is also provided to examine the robustness of the proposed PP-DKF compared to distributed Kalman filtering solutions employing contemporary privacy-preserving techniques. In addition, we propose partial sharing and privacy-preserving distributed learning (PPDL) algorithms that offer communication efficiency while preserving privacy. The proposed PPDL algorithms reduce communication load among agents through partial sharing of information and obtain privacy by noise injection and state decomposition average consensus techniques. We examine the first- and second-order convergence properties of the proposed PPDL algorithms and provide their privacy analysis in the presence of an HBC adversary.

Regarding the research concern in **T3**, the literature has proposed several methods for reducing the impact of Byzantine agents in distributed settings by assigning dynamic weights to measurements that are most likely to originate from a Byzantine agent [25–28]. Moreover, homomorphic encryption-, randomization-, and redundancy-based approaches were proposed to further suppress the impact of Byzantine agents [31, 34–36, 61]. Unlike these methods, which perform by increasing local computations, we address the concerns in **T3** by strengthening the robustness of the distributed Kalman filtering algorithm against coordinated Byzantine attacks without significantly increasing local computations. We formulate the DKF algorithm as a distributed optimization problem with consensus constraints. Using a total variation (TV) penalty term for the objective function and a distributed subgradient algorithm for solving the resulting optimization problem, we derive a suboptimal solution to the DKF that performs robustly in the presence of Byzantine agents. The proposed Byzantine-resilient distributed Kalman filter (BR-DKF) restricts the impact of Byzantine perturbations entirely, and the error bound is influenced only by the number of Byzantine agents.

1.3.1 List of Publications

The author of the dissertation conducted the following research studies in accordance with the research objectives described in Section 1.1. This dissertation is documented in papers **P1** to **P8** that cover the entire detailed contributions listed in Section 1.3. The list includes eight papers, six of which have been published or accepted for publication, and two were submitted during the Ph.D. program.

P1: A. Moradi, N. K. D. Venkategowda and S. Werner, "Coordinated Data-Falsification Attacks in Consensus-based Distributed Kalman Filtering," in Proceedings 8th *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2019, pp. 495-499.

P2: A. Moradi, V. C. Gogineni, N. K. D. Venkategowda and S. Werner, "Distributed Filtering Design with Enhanced Resilience to Coordinated Byzantine Attacks," submitted to *IEEE Transactions on Signal Processing*, pp. 1-10, 2022.

P3: A. Moradi, N. K. D. Venkategowda, S. P. Talebi and S. Werner, "Privacy-Preserving Distributed Kalman Filtering," in *IEEE Transactions on Signal Processing*, vol. 70, pp. 3074-3089, June 2022.

P4: A. Moradi, N. K. D. Venkategowda, S. Pouria Talebi and S. Werner, "Distributed Kalman Filtering with Privacy against Honest-but-Curious Adversaries," in Proceedings 55th *Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 790-794.

P5: A. Moradi, N. K. D. Venkategowda, S. Pouria Talebi and S. Werner, "Securing the Distributed Kalman Filter Against Curious Agents," in Proceedings 24th *IEEE International Conference on Information Fusion*, 2021, pp. 1-7.

P6: A. Moradi, N. K. D. Venkategowda and S. Werner, "Total Variation based Distributed Kalman Filtering for Resiliency Against Byzantines," in *IEEE Sensors Journal*, pp. 1-11, Jan. 2023.

P7: V. C. Gogineni, A. Moradi, N. K. D. Venkategowda and S. Werner, "Communication-Efficient and Privacy-Aware Distributed LMS Algorithm," in Proceedings 25th *IEEE International Conference on Information Fusion*, 2022, pp. 1-6.

P8: V. C. Gogineni, A. Moradi, N. K. D. Venkategowda and S. Werner, "Communication-Efficient and Privacy-Aware Distributed Learning," submitted to *IEEE Transactions on Signal and Information Processing over Networks*, pp. 1-13, 2023.

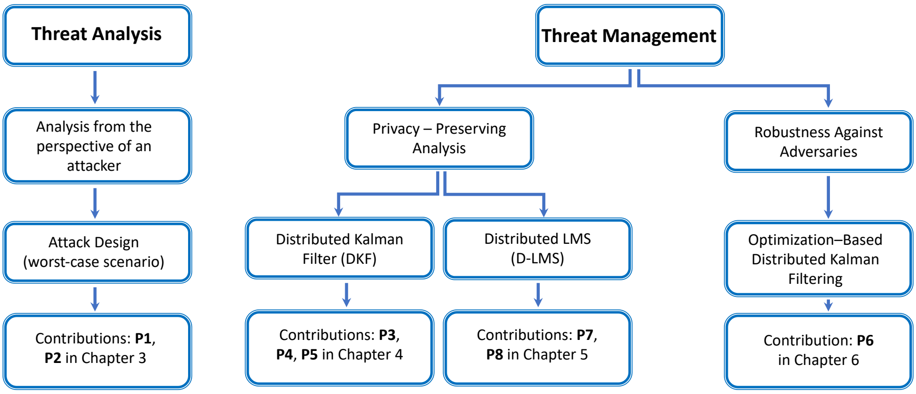

Threat Analysis
Analysis from the perspective of an attacker
Attack Design (worst-case scenario)
Contributions: **P1**, **P2** in Chapter 3
Threat Management
Privacy – Preserving Analysis
Distributed Kalman Filter (DKF)
Contributions: **P3**, **P4**, **P5** in Chapter 4
Distributed LMS (D-LMS)
Contributions: **P7**, **P8** in Chapter 5
Robustness Against Adversaries
Optimization–Based Distributed Kalman Filtering
Contribution: **P6** in Chapter 6

Figure 1.1: Thesis contributions and organization diagram.

A diagram illustrating the thesis organization and contributions can be found in Figure 1.1.

1.4 Thesis Organization

The next chapter mainly focuses on background information and mathematical tools necessary for a deeper understanding of the remainder of the thesis. In particular, we revisit the basics of distributed Kalman filtering algorithms, analysis of their optimal solutions, and potential privacy challenges in Chapter 2. Chapter 3 examines the performance of DKFs in the presence of coordinated data falsification attacks. Furthermore, it provides the design of an optimal attack strategy from the perspective of an attacker to maximize the network MSE. A privacy-preserving DKF is proposed in Chapter 4 that guarantees the privacy of network agents when an EE and an HBC adversary are present. Chapter 5 proposes communication efficient and private distributed learning algorithms that protect local information while reducing inter-agent communication load. Using an alternative methodology to solve the distributed Kalman filtering problem, Chapter 6 provides robustness against Byzantine attacks. Finally, Chapter 7 concludes the thesis with final remarks and future research directions.

Chapter 2

Distributed Estimation and Privacy Challenges

This chapter provides the background information required to follow the rest of the thesis. Section 2.1 investigates the distributed estimation scenarios with a focus on distributed Kalman filtering settings. Afterward, potential privacy breaches in distributed consensus-based algorithms are discussed along with solutions to mitigate their impact in Section 2.2. Further, the concept of privacy and metrics for characterizing it are investigated in Section 2.3. Section 2.4 discusses possible adversaries and their attack strategies, as well as the importance of researching the worst-case scenario from the perspective of an adversary. Lastly, this chapter revisits methods proposed in the literature to mitigate adversarial effects on the network in Section 2.5.

2.1 Distributed State Estimation: Kalman Filtering

The decentralized Kalman filtering problem was first introduced in [62, 63], both of which required a fully connected network. Because of the $\mathcal{O}(N^2)$ communication complexity, N is the number of agents in the network, the solution was not scalable for larger networks, and distributed Kalman filtering solutions were introduced where agents only interact with their neighbors. Using distributed Kalman filtering is one of the most popular methods for solving distributed estimation problems in scalable information fusion scenarios. A centralized Kalman filter provides an optimal estimation of the dynamic system state $\mathbf{x}_n \in \mathbb{R}^m$, at each time n, by observing the global observation $\mathbf{y}_n = [\mathbf{y}_{i,n}^{\mathrm{T}}, \cdots, \mathbf{y}_{i,n}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Nq}$. However, the purpose of distributed Kalman filtering is to convert a centralized Kalman filter into a network of local Kalman filters, where each agent i has a local observation

$\mathbf{y}_{i,n} \in \mathbb{R}^q$ and can only interact with its neighbors to provide the state estimate. One class of distributed Kalman filters (DKF) relies on average consensus operations and can be implemented by decomposing the centralized Kalman filter into local Kalman filters that estimate the state of the system and reach a consensus with neighboring agents on the state estimate [2–4]. Diffusion-based DKFs, however, provide state estimates without requiring neighboring agents to obtain the same estimate in steady-state [6]. The estimates of each agent are updated using convex combinations of the estimates of its neighbors. In consensus-based distributed Kalman filters (CDF), it generally takes multiple iterations to reach the average consensus across the network [64–66]. Although CDFs in [2–4] only require one consensus iteration, their limitations in selecting consensus weights distinguish them from diffusion-based solutions.

A variety of consensus-based distributed Kalman filtering techniques have also been proposed to improve performance in distributed estimation scenarios [8–10]. In [67], authors developed a DKF technique that mimics the operations of a centralized Kalman filter in a distributed fashion and improves the performance of the DKF using embedded average consensus fusion of local state estimates and their associated covariance information. In the following, we revisit the distributed Kalman filtering analysis, where agents employ their local observations to estimate the dynamic state of the system. Following is a summary of the commonly used mathematical operators throughout the thesis.

Mathematical Notations: Scalars, vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters. A white Gaussian sequence $\mathbf{x}(k)$ with covariance $\mathbf{\Sigma}$ is represented as $\mathbf{x}(k) \sim \mathscr{N}(\mathbf{0}, \mathbf{\Sigma})$. The greater than and less than symbols in the scalar inequalities are represented by $>$ and $<$, respectively. A positive semidefinite matrix $\mathbf{A}$ is denoted by $\mathbf{A} \succcurlyeq 0$ and $\mathbf{A} \succcurlyeq \mathbf{B}$ indicates that $\mathbf{A} - \mathbf{B}$ is a positive semidefinite matrix. The ijth element of the matrix $\mathbf{A}$, is denoted by $[\mathbf{A}]_{ij}$, while $\mathcal{A} \subseteq \mathcal{B}$ denotes that set $\mathcal{A}$ is a subset of a set $\mathcal{B}$. The inequality $\mathbf{A} \leq \mathbf{B}$ denotes an element-wise inequality for corresponding elements in matrices $\mathbf{A}$ and $\mathbf{B}$. The maximum and minimum eigenvalues of the square matrix $\mathbf{A}$ are denoted by $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$, respectively.

2.1.1 Consensus-based Distributed Kalman Filtering

A distributed Kalman filter consists of a network of agents modeled as an undirected graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of all agents with $|\mathcal{N}| = N$, and $\mathcal{E}$ is the edge set that represents the communication links between agents. The neighbor set $\mathcal{N}_i$ with the cardinality of $|\mathcal{N}_i|$ comprises all the immediate neighbors of agent i and excludes the agent itself. A network adjacency matrix is denoted by $\mathbf{E}$, with $e_{ij} = 1$ representing a connection between i and j and $e_{ij} = 0$ indicating that

$(i, j) \notin \mathcal{E}$. The diagonal matrix $\mathbf{D}$ containing the degrees of the corresponding nodes on the main diagonal is defined as $\mathbf{D} \triangleq \mathsf{diag}(\{|\mathcal{N}_i|\}_{i=1}^N)$.

We consider a dynamic process with a linear time-varying state model as

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{w}_n, \quad \forall n = 1, 2 \ldots, \tag{2.1}$$

where $\mathbf{x}_n \in \mathbb{R}^m$ denotes the state of the system at time instant n, $\mathbf{A} \in \mathbb{R}^{m\times m}$ is the state-transition matrix, and $\mathbf{w}_n$ is the state noise. In designing a distributed Kalman filtering algorithm, the ultimate goal is to collaboratively estimate the system state using a network of agents. The local observation $\mathbf{y}_{i,n} \in \mathbb{R}^q$ for each agent i is available at every instant n as

$$\mathbf{y}_{i,n} = \mathbf{H}_i\mathbf{x}_n + \mathbf{v}_{i,n}, \tag{2.2}$$

where $\mathbf{H}_i \in \mathbb{R}^{q\times m}$ is the observation matrix and $\mathbf{v}_{i,n}$ denotes the observation noise. The state and observation noise $\mathbf{w}_n$ and $\mathbf{v}_{i,n}$ are mutually independent zero-mean Gaussian processes with covariance matrices $\mathbf{Q} \in \mathbb{R}^{m\times m}$ and $\mathbf{R}_i \in \mathbb{R}^{q\times q}$, respectively. Employing the consensus-based distributed Kalman filter to estimate $\mathbf{x}_n$ in a collaborative manner [7], results in the estimate of the system state at each agent i and time instant n as

$$\hat{\mathbf{x}}_{i,n+1} = \mathbf{A}\hat{\mathbf{x}}_{i,n} + \mathbf{K}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n}\right) - \varepsilon\mathbf{A}\sum_{j\in\mathcal{N}_i}\left(\hat{\mathbf{x}}_{i,n} - \hat{\mathbf{x}}_{j,n}\right), \tag{2.3}$$

where $\mathbf{K}_{i,n} \in \mathbb{R}^{m\times q}$ is the Kalman gain, ε is the consensus gain chosen as $0 < \varepsilon \leq 1/\Delta$ with $\Delta \triangleq \max_i |\mathcal{N}_i|$, and $\hat{\mathbf{x}}_{j,n}$ is the estimate shared by neighboring agent j.

To improve the Kalman filtering state update in (2.3), the Kalman gain $\mathbf{K}_{i,n}$ has to be optimized for each agent $i \in \mathcal{N}$. The optimal Kalman gain $\mathbf{K}_{i,n}$ is designed by minimizing the trace of the estimation error covariance $\mathbf{P}_{i,n} \triangleq \mathbb{E}\{\mathbf{e}_{i,n}\mathbf{e}_{i,n}^{\mathrm{T}}\}$, where, the estimation error $\mathbf{e}_{i,n} \triangleq \hat{\mathbf{x}}_{i,n} - \mathbf{x}_n$ evolves as

$$\mathbf{e}_{i,n+1} = \mathbf{F}_{i,n}\mathbf{e}_{i,n} + \mathbf{K}_{i,n}\mathbf{v}_{i,n} - \mathbf{w}_n - \varepsilon\mathbf{A}\sum_{j\in\mathcal{N}_i}\left(\mathbf{e}_{i,n} - \mathbf{e}_{j,n}\right), \tag{2.4}$$

with $\mathbf{F}_{i,n} = \mathbf{A} - \mathbf{K}_{i,n}\mathbf{H}_i$. After some algebraic manipulations, the estimation error covariance matrix can be expressed as

$$\mathbf{P}_{i,n+1} = \mathbf{F}_{i,n}\mathbf{P}_{i,n}\mathbf{F}_{i,n}^{\mathrm{T}} + \mathbf{K}_{i,n}\mathbf{R}_i\mathbf{K}_{i,n}^{\mathrm{T}} + \mathbf{Q} + \Delta\mathbf{P}_{i,n} \tag{2.5}$$

with

$$\begin{aligned}\Delta\mathbf{P}_{i,n} = &- \varepsilon\mathbf{F}_{i,n}\sum_{s\in\mathcal{N}_i}\big(\mathbf{P}_{i,n} - \mathbf{P}_{is,n}\big)\mathbf{A}^{\mathrm{T}} - \varepsilon\mathbf{A}\sum_{r\in\mathcal{N}_i}\left(\mathbf{P}_{i,n} - \mathbf{P}_{ri,n}\right)\mathbf{F}_{i,n}^{\mathrm{T}} \\ &+ \varepsilon^2\sum_{r\in\mathcal{N}_i}\sum_{s\in\mathcal{N}_i}\mathbf{A}\left(\mathbf{P}_{i,n} - \mathbf{P}_{is,n} - \mathbf{P}_{ri,n} + \mathbf{P}_{rs,n}\right)\mathbf{A}^{\mathrm{T}},\end{aligned}$$

where $\mathbf{P}_{ij,n} \triangleq \mathbb{E}\{\mathbf{e}_{i,n}\mathbf{e}_{j,n}^{\mathrm{T}}\}$ is the cross-covariance term and is given by

$$\mathbf{P}_{ij,n+1} = \mathbf{F}_{i,n}\mathbf{P}_{ij,n}\mathbf{F}_{j,n}^{\mathrm{T}} + \mathbf{Q} + \Delta\mathbf{P}_{ij,n} \tag{2.6}$$

with

$$\begin{aligned}\Delta\mathbf{P}_{ij,n} =& -\varepsilon\mathbf{F}_{i,n}\sum_{s\in\mathcal{N}_j}\left(\mathbf{P}_{ij,n} - \mathbf{P}_{is,n}\right)\mathbf{A}^{\mathrm{T}} - \varepsilon\mathbf{A}\sum_{r\in\mathcal{N}_i}\big(\mathbf{P}_{ij,n} - \mathbf{P}_{rj,n}\big)\mathbf{F}_{j,n}^{\mathrm{T}}\\ &+ \varepsilon^2\sum_{r\in\mathcal{N}_i}\sum_{s\in\mathcal{N}_j}\mathbf{A}\left(\mathbf{P}_{ij,n} - \mathbf{P}_{is,n} - \mathbf{P}_{rj,n} + \mathbf{P}_{rs,n}\right)\mathbf{A}^{\mathrm{T}}.\end{aligned}$$

Subsequently, the optimal Kalman gain that minimizes trace of (2.5) can be computed by differentiating tr($\mathbf{P}_{i,n+1}$) with respect to $\mathbf{K}_{i,n}$ and is given by

$$\mathbf{K}_{i,n}^{*} = \mathbf{A}\Big(\mathbf{P}_{i,n} - \varepsilon\sum_{j\in\mathcal{N}_i}\big(\mathbf{P}_{i,n} - \mathbf{P}_{ji,n}\big)\Big)\mathbf{H}_i^{\mathrm{T}}\mathbf{M}_{i,n}^{-1}, \tag{2.7}$$

where $\mathbf{M}_{i,n} = \mathbf{H}_i\mathbf{P}_{i,n}\mathbf{H}_i^{\mathrm{T}} + \mathbf{R}_i$. As a result, the algorithm requires global information about the state covariances to compute the estimates. In the following section, we examine a distributed Kalman filtering solution that requires only the exchange of local state estimates and their covariance information among neighbors.

2.1.2 Distributed Kalman Filter with Embedded Average Consensus

The distributed Kalman filtering algorithm in [67] improves the filtering performance through embedded average consensus filters (ACF). The authors developed a DKF algorithm based on decomposing the centralized Kalman filtering steps and combining state estimates and covariance information with ACFs.

In a system with the same state dynamics and local observations as (2.1) and (2.2), respectively, the DKF algorithm in [67] estimates the system state by local prediction updates as

$$\begin{aligned}\hat{\mathbf{x}}_{i,n|n-1} &= \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}\\ \mathbf{P}_{i,n|n-1} &= \mathbf{A}\mathbf{P}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}\end{aligned} \tag{2.8}$$

where, for agent i, $\hat{\mathbf{x}}_{i,n|n-1}$ and $\hat{\mathbf{x}}_{i,n|n}$ are the respective *a priori* and *a posteriori* state vector estimates and $\mathbf{P}_{i,n|n-1}$ and $\mathbf{P}_{i,n-1|n-1}$ denote the respective *a priori* and *a posteriori* error covariance matrices. The intermediate information of agent i, at time instant n, denoted by $\mathbf{\Gamma}_{i,n}$, is locally updated as

$$\mathbf{\Gamma}_{i,n} = \mathbf{P}_{i,n|n-1}^{-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i, \tag{2.9}$$

and shared with neighbors to reach the average consensus. We assume that the condition for convergence of the covariance matrices $\{\mathbf{P}_{i,n|n} : \forall i \in \mathcal{N}, n = 1, 2, \ldots\}$

Algorithm 1 DKF algorithm

```
For each agent $i \in \mathcal{N}$
Initialize: $\hat{\mathbf{x}}_{i,0|0}$ and $\mathbf{P}_{i,0|0}$
$\hat{\mathbf{x}}_{i,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$
$\mathbf{P}_{i,n|n-1} = \mathbf{A}\mathbf{P}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$
$\mathbf{\Gamma}_{i,n} = \mathbf{P}^{-1}_{i,n|n-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i$
$\mathbf{P}^{-1}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \mathbf{\Gamma}_{j,n}\}$
$\boldsymbol{r}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + N\mathbf{P}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)$
$\hat{\mathbf{x}}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \boldsymbol{r}_{j,n}\}$
```

to unique stabilizing solutions, as given in [67], are satisfied. Therefore, we have $\lim_{n\to\infty} \mathbf{P}_{i,n|n} = \mathbf{P}_i$ for each $i \in \mathcal{N}$. As shown in [67], the *a posteriori* centralized covariance information is the network average of the updates in (2.9). Hence, the distributed update of $\mathbf{P}^{-1}_{i,n|n}$ is obtained via an ACF, wherein the agents refine their updates through local averaging within their neighborhoods. Finally, the *a posteriori* covariance information $\mathbf{P}^{-1}_{i,n|n}$ is used to determine the local intermediate state estimate

$$\boldsymbol{r}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + N\mathbf{P}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right) \tag{2.10}$$

that minimizes the trace of the error covariance matrix. Subsequently, similar to $\mathbf{\Gamma}_{i,n}$, the local intermediate state estimate is passed through an ACF to get the *a posteriori* state estimate $\hat{\mathbf{x}}_{i,n|n}$. The steps of the distributed Kalman filtering solution are summarized in Algorithm 1. In Algorithm 1, the general ACF algorithm is represented with the following schematic:

$$\mathbf{\Theta}_{i,n}(k) \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i \cup i : \mathbf{\Theta}_{j,n}(0)\} \tag{2.11}$$

where $\mathbf{\Theta}_{j,n}(0)$, $j \in \mathcal{N}_i \cup i$ are the initial inputs to the ACF at node i, and $\mathbf{\Theta}_{i,n}(k)$ is the output at agent i after k consensus iterations.[1] In theory, the ACF output is the average of inputs across the entire network, i.e., $\frac{1}{N}\sum_{j=1}^{N}\mathbf{\Theta}_{j,n}(0)$. However, in practice, the accuracy of the ACF is compromised due to the limited number of iterations. In the following section, we will investigate the dynamic of the average consensus filters in more detail.

2.2 Average Consensus and Breach of Privacy

An average consensus problem involves a group of agents who share their initial information with neighbors and attempt to obtain the average of the initial values

[1] In order to incorporate both the covariance information and the intermediate state estimate consensus updates, the ACF inputs are matrices.

across the network. The ACF benefits from local collaborations, where agents receive local information from neighbors and use that information to update their local states. In particular, in a generic iterative ACF, each agent i begins the process with initial information $\mathbf{\Theta}_{i,n}(0)$ and updates its state at kth consensus as

$$\mathbf{\Theta}_{i,n}(k) = b_{ii}\mathbf{\Theta}_{i,n}(k-1) + \sum_{j \in \mathcal{N}_i} b_{ij}\mathbf{\Theta}_{j,n}(k-1) \tag{2.12}$$

where consensus weights $\{b_{ij} : \forall i, j \in \mathcal{N}\}$ are positive real-valued weights so that the consensus weight matrix $\mathbf{B}$ where $b_{ij} = [\mathbf{B}]_{ij}$ is a doubly stochastic matrix [65]. [2] Since $\mathbf{B}$ is a doubly stochastic matrix, the state of each agent converges to the average consensus value as $k \to \infty$, given as

$$\lim_{k \to \infty} \mathbf{\Theta}_{i,n}(k) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{\Theta}_{j,n}(0) \tag{2.13}$$

which is the desired average consensus throughout the entire network. As seen in (2.12), agents need to share their current state information with their neighbors. The information exchange makes the consensus update vulnerable to malicious attacks, and adversaries can exploit node-sensitive information. Thus, the ultimate goal of agents in a private average consensus procedure is to reach the exact average consensus value without exposing their initial information. In ACF, the initial information of agents needs to be protected as the initial consensus iterations comprise messages with more node-specific information than those towards the end that are all close to the final network average. For example, consider a group of individuals that wants to calculate the average salary of everyone without revealing their personal salaries. Using distributed average consensus as a model for obtaining the average salary, the salary of each individual is the initial value of the ACF, the private information that has to be protected.

2.2.1 Privacy-Preserving Average Consensus Techniques

In the literature, there are many works devoted to the topic of protecting average consensus operations. Regarding privacy concerns in distributed estimation, differential privacy (DP) dominates the literature [39, 47]. In DP, local messages are perturbed with uncorrelated random sequences to protect individual information from being inferred by other agents or an external eavesdropper [39, 47]. In particular, DP randomizes the private information of agents, initial states in the ACF problem, in a manner that an adversary cannot infer the data of any individual agent based on the observation of aggregated output. For further clarifying the DP

[2] Throughout the entire manuscript, the filtering time instant is shown by n as an index and the internal loop that represents the consensus iterations is shown by k in parenthesis.

procedure, we consider two initial states $\mathbf{x}(0) = [\mathbf{x}_1^{\mathrm{T}}(0), \cdots, \mathbf{x}_N^{\mathrm{T}}(0)]^{\mathrm{T}} \in \mathbb{R}^{Nm}$ and $\mathbf{y}(0) = [\mathbf{y}_1^{\mathrm{T}}(0), \cdots, \mathbf{y}_N^{\mathrm{T}}(0)]^{\mathrm{T}} \in \mathbb{R}^{Nm}$ as σ-adjacent when there is an $i \in \mathcal{N}$ such that $\|\mathbf{x}_i(0) - \mathbf{y}_i(0)\| \leq \sigma$ and $\mathbf{x}_j(0) = \mathbf{y}_j(0)$ for each $i \neq j$. Thus, a randomized mechanism $\mathcal{A} : \mathbb{R}^{Nm} \rightarrow \mathcal{R}(\mathcal{A})$ with range $\mathcal{R}(\mathcal{A})$ is (ϵ, δ)-differentially private, if for each subset of range $\mathcal{A}$, i.e., $\mathcal{O} \subseteq \mathcal{R}(\mathcal{A})$, the following inequality holds:

$$\mathbb{P}\{\mathcal{A}(\mathbf{x}(0)) \in \mathcal{O}\} \leq e^{\epsilon}\, \mathbb{P}\{\mathcal{A}(\mathbf{y}(0)) \in \mathcal{O}\} + \delta, \tag{2.14}$$

where $\mathbb{P}\{\cdot\}$ denotes the probability of the argument event and the symbol $\in$ indicates set membership. This essentially means the output of a differentially private algorithm is not significantly affected by changing the data of a single agent. If $\delta = 0$, the randomized mechanism $\mathcal{A}$ is called ϵ-differentially private that guarantees stronger privacy than the (ϵ, δ)-differentially-private mechanism [47, 68–70]. A differentially private procedure provides privacy guarantees, but at the cost of performance. Hence, more recent privacy-preserving consensus approaches such as noise-injection-based methods [48, 49] have gained wide acceptance due to their improved privacy-accuracy tradeoffs.

The noise-injection-based approach designs a correlated noise process to perturb the average consensus states at every iteration without impacting the consensus result. At each agent i and consensus iteration k, the state of the agent is perturbed with the additive perturbation noise sequence $\boldsymbol{\omega}_i(k) \in \mathbb{R}^m$ given as

$$\boldsymbol{\omega}_i(k) = \begin{cases} \boldsymbol{\nu}_i(0) & k = 0 \\ \phi^k \boldsymbol{\nu}_i(k) - \phi^{k-1} \boldsymbol{\nu}_i(k-1) & \text{o.w.} \end{cases} \tag{2.15}$$

with $\phi \in (0, 1)$ as a common constant for all agents and $\boldsymbol{\nu}_i(k) \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_m)$ as an independent and identically distributed white Gaussian sequence for each iteration k and $i \in \mathcal{N}$. In particular, we employ a decaying variance perturbation sequence to protect the initial exchanges more than the output of the ACF. In other words, the decaying variance perturbation sequence injects higher variance noise to the initial values of the ACF, and as the agents converge toward the average consensus, less noise is injected since there is less node-specific information. As a result of (2.15), the initial state of the agents is protected, and the exact value of the average consensus in the ACF problem can also be obtained asymptotically [48]. Decomposition-based techniques, on the other hand, focus primarily on the amount of information that is exchanged with neighbors [50]. In these methods, the initial state of each agent is randomly decomposed into public and private substates, where only the public state is shared with neighbors. The private substate, however, updates internally and remains invisible to other agents.

The majority of distributed algorithms, including DKF and LMS algorithms, must be protected against adversaries because they also involve average consensus steps.

These algorithms must modify their average consensus steps to ensure privacy, but these modifications impact the overall performance of the algorithm. Hence, as an alternative objective, this dissertation also investigates the impact of these modifications on the overall performance of distributed algorithms, in addition to the privacy guarantee provided for network agents. Defining privacy and introducing a metric to measure it are prerequisites for assessing privacy guarantees. To this end, in the next section, we explore the definition and measures of privacy.

2.3 Privacy Measures

The concept of privacy does not have a universally accepted definition [71]. In the information domain, Westin [72] defined privacy as the right to control and handle the amount of information communicated to others. The collection and exchange of data is an essential component of distributed algorithms, although it can cause unwanted privacy violations. In these scenarios, information leakage can be prevented by modifying the original data and preventing the disclosure of individual information. However, the transformation of the data reduces the performance of the algorithm and the quality of the data, thereby causing inaccurate knowledge extraction.

It is crucial to choose a privacy-preserving technique and then a privacy metric based on system properties. Privacy metrics use system properties as inputs, such as information available to the adversary and the type and capabilities of the adversary to quantify the privacy level of agents. In distributed learning and estimation algorithms, DP-based privacy metrics [73] are commonly used when the attack model and the information set available to the adversary are not specified. In DP-based approaches, privacy is not quantified, but rather, the indistinguishability of agent data is guaranteed. The indistinguishability-based privacy metrics, e.g., DP-based metric, determine whether the adversary can distinguish between the information of different agents [74]. In general, DP is a pessimistic metric that gives a worst-case privacy measure, requiring algorithms to employ a larger variance of the perturbations. In other words, DP algorithms inject more perturbation noise than necessary to guarantee privacy, resulting in a significant performance loss and an upper bound instead of a tight bound for privacy leakage. For example, the DP is used as a privacy measure in [60] where a centralized aggregator collects local private data streams from agents to compute minimum MSE estimates of the system state. DP is an appropriate privacy measure in [60] because there is no quantitative definition of privacy and no explicit model of adversary capabilities.

Alternatively, in distributed Kalman filtering, for example, the adversary can exploit the prior information, such as the distribution of states, statistics of perturbation noise, observation dynamics, and network information, to infer the state estimate values. In these scenarios, probabilistic indistinguishability metrics such as differential privacy or mutual information-based methods ignore the impact of prior information [60, 75]. Therefore, these measures are not appropriate for capturing the behavior of adversaries or how accurate their knowledge of private information is. In this work, the main concern is to protect information from being inferred by adversaries, and due to the availability of a specified attack model and information set of the adversary, the MSE metric is better suited. The MSE metric quantifies the uncertainty of the adversary to estimate the private information of agents based on the data it has access to [74]. Overall, the DP technique captures an abstract notion of privacy, whereas the estimation error quantifies the exact knowledge of the adversary about private information. Thus estimation error or MSE is more pragmatic and better suited for practical application in multi-agent networks since it can characterize a tighter privacy bound [48, 76].

2.4 Adversaries and Attack Strategies

According to the previous section, measurement of privacy using the MSE metric requires an understanding of the capabilities of the adversary and the specified attack model. Hence, this section investigates possible adversaries and their attack strategies in distributed learning and estimation scenarios.

2.4.1 Adversaries

In the scope of threat management and analysis, it is imperative to understand possible adversaries and their capabilities clearly. We consider three types of adversaries that can exploit the exchanged data to infer private information:

- An *external eavesdropper*, who is external to the network, is trying to learn private information by accessing all the information exchanged among agents.
- An *honest-but-curious agent* is a legitimate network agent that contributes to the overall estimation task while passively attempting to infer private information from messages shared by its immediate neighbors.
- A *Byzantine agent* is a legitimate network agent that injects falsified information into the estimation process to impair the overall network performance.

Essentially, a Byzantine adversary is an HBC agent that actively injects false information into the network. Based on the definitions, the EE can only access the exchanged information among agents and does not have access to node-specific data, while HBC and Byzantine adversaries are network agents and can access information of neighboring agents and their node-specific data. Adversaries above

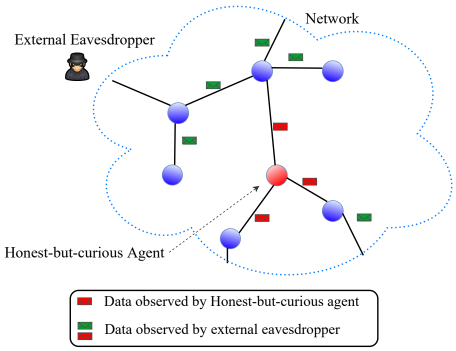

Network
External Eavesdropper
Honest-but-curious Agent
🟥 Data observed by Honest-but-curious agent
🟩🟥 Data observed by external eavesdropper

Figure 2.1: Illustration of information accessible to various adversaries.

can access different types, and amounts of information; Fig 2.1 illustrates the different information types accessible to adversaries.

2.4.2 Attack Strategies

Attacks in multi-agent networks can be classified as either active or passive. In passive attacks, the adversary intercepts a communication link and attempts to infer private information about the network without actively falsifying the exchanged information. The EE and HBC adversaries both perform passive attacks because they neither inject false information into the network nor interfere with information exchange by interrupting communication links. In contrast, active attacks can deteriorate the overall performance of a network by manipulating information or injecting false data. Generally, active attacks are divided into two categories: denial-of-service (DoS) attacks and integrity attacks. During DoS attacks, communication link between agents is blocked, and information cannot be exchanged [14], while integrity attacks occur when adversaries inject false information [15]. In particular, an integrity attack that injects false information into a network without being detected is referred to as a stealthy attack.

In this thesis, we consider both passive and active attack strategies. In the scope of passive attacks, we consider both EE and HBC adversaries that attempt to infer the private information of network agents by accessing different information sets. For example, in the distributed Kalman filtering scenario in Section 2.1.2, agents must share intermediate state estimates and error covariance matrices to perform the filtering operations. Hence, for each filtering time instant n, the accessible information set for the EE is $\mathcal{I}_{\text{EE}} = \{\boldsymbol{\Gamma}_{i,n}, \mathbf{r}_{i,n}, \ \ \forall i \in \mathcal{N}\}$, whereas for the HBC adversary, assuming agent j as an HBC agent, $\mathcal{I}_{\text{HBC}} = \{\boldsymbol{\Gamma}_{j,n}, \mathbf{r}_{j,n}\} \bigcup \{\boldsymbol{\Gamma}_{i,n}, \mathbf{r}_{i,n}, \ \ \forall i \in \mathcal{N}_j\}$ is the available information set which is restricted to its neighborhood.

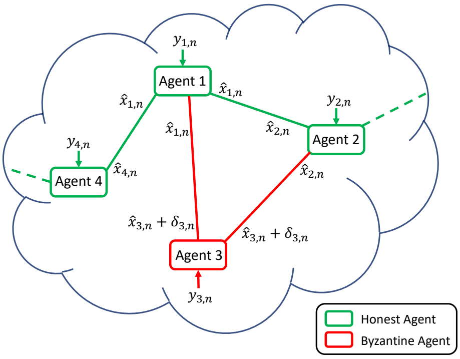


Figure 2.2: Illustration of information exchange in the presence of Byzantine agents.

When it comes to active attacks, we consider the presence of Byzantine agents. We assume that a group of agents are Byzantines, $\mathcal{B} \subset \mathcal{N}$, and in contrast to regular agents, share a falsified version of their information with their neighbors to deteriorate the network performance [15]. To simplify the analysis, we consider a linear data falsification attack, in which Byzantine agents manipulate their information by injecting additive random sequences before sharing it with neighbors. For example, in a DKF algorithm, Byzantine agents participate in the filtering algorithm by sharing the perturbed version of their state estimates. As a result and as shown in Fig. 2.2, the received state estimate at each agent j can be expressed as

$$\bar{\mathbf{x}}_{j,n} = \begin{cases} \hat{\mathbf{x}}_{j,n} + \boldsymbol{\delta}_{j,n} & j \in \mathcal{B} \\ \hat{\mathbf{x}}_{j,n} & j \notin \mathcal{B}, \end{cases} \tag{2.16}$$

where at each time instant n, $\boldsymbol{\delta}_{j,n} \in \mathbb{R}^m$ is the perturbation sequence of the Byzantine agent. To maximize the attack stealthiness, the ability to evade detection, we consider the perturbation sequence to be zero-mean Gaussian with covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{m \times m}$ [54, 77]. In addition, instead of the independent Byzantine attack, i.e., $\mathbb{E}\{\boldsymbol{\delta}_{i,n}\boldsymbol{\delta}_{j,n}^{\mathrm{T}}\} = \mathbf{0}$ for all $i \neq j$, Byzantine agents can further deteriorate the network performance by cooperatively designing their attack covariances. The coordinated attack is modeled with a correlated covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}\{\boldsymbol{\delta}_n\boldsymbol{\delta}_n^{\mathrm{T}}\}$ where $\boldsymbol{\delta}_n = [\boldsymbol{\delta}_{1,n}^{\mathrm{T}}, \cdots, \boldsymbol{\delta}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}$ is the network-wide perturbation sequence and $\boldsymbol{\delta}_{j,n} = \mathbf{0}$ if $j \notin \mathcal{B}$.

2.4.3 Attack Design: Perspective of an Adversary

The evaluation of distributed learning algorithms can be conducted by investigating their effectiveness under the worst-case scenario of a cyber-attack. As a result, potential loopholes and critical agents/links are identified. Hence, analyzing optimal attack strategies from the perspective of an adversary is crucial for developing attack protection methods.

The literature includes many studies on optimal attack designs from the perspective of an adversary; for example, in [16], authors propose a false data injection attack strategy in a remote state estimation scenario that maximizes the trace of the estimation error covariance. Regarding the security concerns of power grid devices, by using a game theory framework, a stealthy attack strategy and its optimal defense mechanism are proposed in [78]. Moreover, authors in [18] propose a linear stealthy attack strategy and its required feasibility constraints to evade detection, while [17] proposes an optimal attack strategy and sufficient conditions to destabilize the system, where estimation error grows unlimited. In addition, considering a distributed cyber-physical system (CPS), [79] designs a data falsification attack that enforces the state estimate of agents to remain within a pre-specified range.

In this thesis, we design the optimal attack strategy by investigating a DKF from the perspective of an adversary and maximizing the network MSE. The adversary determines the optimal attack strategy by jointly optimizing the attack covariances and the subset of agents that it compromises. In the same system, we mitigate the impact of the Byzantine attack by allowing agents to share only a fraction of information at any given instant. In this case, we also find the optimal attack solution by solving an optimization problem where Byzantine agents cooperate on designing their attack covariances or the order of the information fractions they share.

2.5 Attack Mitigation Approaches

After exploring attack strategies and attack design techniques, we investigate protection schemes from the perspective of a network agent. To reduce the impact of adversaries, one approach is to detect the potential adversaries and implement correction measures [19–21]. As an example, [80] proposes a defense strategy that detects adversarial agents based on changes in their innovation signals and tailors their gains accordingly. Studies in [18, 22, 23] have shown that relying on attack detection to mitigate the impact of adversaries has limited utility when attacks are stealthy. Therefore, it is essential to develop robust algorithms for unidentified attacks.

To that end, in distributed filtering scenarios, works in [25, 26] minimize the impact of malicious agents using innovation signal statistics to re-design the consensus weights. Moreover, a Byzantine-resilient distributed state estimation algorithm is proposed in [81], which allows agents to update state estimates locally by selecting the best subset of neighbors for updating the state estimate. In a distributed state estimation scenario, [27, 28] provide resilience to measurement attacks by assigning adaptive weights to received measurements from neighbors. By assigning smaller weights to measurements whose norm exceeds a certain threshold, they would have a smaller impact on the state estimates. Furthermore, [29] proposes a secure state estimation method that employs the median of neighboring estimates rather than the mean, which provides robustness against adversaries.

To further ensure the confidentiality of signals sent over the network, homomorphic encryption schemes have also been used in CPSs [31]. In [32], the authors propose an algorithm employing additively homomorphic encryption, which enables the cloud server and security module to integrate the information of multiple parties while maintaining data privacy. However, authors in [33] propose a modified encoding and decoding scheme that, unlike the previous work in [61], does not negatively affect estimation performance in the absence of attacks and further protects data integrity in multi-sensor networks. Moreover, utilizing randomization-based methods to disrupt and mislead attackers in their malicious activities is a less resource-intensive method to mitigate their impacts on the network [34]. To further improve the resistance against adversarial attacks, redundancy-based approaches were introduced to a CPS at different levels of communication, channels, software, and hardware [35, 36]. Redundant subsystems serve as backups or parallel integrity verification units to reduce the effect of malfunctioning behaviors in the network [37]. An approach based on redundancy demands strict network requirements and can only tolerate a limited number of Byzantine adversaries. Thus, authors in [38] provide resilience to attack by limiting these stringent requirements to only a group of agents. Generally, these approaches reduce the impact of adversarial attacks on the network, but they require more local computations and information transfer in the network, which is undesirable in resource-constrained situations. This manuscript investigates approaches that improve robustness against adversaries and mitigate their impacts without adding extra computational burden to agents.

2.6 Summary

In this chapter, we presented the background information on distributed estimation in multi-agent systems, focusing on distributed Kalman filtering scenarios. Afterward, a discussion of possible privacy breaches in the system and modifications that can be made to safeguard privacy in average consensus scenarios was presented. A rationale for using the MSE as the privacy measure throughout the thesis was established through the introduction of privacy definitions in the literature. Furthermore, possible adversaries and attack strategies were discussed, and the importance of researching an optimal attack strategy from the perspective of an attacker was clarified. Finally, the literature on mitigation approaches to reduce the impact of adversaries on the network has been investigated. In the next chapter, in order to conduct threat analysis, we present the results of publications **P1** and **P2** in which we examine the impact of coordinated data falsification attacks on consensus-based distributed Kalman filtering settings.

Chapter 3

Threat Analysis and Optimal Attack Design

The chapter begins by presenting the results of publication **P1**, which examines how perturbations injected by the Byzantine agents can disrupt the distributed filtering algorithm and degrade the MSE performance. We examine the CDF problem from an adversarial perspective and identify the worst-case attack strategy by optimizing attack sequences and the set of compromised agents. As mentioned in Section 2.5, agents also implement different mitigation approaches to reduce the impact of adversaries in the network. In this regard, in publication **P2**, we propose a CDF algorithm that allows agents to share a fraction of state estimates at each time instant. We show that by sharing only a fraction of information, agents reduce the impact of coordinated data falsification attacks on network performance. Accordingly, the optimal attack is designed by Byzantine agents cooperating to optimize their attack sequences and the order of the information fractions they share to maximize the network MSE.

The remainder of this chapter is organized as follows. Section 3.1 investigates the optimal coordinated data falsification attack by jointly optimizing attack sequences and the set of the compromised agents. A partial-sharing-based CDF algorithm is presented and analyzed for achieving robustness against Byzantine attacks in Section 3.2. Furthermore, Section 3.2.4 analyzes the optimal attack strategy by optimizing the sequence of the information fractions at Byzantine agents. Lastly, Section 3.3 summarizes the chapter.

3.1 Coordinated Data Falsification Attack Design

Optimal attack designs are examined from the perspective of adversaries to identify critical links and agents in a system. The optimal jamming policy that maximizes the estimation error in a remote state estimation scenario was proposed in [52]. An optimal linear deception attack that successfully bypasses a χ^2 false data detector was also proposed in [53]. Further, [54] examines the impact of a stealthy data falsification attack on a single sensor Kalman filter with a Kullback-Leibler (KL) divergence-based detector. In a similar setting, [55] showed that Byzantine agents can maximize MSE, the worst-case stealthy attack strategy, by employing zero-mean Gaussian attack sequences. In contrast to studies in [52–55, 77], we consider a fully distributed scenario and jointly optimize the attack sequence and set of Byzantine agents. Our model considers the attack design in a fully distributed scenario, so comparing it with existing methods in the literature cannot be done fairly. Although our proposed method is benchmarked with different strategies, we may be able to extend the proposed approaches in [53] and [54] to distributed settings and incorporate χ^2 and KL divergence-based false data detectors in our algorithm to provide a fair comparison.

We consider a connected multi-agent network of N agents that collectively aim to estimate the state vector sequence $\{\mathbf{x}_n,\, n = 1,\, 2\, \ldots\}$ from local observations $\{\mathbf{y}_{i,n},\, n = 1,\, 2\, \ldots,\, i \in \mathcal{N}\}$. Employing the CDF algorithm in Section 2.1.1, each agent i tracks the state of the system according to its local estimation as (2.3) that minimizes the trace of the estimation error covariance. We assume that the CDF algorithm suffers from a coordinated Byzantine attack as described in Section 2.4.2, the attacker can only collude with a limited number of agents. Additionally, we consider the case where Byzantine agents have limited computation and energy resources. Therefore, to satisfy resource limitations and keep the Byzantine attack stealthy, i.e., the ability to evade detection, we restrict the variance of the injected sequences into the network, i.e., $\mathsf{tr}(\mathbf{\Sigma}) \leq \eta$. Thus, the stealthiness constraint states that information from neighbors cannot be trusted as an honest agent is indistinguishable from a Byzantine agent.

3.1.1 Problem Statement

In this section, the distributed filtering algorithm is examined from the perspective of an adversary to determine the optimal attack strategy for causing maximum network performance degradation. The main objective of the Byzantine attack is to maximize the network-wide mean squared error (NMSE), defined as

$$\text{NMSE} \triangleq \limsup_{N' \to \infty} \frac{1}{N'} \sum_{n=1}^{N'} \sum_{i=1}^{N} \mathsf{tr}\big(\mathbf{P}_{i,n}\big), \tag{3.1}$$

while still maintaining a desired level of stealthiness. Due to limited resources at the adversary, only a subset of agents can be Byzantines. We need to decide the subset of agents that participate in the attack and determine the covariance matrices $\mathbf{\Sigma}_j$, $j \in \mathcal{B}$, of the corresponding falsification sequences. To that end, we introduce the Boolean variable $z_j = 1$ if $j \in \mathcal{B}$ and zero otherwise. Accordingly, we define the selection vector $\mathbf{z} \triangleq [z_1, z_2, \ldots, z_N]^{\mathrm{T}}$ by stacking all the Boolean variables [82]. As a result, the optimal attack strategy can be expressed as an optimization problem given by

$$\begin{aligned} \max_{\mathbf{\Sigma},\,\mathbf{z}}. \quad & \text{NMSE} \\ \text{s. t.} \quad & \mathsf{tr}(\mathbf{\Sigma}) \leq \eta, \\ & \mathbf{\Sigma} \succcurlyeq 0, \\ & \mathbf{z} \in \{0,\, 1\}^N, \quad \mathbf{1}^{\mathrm{T}}\mathbf{z} = B, \end{aligned} \tag{3.2}$$

where the first constraint is related to stealthiness and the last constraint limits the number of Byzantine agents to $|\mathcal{B}| = B$. The parameter η is employed to limit the total power of the falsification sequences and satisfy detection-avoidance targets, and $\mathbf{\Sigma}$ represents the network-wide attack covariance that must be designed as a positive semidefinite matrix. In the next section, we compute the network-wide mean squared error as a function of attack covariance matrices. The ultimate goal is to propose a strategy for the joint design of the attack sequence and subset of Byzantines to maximize the trace of the steady-state error covariance.

3.1.2 Joint Selection of Byzantine Agents and Attack Sequences

To solve problem (3.2), we must first derive the expression for the objective function to capture the NMSE. Given the state estimate (2.3), we assume that the Byzantine attack begins at $n = n_0$ when the Kalman filter has reached the steady-state. The local state estimate of each agent i is updated as

$$\tilde{\mathbf{x}}_{i,n+1} = \mathbf{A}\tilde{\mathbf{x}}_{i,n} + \tilde{\mathbf{K}}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\tilde{\mathbf{x}}_{i,n}\right) - \varepsilon\mathbf{A}\sum_{j\in\mathcal{N}_i}\left(\tilde{\mathbf{x}}_{i,n} - \bar{\mathbf{x}}_{j,n}\right), \tag{3.3}$$

where at each time instant n, $\tilde{\mathbf{x}}_{i,n}$ denotes the state estimate of agent i in the presence of Byzantine attack and $\bar{\mathbf{x}}_{j,n}$ represents the received information from the neighboring agent j as in (2.16). Given the local estimation error as $\tilde{\mathbf{e}}_{i,n} \triangleq \tilde{\mathbf{x}}_{i,n} - \mathbf{x}_n$, the network-wide estimation error in presence of Byzantine attack is defined as $\tilde{\mathbf{e}}_n \triangleq [\tilde{\mathbf{e}}_{1,n}^{\mathrm{T}}, \ldots, \tilde{\mathbf{e}}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}$ where the local estimation error evolves as

$$\tilde{\mathbf{e}}_{i,n+1} = (\mathbf{A} - \tilde{\mathbf{K}}_{i,n}\mathbf{H}_i)\tilde{\mathbf{e}}_{i,n} - \mathbf{w}_n + \tilde{\mathbf{K}}_{i,n}\mathbf{v}_{i,n} - \varepsilon\mathbf{A}\sum_{j\in\mathcal{N}_i}\big(\tilde{\mathbf{e}}_{i,n} - \tilde{\mathbf{e}}_{j,n} - \boldsymbol{\delta}_{j,n}\big). \tag{3.4}$$

Accordingly, the evolution of the network-wide estimation error is expressed as

$$\tilde{\mathbf{e}}_{n+1} = \bar{\mathbf{A}}_n \tilde{\mathbf{e}}_n + \bar{\mathbf{b}}_n + \varepsilon \boldsymbol{\Gamma} \boldsymbol{\delta}_n, \tag{3.5}$$

where $\boldsymbol{\Gamma} = \mathbf{E}\,\mathsf{diag}(\mathbf{z}) \otimes \mathbf{A}$,

$$\begin{aligned} \bar{\mathbf{A}}_n &= (\mathbf{I}_N - \varepsilon \mathbf{L}) \otimes \mathbf{A} - \mathsf{Blockdiag}(\{\tilde{\mathbf{K}}_{i,n}\mathbf{H}_i\}_{i=1}^{N}), \\ \bar{\mathbf{b}}_n &= \mathsf{diag}(\{\tilde{\mathbf{K}}_{i,n}\mathbf{v}_{i,n}\}_{i=1}^{N}) - \mathbf{1}_N \otimes \mathbf{w}_n, \end{aligned} \tag{3.6}$$

with $\tilde{\mathbf{K}}_{i,n}$ as the Kalman gain and $\mathbf{L} = \mathbf{D} - \mathbf{E}$ is the network Laplacian. From (3.5), the error covariance $\tilde{\mathbf{P}}_{n+1} \triangleq \mathbb{E}\{\tilde{\mathbf{e}}_{n+1}\tilde{\mathbf{e}}_{n+1}^{\mathrm{T}}\}$ is obtained as

$$\tilde{\mathbf{P}}_{n+1} = \bar{\mathbf{A}}_n \tilde{\mathbf{P}}_n \bar{\mathbf{A}}_n^{\mathrm{T}} + \bar{\mathbf{Q}}_n + \varepsilon^2 \boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^{\mathrm{T}}, \tag{3.7}$$

where $\bar{\mathbf{Q}}_n = \mathsf{Blockdiag}(\{\tilde{\mathbf{K}}_{i,n}\mathbf{R}_i\tilde{\mathbf{K}}_{i,n}^{\mathrm{T}}\}_{i=1}^{N} + \mathbf{1}_N\mathbf{1}_N^{\mathrm{T}} \otimes \mathbf{Q}$. The optimal Kalman gain is found by differentiating the trace of (3.7) with respect to $\tilde{\mathbf{K}}_{i,n}$, given by

$$\tilde{\mathbf{K}}_{i,n} = \mathbf{A}\Big(\tilde{\mathbf{P}}_{i,n} - \varepsilon \sum_{j \in \mathcal{N}_i} \big(\tilde{\mathbf{P}}_{i,n} - \tilde{\mathbf{P}}_{ji,n}\big)\Big)\mathbf{H}_i^{\mathrm{T}}\tilde{\mathbf{M}}_{i,n}^{-1}, \tag{3.8}$$

where $\tilde{\mathbf{M}}_{i,n} = \mathbf{H}_i\tilde{\mathbf{P}}_{i,n}\mathbf{H}_i^{\mathrm{T}} + \mathbf{R}_i$. Therefore, when Byzantine attacks are present, dynamics of the error covariance and Kalman gain are captured as in (3.7) and (3.8). Assuming that the network is connected, controllability and observability requirements of the system are met, it can be shown that $\lim_{n\to\infty}\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}$ i.e., $\tilde{\mathbf{P}}_n$ converges to a bounded matrix. In other words, there exists a matrix $\tilde{\mathbf{K}}_{i,n}$ such that $\tilde{\mathbf{P}}_n$ is bounded and converges to a unique positive definite matrix for all n and any initial non-negative symmetric matrix. Since obtaining a closed form expression for the covariance matrix of the error in (3.4) is intractable, we employ $\mathsf{tr}(\tilde{\mathbf{P}})$ as a proxy to the objective function. Since the actual NMSE represents a time-average of error covariances, $\mathsf{tr}(\tilde{\mathbf{P}})$ can be considered as a lower bound of the NMSE in (3.1).

The solution to the Riccati equation in (3.7) can be obtained by solving a semi-definite programming (SDP) problem [83]. Motivated by this fact and substituting $\mathsf{NMSE} = \mathsf{tr}(\tilde{\mathbf{P}})$ in (3.2), we express the joint Byzantine agent selection and attack design optimization problem as

$$\begin{aligned} \mathcal{P}: \quad \max_{\mathbf{X},\boldsymbol{\Sigma},\mathbf{z}} \quad & \mathsf{tr}(\mathbf{X}) \\ \text{s. t.} \quad & \mathbf{X} \succeq \bar{\mathbf{A}}\mathbf{X}\bar{\mathbf{A}}^{\mathrm{T}} + \bar{\mathbf{Q}} + \varepsilon^2\boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^{\mathrm{T}}, \\ & \boldsymbol{\Gamma} = \mathbf{E}\,\mathsf{diag}(\mathbf{z}) \otimes \mathbf{A}, \\ & \mathbf{X} \succeq 0, \\ & \mathsf{tr}(\boldsymbol{\Sigma}) \le \eta, \\ & \boldsymbol{\Sigma} \succeq 0, \\ & \mathbf{1}^{\mathrm{T}}\mathbf{z} \le B, \quad z_i \in \{0,1\}, \quad i = 1, \ldots N. \end{aligned} \tag{3.9}$$

The problem above is NP-hard [84] and difficult to solve due to the non-convex quadratic terms in the first constraint and Boolean variables in the last constraint. Several methods are presented in subsequent sections to find a suboptimal solution to the problem in (3.9).

Block-Coordinate Descent (BCD) Approach

The problem in (3.9) is non-convex due to the Boolean variables. To circumvent this, we relax the Boolean constraint $z_i \in \{0, 1\}$ to a linear inequality constraint $0 \leq z_i \leq 1$. We see that for a given $\mathbf{z}$ or $\mathbf{\Sigma}$, the problem (3.9) is an SDP, as its first constraint is convex. Therefore, we employ the block-coordinate descent (BCD) method where $(\mathbf{X}, \mathbf{\Sigma})$ and $(\mathbf{X}, \mathbf{z})$ are alternately optimized with the other variable fixed. Also, applying the trace operator on both sides of the convergence constraint leads to a linear approximation with respect to $\mathbf{z}$ and $\mathbf{\Sigma}$. The BCD approach starts with an arbitrary $\mathbf{z}_0$ as the initial condition, and the first step is given by

$$\begin{aligned} \mathcal{P}_1: \quad \max_{\mathbf{X},\mathbf{\Sigma}}. \quad & \mathsf{tr}(\mathbf{X}) \\ \text{s. t.} \quad & \mathsf{tr}(\mathbf{X}) \succeq \mathsf{tr}(\bar{\mathbf{A}}\mathbf{X}\bar{\mathbf{A}}^{\mathrm{T}} + \bar{\mathbf{Q}}) + \varepsilon^2 \mathsf{tr}(\mathbf{\Gamma}\mathbf{\Sigma}\mathbf{\Gamma}^{\mathrm{T}}), \\ & \mathbf{X} \succeq 0, \\ & \mathsf{tr}(\mathbf{\Sigma}) \leq \eta, \\ & \mathbf{\Sigma} \succeq 0. \end{aligned} \tag{3.10}$$

The second step of the BCD approach determines the Byzantine agents by solving

$$\begin{aligned} \mathcal{P}_2: \quad \max_{\mathbf{X},\mathbf{z}}. \quad & \mathsf{tr}(\mathbf{X}) \\ \text{s. t.} \quad & \mathsf{tr}(\mathbf{X}) \succeq \mathsf{tr}(\bar{\mathbf{A}}\mathbf{X}\bar{\mathbf{A}}^{\mathrm{T}} + \bar{\mathbf{Q}}) + \varepsilon^2 \mathsf{tr}(\mathbf{\Gamma}\mathbf{\Sigma}\mathbf{\Gamma}^{\mathrm{T}}), \\ & \mathbf{\Gamma} = \mathbf{E}\ \mathsf{diag}(\mathbf{z}) \otimes \mathbf{A}, \\ & \mathbf{X} \succeq 0, \\ & \mathbf{1}^{\mathrm{T}}\mathbf{z} \leq B, \quad 0 \leq z_i \leq 1, \quad i = 1, \ldots N. \end{aligned} \tag{3.11}$$

The subproblems (3.10) and (3.11) are convex, and (3.10) has a unique solution for a given $\mathbf{z}$. In light of [85, Theorem 1], we conclude that iterating (3.10) and (3.11) for T iterations results in convergence to a stationary point. The steps in (3.10) and (3.11) reduce the problem in (3.9) to that of solving a sequence of SDPs, which can be efficiently solved by interior-point methods. The optimal solution $\mathbf{z}^* \in [0, 1]^N$ is not Boolean due to the relaxation employed in (3.11). Hence, we recover a feasible solution $\mathbf{z}'$ of (3.9) by sorting the elements of $\mathbf{z}^*$ in descending order and set $z'_i = 1$ for the agents corresponding to the B largest elements.

Backward Stepwise Selection based Attack Strategy

We also propose an improved exhaustive search-based method for finding the optimal subset of Byzantine agents to assess the performance of our proposed strategy. For a given Byzantine selection vector $\mathbf{z}$, the problem in (3.9) is an SDP. Hence, instead of relaxing the Boolean constraints, we employ an improved exhaustive search-based method to determine the set of Byzantine agents and then find the corresponding optimal covariance matrices from (3.10). To select Byzantine agents, we adopt the backward stepwise selection algorithm [86].[1] In this method, the algorithm begins by considering all agents as Byzantine, i.e., $\mathcal{B} = \mathcal{N}$, and then iteratively removes the agent that contributes least to the overall objective. The algorithm stops when only B most effective agents remain. You can find more details regarding the backward stepwise operation in **P1** and [86].

3.1.3 Numerical Results

We consider a randomly generated undirected connected network with $N = 25$ sensor agents, maximum degree of $\Delta = 11$, and consensus gain $\varepsilon = 0.08$, see Figure 3.1. The system and agent parameters are considered to be

$$\mathbf{A} = \begin{bmatrix} 0.6 & 0.005 \\ 0.25 & 0.6 \end{bmatrix},$$

and for all agents $i \in \mathcal{N}$, we have $\mathbf{Q} = 0.1\mathbf{I}_2$, $\mathbf{R}_i = \mathbf{I}_2$, and $\mathbf{H}_i = \mu_i \mathbf{I}_2$ with $\mu_i \sim \mathcal{U}(0, 1)$. We set $T = 10$ iterations for the BCD method and assume that the attack starts at $n_0 = 20$ with the stealthiness parameter $\eta = N$. The simulations are conducted using MATLAB, and the results are averaged over 200 independent experiments.

The proposed attack strategies are compared with two naive strategies, namely, random selection attack and uniform perturbation attack. The former strategy randomly selects the Byzantine agents, while the associate covariance matrices are obtained from (3.10). The latter strategy, choose the attack sequence covariance matrices as $\Sigma_j = \frac{\eta}{B}\mathbf{I}_m$ for all $j \in \mathcal{B}$ and the set of Byzantines are determined from (3.11). Essentially, the random selection attack strategy demonstrates how the attack covariance design in (3.10) affects network performance, while the uniform perturbation attack strategy depicts the impact of optimal Byzantine set selection in (3.11) on network performance.

Figure 3.2 illustrates the steady-state NMSE versus the time instant n for the different strategies. It shows that the proposed methods significantly outperform the

[1] This method is called improved exhaustive search-based strategy, as it does not check all the possible subsets of Byzantine agents and follows a backward stepwise selection algorithm that is less complex, $\mathcal{O}(\frac{N(N+1)-B(B+1)}{2})$ instead of $\mathcal{O}(2^N)$, than the exhaustive search mechanism.

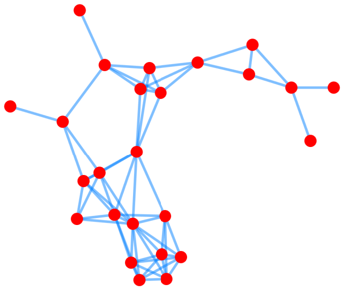


Figure 3.1: Randomly generated network topology.

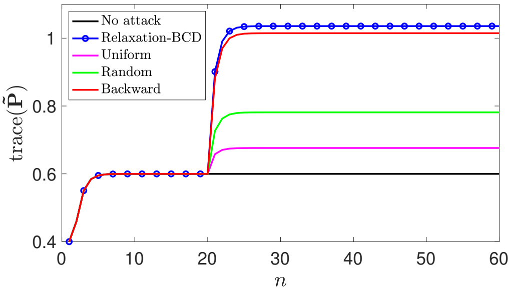


Figure 3.2: The NMSE for different attack strategies in a network of $N = 25$ and $B = 5$.

naive random and uniform attack strategies. The BCD-based approach is computationally less intensive and performs close to the exhaustive search-based method. Figure 3.2 also demonstrates that the covariance design influences the overall performance more than Byzantine agent selection. Figure 3.3 shows the NMSE versus the percentage of Byzantine agents for fixed stealthiness parameter. We observe that the joint attack strategy performs close to the backward stepwise selection-based method. When compared with random and uniform attack strategies, the BCD- and backward stepwise selection-based methods cause larger degradation in the NMSE for a fixed percentage of Byzantine agents.

3.2 Enhanced Resilience to Byzantine Attacks

This section develops a consensus-based distributed filtering algorithm that simultaneously reduces inter-agent communication load and the impact of coordinated Byzantine attacks on network performance. The partial-sharing-based approaches were originally proposed in [56, 57] as alternative solutions to reduce local communication among agents. The partial-sharing strategy allows agents to participate in distributed learning by sharing only a fraction of their information during

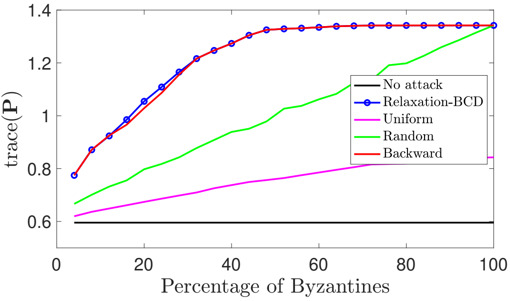


Figure 3.3: The NMSE versus percentage of Byzantines for a network with $N = 25$.

each inter-agent interaction. The simplicity of implementation and efficiency of computation make partial-sharing strategies very popular in distributed processing scenarios. To the best of our knowledge, partial-sharing-based approaches have not been investigated in an adversarial environment. As a result, there are no other approaches in the literature that can be compared with the results presented in this section. However, we benchmark our proposed method against the ideal scenario in the literature, where there are no adversaries.

3.2.1 Byzantine-Resilient Distributed Kalman Filter

By applying the partial-sharing technique to the state estimates in the CDF algorithm [7], we reduce the amount of information flowing between agents at any given instant while maintaining the advantages of cooperation. In particular, each agent only shares a fraction of its state estimate with neighbors rather than the entire state estimate vector, i.e., l entries of $\hat{\mathbf{x}}_{j,n} \in \mathbb{R}^m$ is received at each agent i, with $l \leq m$. Although partial-sharing was originally introduced to reduce communication overhead, we show that adopting this idea in the CDF setting can improve robustness to Byzantine attacks.

The entry selection process at each agent j is performed using a diagonal matrix of size $m \times m$, i.e., the selection matrix $\mathbf{S}_{j,n}$, whose main diagonal contains l ones and $m - l$ zeros. The ones on the main diagonal specify the entries of the state estimate $\hat{\mathbf{x}}_{j,n}$ that should be shared with neighbors. Selecting l entries from m can either be done stochastically, or sequentially, as in [56] and [57], respectively. Here, we use uncoordinated partial-sharing, which is a special case of stochastic partial-sharing [56]. In uncoordinated partial-sharing, each agent j is initialized with random selection matrices. The selection matrix at the current time instant, i.e., $\mathbf{S}_{j,n}$, can be obtained by performing τ right-circular shift operations on the main diagonal of the selection matrix used in the previous time instant. In other words, if $\mathbf{s}_{j,n} \in \mathbb{R}^m$ contains the main diagonal entries of $\mathbf{S}_{j,n}$ at the current instant, then

$\mathbf{s}_{j,n}$ = right-circular-shift$\{\mathbf{s}_{j,n-1}, \tau\}$. Then the selection matrix at current instant can be constructed as $\mathbf{S}_{j,n} = \mathsf{diag}\{\mathbf{s}_{j,n}\}$. This allows each agent j to share only the initial selection matrix $\mathbf{S}_{j,0}$ with its neighbors, and maintain a record of the indices of parameters shared without needing any additional mechanisms. As a result, the frequency of each entry of the state estimate being shared is equal to $p_e = \frac{l}{m}$.

Due to partial sharing, every agent only receives a fraction of the entire state estimate vector from its neighbors, i.e., $\mathbf{S}_{j,n}\hat{\mathbf{x}}_{j,n}$. Thus, the received information here must be compensated to fill in the missing entries. At each agent i, the unavailable entries from neighbors, i.e., $(\mathbf{I} - \mathbf{S}_{j,n})\hat{\mathbf{x}}_{j,n}$, is replaced with local entries as $(\mathbf{I} - \mathbf{S}_{j,n})\hat{\mathbf{x}}_{i,n}$. Subsequently, the state update at agent i is modified as

$$\hat{\mathbf{x}}_{i,n+1} = \mathbf{A}\hat{\mathbf{x}}_{i,n} + \mathbf{K}_{i,n}\big(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n}\big) + \mathbf{C}_i \sum_{j\in\mathcal{N}_i} \mathbf{S}_{j,n}\big(\hat{\mathbf{x}}_{j,n} - \hat{\mathbf{x}}_{i,n}\big), \quad (3.12)$$

where $\mathbf{C}_i \in \mathbb{R}^{m\times m}$ denotes the consensus gain. However, in the presence of Byzantine attack, as described in Section 2.4.2, every agent only receives a fraction of the perturbed state estimate vectors from its neighbors, i.e., $\mathbf{S}_{j,n}\bar{\mathbf{x}}_{j,n}$ where $\bar{\mathbf{x}}_{j,n}$ represents the received information from the neighboring agent j as in (2.16). We assume that the Byzantine attack starts once system reaches steady-state, i.e., $n = n_0$. Accordingly, for $n \geq n_0$ and at each agent i, the received information from neighboring agent j is compensated by replacing the missing entries $(\mathbf{I} - \mathbf{S}_{j,n})\bar{\mathbf{x}}_{j,n}$ with corresponding local entries $(\mathbf{I} - \mathbf{S}_{j,n})\tilde{\mathbf{x}}_{i,n}$ where $\tilde{\mathbf{x}}_{i,n}$ denotes the state estimate in the presence of the attack. Subsequently, the state estimate (3.12) is alternatively expressed as

$$\begin{aligned}\tilde{\mathbf{x}}_{i,n+1} =& \mathbf{A}\tilde{\mathbf{x}}_{i,n} + \mathbf{K}_{i,n}\big(\mathbf{y}_{i,n} - \mathbf{H}_i\tilde{\mathbf{x}}_{i,n}\big) \\ &+ \mathbf{C}_i \sum_{j\in\mathcal{N}_i} \mathbf{S}_{j,n}\big(\tilde{\mathbf{x}}_{j,n} - \tilde{\mathbf{x}}_{i,n}\big) + \mathbf{C}_i \sum_{j\in\mathcal{N}_i} \mathbf{S}_{j,n}\boldsymbol{\delta}_{j,n}.\end{aligned} \quad (3.13)$$

The Kalman gain can be obtained by minimizing the trace of the estimation error covariance $\tilde{\mathbf{P}}_{i,n} \triangleq \mathbb{E}\{\tilde{\mathbf{e}}_{i,n}\tilde{\mathbf{e}}_{i,n}^{\mathrm{T}}\}$ with the estimation error evolving as

$$\begin{aligned}\tilde{\mathbf{e}}_{i,n+1} &= \tilde{\mathbf{x}}_{i,n+1} - \mathbf{x}_{n+1} \\ &= \mathbf{F}_{i,n}\tilde{\mathbf{e}}_{i,n} + \mathbf{C}_i \sum_{j\in\mathcal{N}_i} \mathbf{S}_{j,n}\tilde{\mathbf{e}}_{j,n} + \mathbf{K}_{i,n}\mathbf{v}_{i,n} - \mathbf{w}_n + \mathbf{C}_i \sum_{j\in\mathcal{N}_i} \mathbf{S}_{j,n}\boldsymbol{\delta}_{j,n}\end{aligned} \quad (3.14)$$

where

$$\mathbf{F}_{i,n} = \mathbf{A} - \mathbf{K}_{i,n}\mathbf{H}_i - \mathbf{C}_i \sum_{j\in\mathcal{N}_i} \mathbf{S}_{j,n}. \quad (3.15)$$

As a result, the error covariance matrix at each agent i is derived as

$$\begin{aligned}\tilde{\mathbf{P}}_{i,n+1} =& \mathbf{F}_{i,n}\tilde{\mathbf{P}}_{i,n}\mathbf{F}_{i,n}^{\mathrm{T}} + \mathbf{K}_{i,n}\mathbf{R}_i\mathbf{K}_{i,n}^{\mathrm{T}} + \mathbf{Q} + \Delta\tilde{\mathbf{P}}_{i,n} \\ &+ \mathbf{C}_i \sum_{s\in\mathcal{N}_i}\sum_{p\in\mathcal{N}_i} \mathbf{S}_{s,n}\boldsymbol{\Sigma}_{sp}\mathbf{S}_{p,n}^{\mathrm{T}}\mathbf{C}_i^{\mathrm{T}}\end{aligned} \tag{3.16}$$

where $\boldsymbol{\Sigma}_{sp} = \mathbb{E}\{\boldsymbol{\delta}_{s,n}\boldsymbol{\delta}_{p,n}^{\mathrm{T}}\}$ and

$$\begin{aligned}\Delta\tilde{\mathbf{P}}_{i,n} =& \mathbf{F}_{i,n}\sum_{j\in\mathcal{N}_i}\tilde{\mathbf{P}}_{ij,n}\mathbf{S}_{j,n}^{\mathrm{T}}\mathbf{C}_i^{\mathrm{T}} + \mathbf{C}_i\sum_{j\in\mathcal{N}_i}\mathbf{S}_{j,n}\tilde{\mathbf{P}}_{ji,n}\mathbf{F}_{i,n}^{\mathrm{T}} \\ &+ \mathbf{C}_i\sum_{s\in\mathcal{N}_i}\sum_{p\in\mathcal{N}_i}\mathbf{S}_{s,n}\tilde{\mathbf{P}}_{sp,n}\mathbf{S}_{p,n}^{\mathrm{T}}\mathbf{C}_i^{\mathrm{T}}.\end{aligned}$$

Similarly, cross-terms of the error covariance, i.e., $\tilde{\mathbf{P}}_{ij,n} \triangleq \mathbb{E}\{\mathbf{e}_{i,n}\mathbf{e}_{j,n}^{\mathrm{T}}\}$, evolve as

$$\tilde{\mathbf{P}}_{ij,n+1} = \mathbf{F}_{i,n}\tilde{\mathbf{P}}_{ij,n}\mathbf{F}_{j,n}^{\mathrm{T}} + \mathbf{Q} + \Delta\tilde{\mathbf{P}}_{ij,n} + \mathbf{C}_i\sum_{s\in\mathcal{N}_i}\sum_{p\in\mathcal{N}_j}\mathbf{S}_{s,n}\boldsymbol{\Sigma}_{sp}\mathbf{S}_{p,n}^{\mathrm{T}}\mathbf{C}_j^{\mathrm{T}}$$

with

$$\begin{aligned}\Delta\tilde{\mathbf{P}}_{ij,n} =& \mathbf{F}_{i,n}\sum_{p\in\mathcal{N}_j}\tilde{\mathbf{P}}_{ip,n}\mathbf{S}_{p,n}^{\mathrm{T}}\mathbf{C}_j^{\mathrm{T}} + \mathbf{C}_i\sum_{s\in\mathcal{N}_i}\mathbf{S}_{s,n}\tilde{\mathbf{P}}_{sj,n}\mathbf{F}_{j,n}^{\mathrm{T}} \\ &+ \mathbf{C}_i\sum_{s\in\mathcal{N}_i}\sum_{p\in\mathcal{N}_j}\mathbf{S}_{s,n}\tilde{\mathbf{P}}_{sp,n}\mathbf{S}_{p,n}^{\mathrm{T}}\mathbf{C}_j^{\mathrm{T}}.\end{aligned}$$

Differentiating the trace of (3.16) with respect to $\mathbf{K}_{i,n}$ gives

$$\mathbf{K}_{i,n}^{*} = \big((\mathbf{A} - \mathbf{C}_i\sum_{j\in\mathcal{N}_i}\mathbf{S}_{j,n})\tilde{\mathbf{P}}_{i,n} + \mathbf{C}_i\sum_{j\in\mathcal{N}_i}\mathbf{S}_{j,n}\tilde{\mathbf{P}}_{ji,n}\big)\mathbf{H}_i^{\mathrm{T}}\mathbf{M}_{i,n}^{-1}$$

with $\mathbf{M}_{i,n} = \mathbf{R}_i + \mathbf{H}_i\tilde{\mathbf{P}}_{i,n}\mathbf{H}_i^{\mathrm{T}}$.

We see that the local covariance update in (3.16) requires access to cross-terms of the error covariance, resulting in considerable communication overhead. To reduce the communication overhead, for sufficiently small gain values, i.e., $\mathbf{C}_i$, we can ignore the term $\Delta\tilde{\mathbf{P}}_{i,n}$ in (3.16) and the last term of $\mathbf{F}_{i,n}$ in (3.15) [7], i.e., we have

$$\tilde{\mathbf{P}}_{i,n+1} = \hat{\mathbf{F}}_{i,n}\tilde{\mathbf{P}}_{i,n}\hat{\mathbf{F}}_{i,n}^{\mathrm{T}} + \mathbf{K}_{i,n}\mathbf{R}_i\mathbf{K}_{i,n}^{\mathrm{T}} + \mathbf{Q} + \mathbf{C}_i\sum_{s\in\mathcal{N}_i}\sum_{p\in\mathcal{N}_i}\mathbf{S}_{s,n}\boldsymbol{\Sigma}_{sp}\mathbf{S}_{p,n}^{\mathrm{T}}\mathbf{C}_i^{\mathrm{T}} \tag{3.17}$$

Algorithm 2 BR-CDF algorithm

```
For each agent $i \in \mathcal{N}$
Initialize: $\hat{\mathbf{x}}_{i,0} = \mathbf{x}_0$, $\mathbf{P}_{i,0} = \mathbf{P}_0$, and share $\mathbf{S}_{i,0} = \mathsf{diag}(\mathbf{s}_{i,0})$ with all $j \in \mathcal{N}_i$
for all $k > 0$ do
    For all $j \in \mathcal{N}_i$ receive $\{\mathbf{S}_{j,n}\bar{\mathbf{x}}_{j,n}\}$
    $\mathbf{K}_{i,n} = \mathbf{A}\mathbf{P}_{i,n}\mathbf{H}_i^{\mathrm{T}}\left(\mathbf{R}_i + \mathbf{H}_i\mathbf{P}_{i,n}\mathbf{H}_i^{\mathrm{T}}\right)^{-1}$
    Update the state estimate
        $\hat{\mathbf{x}}_{i,n+1} = \mathbf{A}\hat{\mathbf{x}}_{i,n} + \mathbf{K}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n}\right) + \mathbf{C}_i\sum_{j\in\mathcal{N}_i}\left(\mathbf{S}_{j,n}\bar{\mathbf{x}}_{j,n} - \mathbf{S}_{j,n}\hat{\mathbf{x}}_{i,n}\right)$
    $\hat{\mathbf{F}}_{i,n} = \mathbf{A} - \mathbf{K}_{i,n}\mathbf{H}_i$
    Update the error covariance: $\mathbf{P}_{i,n+1} = \hat{\mathbf{F}}_{i,n}\mathbf{P}_{i,n}\hat{\mathbf{F}}_{i,n}^{\mathrm{T}} + \mathbf{K}_{i,n}\mathbf{R}_i\mathbf{K}_{i,n}^{\mathrm{T}} + \mathbf{Q}$
    $\mathbf{s}_{i,n+1} = $ right-circular-shift$\{\mathbf{s}_{i,n}, \tau\}$
    $\mathbf{S}_{i,n+1} = \mathsf{diag}\left(\mathbf{s}_{i,n+1}\right)$
    Share $\mathbf{S}_{i,n+1}\bar{\mathbf{x}}_{i,n+1}$ with all $j \in \mathcal{N}_i$
end for
```

with $\hat{\mathbf{F}}_{i,n} = \mathbf{A} - \mathbf{K}_{i,n}\mathbf{H}_i$. Accordingly, the optimal Kalman gain reduces to

$$\mathbf{K}_{i,n} = \mathbf{A}\mathbf{P}_{i,n}\mathbf{H}_i^{\mathrm{T}}\left(\mathbf{R}_i + \mathbf{H}_i\mathbf{P}_{i,n}\mathbf{H}_i^{\mathrm{T}}\right)^{-1}\cdot \tag{3.18}$$

With the above approximations, we obtain a distributed consensus-based Kalman filter, albeit suboptimal [2, 7], that only requires local variables in the error covariance update at each agent. It is worth noting that the last term in (3.17) is only used to characterize the impact of the perturbation covariances, and since the attack is stealthy from the perspective of an agent, it is excluded from the filtering algorithm. As a result, in addition to the initial selection matrix $\mathbf{S}_{j,0}$, at each time n, agent j shares a fraction of the perturbed state estimate, i.e., $\mathbf{S}_{j,n}\bar{\mathbf{x}}_{j,n}$, with its neighbors. The proposed BR-CDF algorithm is summarized in Algorithm 2.

3.2.2 Stability and Performance Analysis

This section provides the stability analysis of the BR-CDF in Algorithm 2. In order to develop the ensuing analysis, we make the following assumption:

Assumption: *For all $i \in \mathcal{N}$, the selection matrix $\mathbf{S}_{i,n}$ is independent of any other data and the selection matrices $\mathbf{S}_{j,s}$ for all $i \neq j$ and $n \neq s$.*

Our main result on the stability of the BR-CDF algorithm is summarized by the following theorem.

Theorem 3.1. Consider the BR-CDF in Algorithm 2 with consensus gain $\mathbf{C}_i =$

$\gamma\mathbf{A}\left(\mathbf{P}_{i,n}^{-1}+\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i\right)^{-1}$ and a sufficiently small γ satisfying

$$\gamma \leq \gamma^* = \frac{1}{\sqrt{p_e}}\left(\frac{\lambda_{\min}(\mathbf{\Lambda}_{\mathrm{I}})}{\lambda_{\max}((\mathbf{L}\otimes\mathbf{I})\mathbf{\Lambda}_{\mathrm{II}}(\mathbf{L}\otimes\mathbf{I}))}\right)^{\frac{1}{2}}$$

where $p_e = \frac{l}{m}$, $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ return the maximum and minimum eigenvalues of the argument matrix, respectively, and

$$\begin{aligned}\mathbf{\Lambda}_{\mathrm{I}} &= \mathsf{diag}(\{\left(\mathbf{P}_i + (\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i)^{-1}\right)^{-1}\}_{i=1}^{N}),\\ \mathbf{\Lambda}_{\mathrm{II}} &= \mathsf{diag}(\{\left(\mathbf{P}_i^{-1} + \mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i\right)^{-1}\}_{i=1}^{N}),\end{aligned}$$

with $\mathbf{P}_i = \lim_{n\to\infty}\mathbf{P}_{i,n}$. Then, the error dynamics of the BR-CDF algorithm is globally asymptotically stable and all local estimators asymptotically reach a consensus on state estimates, i.e., $\hat{\mathbf{x}}_{1,n} = \hat{\mathbf{x}}_{2,n} = \cdots = \hat{\mathbf{x}}_{N,n} = \mathbf{x}_n$.

Proof. The detailed proof is given in **P2**. □

3.2.3 Resilience to the Byzantine Attack

This section investigates the robustness of the solution for the BR-CDF in Algorithm 2 in the presence of a data falsification attack. We assume that Byzantine agents start perturbing the information once the network reaches steady-state, i.e., $n = n_0 > 0$. We further assume that the attack remains stealthy from the perspective of agents; thus, the consensus gain $\mathbf{C}_i$ remains fixed for $n \geq n_0$.

In steady-state, and after applying statistical expectation $\mathbb{E}\{\cdot\}$ with respect to selection matrices, the error covariance matrix in (3.17) satisfies

$$\tilde{\mathbf{P}}_i = \hat{\mathbf{F}}_i\tilde{\mathbf{P}}_i\hat{\mathbf{F}}_i^{\mathrm{T}} + \mathbf{K}_i\mathbf{R}_i\mathbf{K}_i^{\mathrm{T}} + \mathbf{Q} + \mathbf{C}_i\sum_{s\in\mathcal{N}_i}\sum_{p\in\mathcal{N}_i}\mathbb{E}\left\{\mathbf{S}_{s,n}\mathbf{\Sigma}_{sp}\mathbf{S}_{p,n}^{\mathrm{T}}\right\}\mathbf{C}_i^{\mathrm{T}} \tag{3.19}$$

where $\tilde{\mathbf{P}}_i = \lim_{n\to\infty}\tilde{\mathbf{P}}_{i,n}$. Defining

$$\begin{aligned}\tilde{\mathbf{P}} &\triangleq \mathsf{Blockdiag}(\{\tilde{\mathbf{P}}_i\}_{i=1}^{N})\\ \hat{\mathbf{F}} &\triangleq \mathsf{Blockdiag}(\{\hat{\mathbf{F}}_i\}_{i=1}^{N})\\ \mathbf{K} &\triangleq \mathsf{Blockdiag}(\{\mathbf{K}_i\}_{i=1}^{N})\\ \mathbf{C} &\triangleq \mathsf{Blockdiag}(\{\mathbf{C}_i\}_{i=1}^{N})\\ \mathbf{R} &\triangleq \mathsf{Blockdiag}(\{\mathbf{R}_i\}_{i=1}^{N})\end{aligned}$$

the network-wide version of (3.19) can be stated as

$$\begin{aligned}\tilde{\mathbf{P}} =&\hat{\mathbf{F}}\tilde{\mathbf{P}}\hat{\mathbf{F}}^{\mathrm{T}} + \mathbf{K}\mathbf{R}\mathbf{K}^{\mathrm{T}} + \mathbf{I}_N \otimes \mathbf{Q} \\ &+ \mathbf{C}\mathbb{E}\left\{(\mathbf{I}_N \otimes \mathbf{I}) \odot \big((\mathbf{E} \otimes \mathbf{I})\mathbf{S}_n\mathbf{\Sigma}\mathbf{S}_n^{\mathrm{T}}(\mathbf{E} \otimes \mathbf{I})\big)\right\}\mathbf{C}^{\mathrm{T}}\end{aligned} \tag{3.20}$$

where $\mathbf{S}_n \triangleq \mathsf{Blockdiag}(\{\mathbf{S}_{i,n}\}_{i=1}^N)$. Under **Assumption 1**, we have

$$\begin{aligned}\tilde{\mathbf{P}} =&\hat{\mathbf{F}}\tilde{\mathbf{P}}\hat{\mathbf{F}}^{\mathrm{T}} + \mathbf{K}\mathbf{R}\mathbf{K}^{\mathrm{T}} + \mathbf{I}_N \otimes \mathbf{Q} \\ &+ \mathbf{C}\big((\mathbf{I}_N \otimes \mathbf{I}) \odot \big((\mathbf{E} \otimes \mathbf{I})\mathbb{E}\left\{\mathbf{S}_n\mathbf{\Sigma}\mathbf{S}_n^{\mathrm{T}}\right\}(\mathbf{E} \otimes \mathbf{I})\big)\big)\mathbf{C}^{\mathrm{T}}\end{aligned} \tag{3.21}$$

where the expectation term can be simplified as

$$\begin{aligned}\mathbb{E}\{\mathbf{S}_n\mathbf{\Sigma}\mathbf{S}_n^{\mathrm{T}}\} &= \mathbb{E}\{\mathrm{vec}^{-1}\left(\mathrm{vec}\left(\mathbf{S}_n\mathbf{\Sigma}\mathbf{S}_n^{\mathrm{T}}\right)\right)\} \\ &= \mathbb{E}\{\mathrm{vec}^{-1}\left((\mathbf{S}_n \otimes \mathbf{S}_n)\,\mathrm{vec}\,(\mathbf{\Sigma})\right)\} \\ &= \mathrm{vec}^{-1}\left(\mathbb{E}\{(\mathbf{S}_n \otimes \mathbf{S}_n)\}\mathrm{vec}\,(\mathbf{\Sigma})\right)\end{aligned}$$

Following the approach in [56, Appendix B], we can show that $\mathbb{E}\{\mathbf{S}_n \otimes \mathbf{S}_n\} \leq p_e(\mathbf{I} \otimes \mathbf{I})$ with $0 < p_e \leq 1$, [2] and we have

$$\mathbb{E}\{\mathbf{S}_n\mathbf{\Sigma}\mathbf{S}_n^{\mathrm{T}}\} \leq p_e\mathrm{vec}^{-1}\left(\mathrm{vec}\,(\mathbf{\Sigma})\right) = p_e\mathbf{\Sigma}\cdot \tag{3.22}$$

Using the result of (3.22) and knowing that $\mathbf{P}$ is a positive definite matrix, we finally have

$$\tilde{\mathbf{P}} \leq \hat{\mathbf{F}}\tilde{\mathbf{P}}\hat{\mathbf{F}}^{\mathrm{T}}+\mathbf{K}\mathbf{R}\mathbf{K}^{\mathrm{T}}+\mathbf{I}_N\otimes\mathbf{Q}+p_e\mathbf{C}\big((\mathbf{I}_N\otimes\mathbf{I})\odot\big((\mathbf{E}\otimes\mathbf{I})\mathbf{\Sigma}(\mathbf{E}\otimes\mathbf{I})\big)\big)\mathbf{C}^{\mathrm{T}} \tag{3.23}$$

The last term in (3.23) describes the impact of the coordinated Byzantine attack on the error covariance matrix that is scaled by the selection parameter p_e. Thus, similar to Section 3.1.2, we use

$$\mathrm{NMSE} \sim \lim_{n\to\infty} \mathrm{tr}(\mathbb{E}\{\tilde{\mathbf{P}}_n\}) \tag{3.24}$$

as a proxy to capture the NMSE. We see that partial sharing of information, i.e., $p_e < 1$, results in lower steady-state NMSE compared to full information sharing, i.e., $p_e = 1$, that indicates robustness to coordinated Byzantine attacks.

3.2.4 Coordinated Byzantine Attack Design

To analyze the worst-case performance of the BR-CDF algorithm, we consider a scenario where Byzantine agents design a coordinated attack to maximize the NMSE. Based on the attack model in (2.16) and the error covariance in (3.16), Byzantine agents have the following two levers to design their coordinated attack:

[2] $\mathbf{A} \leq \mathbf{B}$ denotes an element-wise inequality for corresponding elements in $\mathbf{A}$ and $\mathbf{B}$.

- The design of perturbation covariance matrices, modeled as the covariance of zero-mean Gaussian sequences.
- The choice of selection matrices that impacts the sequence of information fractions that Byzantine agents share at the beginning of the attack.

We ensure that the attack remains stealthy from the perspective of regular agents by setting an upper bound on the energy of the perturbation sequences, i.e., $\text{tr}(\mathbf{\Sigma}) \leq \eta$. Assuming Byzantines start perturbing information once agents reach steady-state, i.e., $n = n_0$, we derive an expression for the network-wide steady-state MSE of the estimator in (3.13). The network-wide evolution of the estimation error of the BR-CDF algorithm, given in (3.14), is stated as

$$\tilde{\mathbf{e}}_{n+1} = \tilde{\mathbf{A}}_n \tilde{\mathbf{e}}_n + \tilde{\mathbf{b}}_n + \mathbf{\Gamma}_n \boldsymbol{\delta}_n \tag{3.25}$$

where $\mathbf{\Gamma}_n = \mathbf{C}\big(\mathbf{E} \otimes \mathbf{I}\big)\mathbf{S}_n$,

$$\begin{aligned}\tilde{\mathbf{A}}_n &= \mathsf{Blockdiag}(\{\mathbf{F}_{i,n}\}_{i=1}^{N}) + \mathbf{C}\big(\mathbf{E} \otimes \mathbf{I}\big)\mathbf{S}_n,\\ \tilde{\mathbf{b}}_n &= \mathsf{diag}(\{\mathbf{K}_{i,n}\mathbf{v}_{i,n}\}_{i=1}^{N}) - \mathbf{1}_N \otimes \mathbf{w}_n.\end{aligned}$$

As a result, the network-wide error covariance matrix, unlike (3.20) including cross-terms of the error covariance, is given by

$$\tilde{\mathbf{P}}_{n+1} = \tilde{\mathbf{A}}_n \tilde{\mathbf{P}}_n \tilde{\mathbf{A}}_n^{\mathrm{T}} + \tilde{\mathbf{Q}}_n + \mathbf{\Gamma}_n \mathbf{\Sigma} \mathbf{\Gamma}_n^{\mathrm{T}} \tag{3.26}$$

where $\tilde{\mathbf{Q}}_n = \mathsf{Blockdiag}(\{\mathbf{K}_{i,n}\mathbf{R}_i\mathbf{K}_{i,n}^{\mathrm{T}}\}_{i=1}^{N} + \mathbf{1}_N\mathbf{1}_N^{\mathrm{T}} \otimes \mathbf{Q}$. In (3.26), the last term is due to the injected noise and is given by

$$\mathbf{\Gamma}_n \mathbf{\Sigma} \mathbf{\Gamma}_n^{\mathrm{T}} = \mathbf{C}\big(\mathbf{E} \otimes \mathbf{I}\big)\mathbf{S}_n \mathbf{\Sigma} \mathbf{S}_n^{\mathrm{T}}\big(\mathbf{E} \otimes \mathbf{I}\big)\mathbf{C}^{\mathrm{T}} \tag{3.27}$$

which, compared to the Byzantine-free case, increases the NMSE. Considering the NMSE in (3.24), we define two optimization problems to find the optimal coordinated Byzantine attacks by designing the partial-sharing selection matrices at $n = n_0$ and attack covariance matrices of Byzantine agents.

Optimizing the attack begins by stating that maximizing the trace of the estimation error covariance in (3.26) is equivalent to maximizing the trace of its last term [87], since it is the only term that depends on the attack. The last term of the error covariance $\mathbf{P}_n$ in (3.26) depends on the selection matrix $\mathbf{S}_n$ and given the attack covariance $\mathbf{\Sigma}$, we can show that

$$\begin{aligned}\text{tr}(\mathbf{\Gamma}_n \mathbf{\Sigma} \mathbf{\Gamma}_n^{\mathrm{T}}) &= \text{tr}\big(\mathbf{C}(\mathbf{E} \otimes \mathbf{I})\mathbf{S}_n \mathbf{\Sigma} \mathbf{S}_n^{\mathrm{T}}(\mathbf{E} \otimes \mathbf{I})\mathbf{C}^{\mathrm{T}}\big)\\ &= \text{tr}\big((\mathbf{E} \otimes \mathbf{I})\mathbf{C}^{\mathrm{T}}\mathbf{C}(\mathbf{E} \otimes \mathbf{I})\mathbf{S}_n \mathbf{\Sigma} \mathbf{S}_n^{\mathrm{T}}\big)\\ &= \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}} \text{tr}\left(\mathbf{U}_{ij}\mathbf{S}_{j,n}\mathbf{\Sigma}_{ji}\mathbf{S}_{i,n}\right)\end{aligned} \tag{3.28}$$

where $\mathbf{U}_{ij} = \sum_{i\in\mathcal{N}_i}\sum_{j\in\mathcal{N}_i}\mathbf{C}_i^{\mathrm{T}}\mathbf{C}_j$. Thus, the optimization problem that maximizes the steady-state NMSE can be stated as

$$\begin{aligned}\max_{\{\mathbf{S}_i',\, i\in\mathcal{B}\}} & \sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}} \mathrm{tr}\left(\mathbf{U}_{ij}\mathbf{S}_j'\boldsymbol{\Sigma}_{ji}\mathbf{S}_i'\right) \\ \text{s. t.}\ & \mathbf{0}\le\mathbf{S}_i'\le\mathbf{I}\quad\forall i\in\mathcal{B} \\ & [\mathbf{S}_i']_{rs}\in\{0,1\} \\ & \mathrm{tr}(\mathbf{S}_i')\le l\quad\forall i\in\mathcal{B}\end{aligned}\tag{3.29}$$

where the resulting solution for $\mathbf{S}_i'$ determines the $\mathbf{S}_i(n_0)$ and the first two constraints restrict the selection matrix to be diagonal with 0 or 1 elements on the main diagonal. The last constraint enforces that only l elements of the state vector are shared with neighbors at each given instant. We relax the non-convex Boolean constraint on the elements of $\mathbf{S}_i'$ and rewrite the optimization problem as

$$\begin{aligned}\max_{\{\mathbf{S}_i',\, i\in\mathcal{B}\}} & \sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}} \mathrm{tr}\left(\mathbf{U}_{ij}\mathbf{S}_j'\boldsymbol{\Sigma}_{ji}\mathbf{S}_i'\right) \\ \text{s. t.}\ & \mathbf{0}\le\mathbf{S}_i'\le\mathbf{I}\quad\forall i\in\mathcal{B} \\ & \mathrm{tr}(\mathbf{S}_i')\le l\quad\forall i\in\mathcal{B}\end{aligned}\tag{3.30}$$

The objective function in (3.30) can be further simplified as

$$\begin{aligned}\sum_{i\in\mathcal{B}}\sum_{j\in\mathcal{B}} \mathrm{tr}\left(\mathbf{U}_{ij}\mathbf{S}_j'\boldsymbol{\Sigma}_{ji}\mathbf{S}_i'\right) = \sum_{i\in\mathcal{B}} \Bigg(& \mathrm{tr}\left(\mathbf{U}_{ii}\mathbf{S}_i'\boldsymbol{\Sigma}_i\mathbf{S}_i'\right) \\ & + \sum_{j\in\mathcal{B}/\{i\}} \frac{1}{2}\mathrm{tr}\left(\mathbf{U}_{ij}\mathbf{S}_j'\boldsymbol{\Sigma}_{ji}\mathbf{S}_i' + \mathbf{U}_{ji}\mathbf{S}_i'\boldsymbol{\Sigma}_{ij}\mathbf{S}_j'\right)\Bigg)\end{aligned}\tag{3.31}$$

which still contains non-convex quadratic terms. To overcome this problem, we employ the BCD algorithm where each Byzantine agent i, given the selection matrix of other Byzantines, optimizes its own selection matrix. The BCD algorithm is iterated for T iterations and at each iteration, $t+1$, agent i employs the selection matrix of other Byzantine agents from the previous iteration, i.e. $\{\mathbf{S}_{j,t}'\}_{j\in\mathcal{B}\setminus\{i\}}$.

Hence, the optimization problem in (3.30) can be solved by employing the BCD method, where at each agent $i\in\mathcal{B}$ and BCD iteration $t+1$, the optimization problem is modeled as

$$\begin{aligned}\mathcal{P}:\quad \max_{\mathbf{S}_i'}\quad & f\left(\mathbf{S}_i', \{\mathbf{S}_{j,t}'\}_{j\in\mathcal{B}\setminus\{i\}}\right) \\ \text{s. t.}\quad & \mathbf{0}\le\mathbf{S}_i'\le\mathbf{I} \\ & \mathrm{tr}(\mathbf{S}_i')\le l\end{aligned}\tag{3.32}$$

Algorithm 3 BCD-based attack design

```
For  each agent $i \in \mathcal{B}$
Receive  $\mathbf{S}'_{j,0} = \mathbf{S}_{j,n_0}$ from $j \in \mathcal{B}\backslash\{i\}$
Share  $\mathbf{S}'_{i,0} = \mathbf{S}_{i,n_0}$ with $j \in \mathcal{B}\backslash\{i\}$
for $t = 1$ to T do
    Find $\mathbf{S}'_i$ by solving $\mathcal{P}$ in (3.32)
    Set $\mathbf{S}_{i,t} = \mathbf{S}'_i$ and share with $j \in \mathcal{B}\backslash\{i\}$
    Receive $\{\mathbf{S}'_{j,t}\}_{j\in\mathcal{B}\backslash\{i\}}$
end for
For the main diagonal of $\mathbf{S}'_{i,T}$, set the $l$ largest element to 1 and others to 0.
Set $\mathbf{S}_{i,n_0} = \mathbf{S}'_{i,T}$
```

with $\mathbf{S}'_{j,t}$ as the selection matrix of Byzantine agent j at the former BCD iteration and the objective function

$$\begin{aligned} f\left(\mathbf{S}'_i, \{\mathbf{S}'_{j,t}\}_{j\in\mathcal{B}\backslash\{i\}}\right) =& \text{tr}\left(\mathbf{U}_{ii}\mathbf{S}'_i\boldsymbol{\Sigma}_i\mathbf{S}'_i\right) \\ &+\sum_{j\in\mathcal{B}/\{i\}} \frac{1}{2}\text{tr}\left(\mathbf{U}_{ij}\mathbf{S}'_{j,t}\boldsymbol{\Sigma}_{ji}\mathbf{S}'_i + \mathbf{U}_{ji}\mathbf{S}'_i\boldsymbol{\Sigma}_{ij}\mathbf{S}'_{j,t}\right) \cdot \end{aligned} \tag{3.33}$$

Algorithm 3 summarizes the BCD algorithm used to solve the optimization problem in (3.32). Next, we investigate how optimizing the perturbation covariance matrix impacts the NMSE.

Given the selection matrices at the beginning of the attack, i.e., $\mathbf{S}_{i,n_0}$ for $i \in \mathcal{N}$, Byzantine agents can maximize the steady-state NMSE by cooperatively designing their attack covariances in the following optimization problem

$$\begin{aligned} \max_{\boldsymbol{\Sigma}} \quad & \text{tr}(\boldsymbol{\Gamma}_{n_0}\boldsymbol{\Sigma}\boldsymbol{\Gamma}_{n_0}^{\mathrm{T}}) \\ \text{s. t.} \quad & \boldsymbol{\Sigma} \succcurlyeq 0 \\ & \text{tr}(\boldsymbol{\Sigma}) \leq \eta \end{aligned} \tag{3.34}$$

where $\boldsymbol{\Gamma}_{n_0} = \mathbf{C}\big(\mathbf{E}\otimes\mathbf{I}\big)\mathbf{S}_{n_0}\big(\,\mathsf{diag}(\mathbf{z})\otimes\mathbf{I}\big)$. The first constraint in (3.34) guarantees that the designed attack covariance is positive semidefinite and the last constraint is related to stealthiness. The energy of the Byzantine noise sequences is assumed to be limited as $\text{tr}(\boldsymbol{\Sigma}) \leq \eta$ to maintain the attack stealthiness.

Remark 3.1. The optimization problem in (3.34) is a semidefinite programming (SDP) problem that can be efficiently solved by interior-point methods.

3.2.5 Numerical Results

To illustrate the robustness of the BR-CDF algorithm to Byzantine attack, we examined a target tracking system with the state vector length of $m = 8$ and described by a linear model

$$\mathbf{x}_{n+1} = \left(\begin{bmatrix} 0.6 & 0.005 \\ 0.25 & 0.6 \end{bmatrix} \otimes \mathbf{I}_4\right) \mathbf{x}_n + \mathbf{w}_n.$$

We considered the same network as in Figure 3.1 where at each agent i, the state noise covariance is $\mathbf{Q} = 0.1\mathbf{I}$ and the local observation is given by

$$\mathbf{y}_{i,n} = \left(\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \otimes \mathbf{I}_2\right) \mathbf{x}_n + \mathbf{v}_{i,n}.$$

In addition, at each agent i, we considered the observation noise covariance as $\mathbf{R}_i = \mu_i \mathbf{I}$, where $\mu_i \sim \mathcal{U}(0,1)$. The average NMSE is considered as a performance metric that is defined as

$$\text{MSE} \triangleq \frac{1}{N} \textstyle\sum_{i=1}^{N} \text{tr}(\tilde{\mathbf{P}}_i) \tag{3.35}$$

with $\tilde{\mathbf{P}}_i$ as the steady-state error covariance matrix of agent i in (3.19). The simulations are conducted using MATLAB, and the results are averaged over 200 independent experiments.

We simulated the BR-CDF algorithm for different values of l, e.g., 2, 4, 6, and 8 (i.e., 25%, 50%, 75% and full information sharing). The 25%-sharing, for example, means that, at each time, we only share $l = 2$ elements from the state estimate. Figure 3.4 shows MSE versus time n, when no attacks occur in the network. It shows that the performance degradation is inversely proportional to the amount of shared information. Although sharing a smaller fraction of information results in higher MSE, the difference is negligible in this experiment.

Next, we examined the robustness of the BR-CDF algorithm in the presence of Byzantine attack. After the network has reached convergence, Byzantine agents launch an attack at $n_0 = 30$. The Byzantine agents are chosen as the $B = 5$ nodes with the highest degree in the network graph and the energy of the attack sequences is restricted with parameter $\eta = N$. We then compared the accuracy of the proposed suboptimal BR-CDF in Algorithm 2 to the solution of the BR-CDF that shares all necessary variables. Fig. 3.5 illustrates MSE versus time index n for different values of l. We observe that the suboptimal solution performs closely

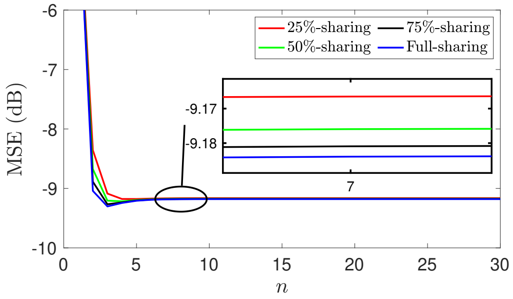


Figure 3.4: MSE of the BR-CDF algorithm versus time index n without attack.

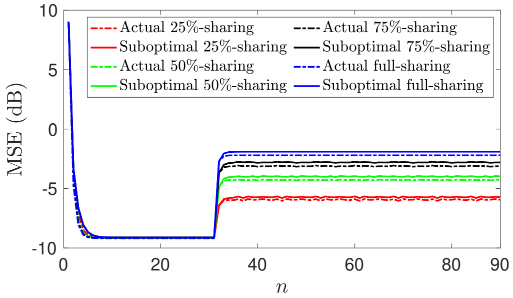


Figure 3.5: MSE of the BR-CDF algorithm and its suboptimal solution versus time index n.

to the solution that shares all necessary variables. Furthermore, the proposed algorithms provide robustness to Byzantine attacks since sharing less information results in lower MSE.

In Fig. 3.6, in order to observe the fluctuation caused by the selection matrices, we plot the MSE in (3.35) and $\text{MSE}' = \frac{1}{N}\lim_{n\to\infty}\sum_{i=1}^{N}\text{tr}(\tilde{\mathbf{P}}_{i,n})$ with $\tilde{\mathbf{P}}_{i,n}$ in (3.17). [3] Thus, we can examine the accuracy of our theoretical finding to compute the expected value of the error covariance with respect to the selection matrices in (3.19). The close performance of the MSE and MSE$'$ in Fig. 3.6 demonstrates that simulation results match theoretical findings.

To solve the optimization problem $\mathcal{P}$ in (3.32), we performed the simulation by the BCD algorithm with $T = 10$ iterations and designed the selection matrices $\{\mathbf{S}_{j,n_0}\}_{j\in\mathcal{B}}$ at $n_0 = 30$. We can see from Fig. 3.7 that the designed selection matrices deteriorate the network MSE. Also, it can be seen that designing the se-

[3] The difference between MSE$'$ and MSE is that the MSE$'$ does not include the statistical expectation with respect to the selection matrices.

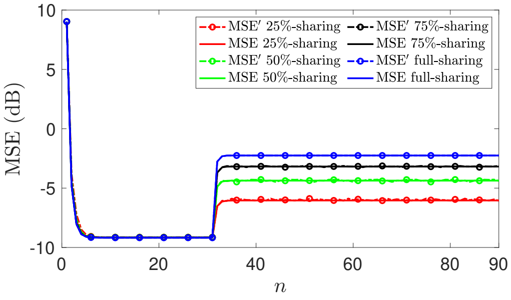


Figure 3.6: MSE and MSE′ versus time index n.

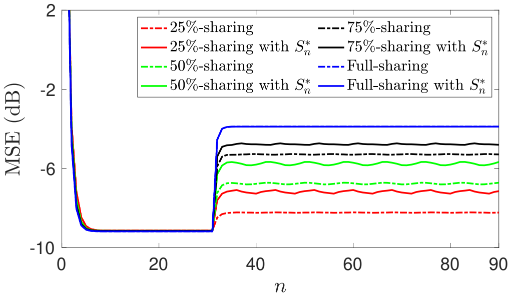


Figure 3.7: MSE versus time index n for optimized selection matrix $\mathbf{S}_n^*$ and random selection matrix $\mathbf{S}_n$.

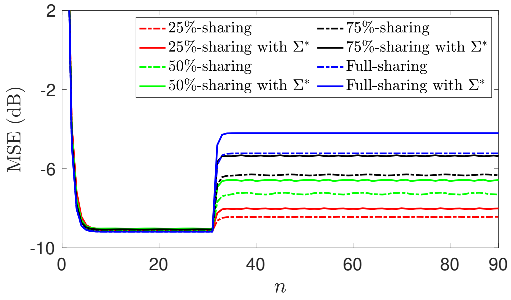


Figure 3.8: MSE versus time index n for optimized attack covariance $\mathbf{\Sigma}^*$ and random attack covariance $\mathbf{\Sigma}$.

lection matrices has a higher impact on degrading the network performance when smaller fraction of the information is shared.

By solving the optimization problem in (3.34), we examine the impact of optimiz-

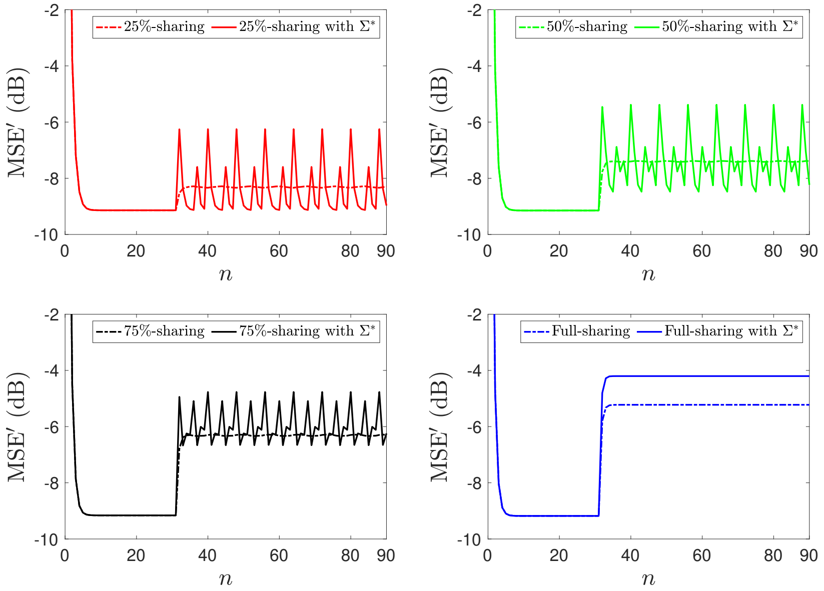


Figure 3.9: MSE$'$ versus time index n for optimized attack covariance $\mathbf{\Sigma}^*$ and random attack covariance $\mathbf{\Sigma}$.

ing the attack covariance on the MSE in comparison to random attack covariance selection. To this end, we fixed the constraint on the energy of the perturbation sequences, i.e., η. Fig. 3.8 shows that optimizing the perturbation covariance $\mathbf{\Sigma}^*$ increases the MSE while using partial sharing of information enhances robustness to Byzantine attacks by restricting the growth in MSE. In other words, as we share more information with neighbors, the impact of optimizing the perturbation covariance matrix increases. As an example, full information sharing results in a higher performance decrease, i.e., increasing MSE, when optimizing perturbation covariance compared to 25%-sharing case.

For different values of l, Fig. 3.9 plots the MSE$'$ versus time index n for optimized and random selection of the attack covariance. It can be seen that when less information is shared, the sensitivity to perturbation sequences with optimized covariance increases, resulting in high levels of fluctuation in the MSE$'$. In addition, Fig. 3.7 and Fig. 3.8 show that the optimized selection matrices have a greater impact when less information is shared, e.g., 25% and 50%-sharing, while optimal attack covariance has a higher impact when larger fractions of information are shared, e.g., 75% and full-sharing.

In order to analyze the robustness of the proposed BR-CDF algorithm to the num-

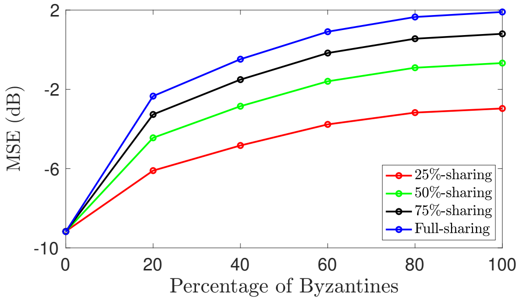


Figure 3.10: MSE versus percentage of the Byzantine agents in the network.

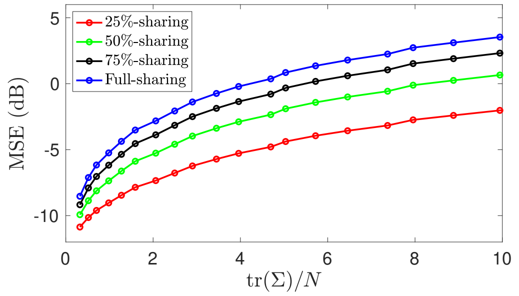


Figure 3.11: MSE versus trace of the attack covariance, i.e., $\mathrm{tr}(\mathbf{\Sigma})/N$.

ber of Byzantine agents, Fig. 3.10 plots the MSE versus the percentage of Byzantine agents in the network. As expected, we see that as the percentage of Byzantine agents increases, the MSE grows; however, partial sharing of information can significantly improve the resilience to Byzantine attacks, as illustrated by obtaining the lower MSE. In addition, Fig. 3.11 illustrates the MSE versus the trace of the attack covariance in order to assess the robustness of the BR-CDF algorithm to perturbation sequences. It can be seen that partial sharing of information improves robustness to injected noise by obtaining lower MSE.

3.3 Summary

This chapter investigated consensus-based distributed Kalman filtering algorithms in the presence of coordinated data falsification attacks. It was shown that the optimal set of Byzantine agents and their attack covariances that maximize the network-wide estimation error could be obtained by solving a sequence of semidefinite programs. It was also shown that partial sharing of information provides robustness to Byzantine attacks and reduces the communication load among agents by sharing a smaller portion of data at each time instant. Furthermore, this chapter characterized the performance and convergence of the BR-CDF algorithm and investigated the impact of coordinated data falsification attacks. In addition, the worst-case scenario of data falsification attacks was analyzed where Byzantine agents cooperate on designing the covariance of their falsification data or order of the information fractions they share. In the next chapter, in an attempt to conduct threat management, we will focus primarily on developing strategies for protecting agent information that is exposed to adversaries during DKF iterations. To this end, random decomposition and noise injection techniques are employed to limit the ability of the adversary to estimate private information.

Chapter 4

Privacy-Preserving Distributed Kalman Filtering

This chapter consists of the results of publications **P3**, **P4**, and **P5**, which focus on threat management and strategies for protecting private information from adversaries. The main focus of the chapter is on **P3** which covers the contributions in **P4** and **P5**. The work in **P3** considers the DKF setting in Section 2.1.2 and provides privacy by allowing agents to randomly decompose their private information into public and private substates, where the private substates will not be shared with neighbors. The public substates, however, would be perturbed by correlated noise sequences before sharing. Moreover, we investigate the performance and convergence of the privacy-preserving DKF (PP-DKF) and derive the guarantee of privacy bounds for network agents.

The rest of the chapter is organized as follows. Section 4.1 presents the PP-DKF and analyzes its stability and convergence. Section 4.2 characterizes agent privacy and provides privacy bounds for agents in the presence of an EE and an HBC adversary. Furthermore, Section 4.3 corroborates the theoretical findings with numerical simulations, and lastly, Section 4.4 summarizes the chapter.

4.1 Privacy-Preserving Distribute Kalman Filtering

Throughout this section, we discuss the distributed Kalman filtering algorithm in [67] and how to protect private data from being inferred by various adversaries. The literature contains various methods that address the privacy issues in distributed processing problems, such as distributed consensus [39–41, 88–90], distributed optimization [42, 91], and distributed filtering [92].

Regarding privacy concerns in Kalman filtering settings, the work in [60] designs a differentially private Kalman filter in both input and output perturbation cases. Furthermore, differentially private Kalman filtering solutions that minimize the achieved MSE under the DP constraints are proposed in [11, 58, 59]. From the privacy point of view, these works respect the privacy of individual data by employing DP constraints over private information. In contrast, we apply privacy constraints to protect the value of private information from being estimated by adversaries. The proposed privacy-aware Kalman filter in [93] linearly transforms the sensor measurements before releasing them to the fusion center to maximize the estimation error for the private state and minimize that for the public state. Although privacy-preserving Kalman filtering algorithms have been thoroughly investigated in the literature, the privacy-preserving framework for distributed Kalman filtering solutions is not adequately covered. The results in the literature studies and our proposed method cannot be compared due to differences in privacy measures. The literature also focuses primarily on centralized privacy-preserving KFs, which must be extended to distributed settings in order to be comparable to our proposed approach. Even then, only filtering performances can be fairly compared under the same perturbation sequence, and privacy still cannot be compared.

4.1.1 PP-DKF Algorithm

According to the privacy challenges in Section 2.2, in a DKF setting, information leakage occurs when agents share private information with each other. Considering the DKF in Section 2.1.2 and without loss of generality, we will consider the local states, $\boldsymbol{r}_{i,n}$ for each agent i, private. We aim to protect private information from being estimated by an adversary. For this purpose, we decompose the local states into public and private substates, where only noisy versions of the public substates are shared with neighbors.

The PP-DKF algorithm begins the system state estimation with local updates to the *a priori* state estimate and error covariance as in (2.8). The intermediate information of agent i, at time instant n, denoted by $\boldsymbol{\Gamma}_{i,n}$, is updated as in (2.9), and shared with neighbors to reach the average consensus. We assume that the condition for convergence of the covariance matrices to unique stabilizing solutions, as given in [67], are satisfied. Therefore, we have $\lim_{n\to\infty} \mathbf{P}_{i,n|n} = \mathbf{P}_i$ for each $i \in \mathcal{N}$. Then, the updated error covariance is employed to compute the intermediate state estimate of agent i as in (2.10). Sharing the local state estimate $\boldsymbol{r}_{i,n}$ with neighbors improves the estimation accuracy while exposing private information to adversaries. Thus, we protect the private data by modifying the ACF steps for local state estimates. To this end, the local state estimate is decomposed into a public substate $\boldsymbol{\alpha}_{i,n} \in \mathbb{R}^m$ and a private substate $\boldsymbol{\beta}_{i,n} \in \mathbb{R}^m$ before being shared among neighbors.

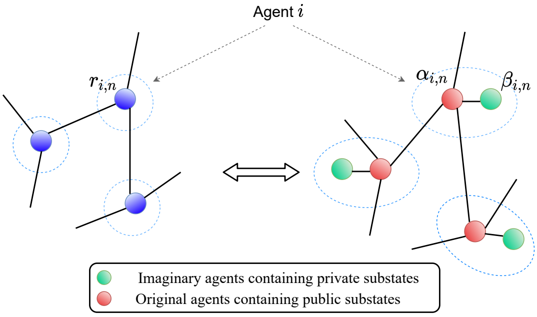

Agent i
$r_{i,n}$
$\alpha_{i,n}$
$\beta_{i,n}$
🟢 Imaginary agents containing private substates
🔴 Original agents containing public substates

Figure 4.1: State decomposition representation of $\boldsymbol{r}_{i,n}$ to public substate $\boldsymbol{\alpha}_{i,n}$ and private substate $\boldsymbol{\beta}_{i,n}$.

In particular, PP-DKF chooses the initial values $\boldsymbol{\alpha}_{i,n}(0)$ and $\boldsymbol{\beta}_{i,n}(0)$ randomly from the set of all real numbers in a manner that they satisfy $\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0) = 2\boldsymbol{r}_{i,n}$, [50], with $\boldsymbol{r}_{i,n}$ as the initial information of agent i to start the privacy-preserving average consensus mechanism. The substate $\boldsymbol{\alpha}_{i,n}$ is the only value that is shared with neighbors, while substate $\boldsymbol{\beta}_{i,n}$ evolves internally and will not be observed by neighbors, as represented in Figure 4.1. Although $\boldsymbol{\beta}_{i,n}$ remains invisible to neighbors, it directly affects the evolution of $\boldsymbol{\alpha}_{i,n}$. To improve privacy preservation, we also inject noise into the messages shared with neighbors; see, e.g., [48]. To that end, at each consensus iteration k, agent i shares a perturbed version of its public substate as $\tilde{\boldsymbol{\alpha}}_{i,n}(k) = \boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\omega}_i(k)$ where the perturbation noise sequence $\boldsymbol{\omega}_i(k)$ is given as in (2.15).

Accordingly, at each consensus iteration k, agent i updates its local substates using the received information from neighbors as follows:

$$\begin{cases} \boldsymbol{\alpha}_{i,n}(k+1) = \boldsymbol{\alpha}_{i,n}(k) + \varepsilon \mathbf{U}_i(k)\left(\boldsymbol{\beta}_{i,n}(k) - \boldsymbol{\alpha}_{i,n}(k)\right) \\ \qquad\qquad\qquad + \varepsilon \sum_{j \in \mathcal{N}_i} w_{ij}(k)\left(\tilde{\boldsymbol{\alpha}}_{j,n}(k) - \boldsymbol{\alpha}_{i,n}(k)\right) \\ \boldsymbol{\beta}_{i,n}(k+1) = \boldsymbol{\beta}_{i,n}(k) + \varepsilon \mathbf{U}_i(k)\left(\boldsymbol{\alpha}_{i,n}(k) - \boldsymbol{\beta}_{i,n}(k)\right) \end{cases} \tag{4.1}$$

where ε is the consensus step size, residing in $(0, \frac{1}{\Delta+1}]$ with $\Delta \triangleq \max_i N_i$. At consensus iteration k, $w_{ij}(k) = w_{ji}(k)$ denotes the interaction weight of agents i and j, while $\mathbf{U}_i(k) \triangleq \text{diag}(\mathbf{u}_i(k))$ is a diagonal matrix defined by the coupling weight vector $\mathbf{u}_i(k) \in \mathbb{R}^m$ of agent i. In particular, for $k = 0$, $w_{ij}(0) = w_{ji}(0)$ can be arbitrarily chosen from the set of all real numbers, while, for $k > 0$, we require that there exists a scalar $0 < \eta < 1$ such that all $w_{ij}(k) = w_{ji}(k)$, $j \in \mathcal{N}_i$ must reside in the range $[\eta, 1)$. This assumption ensures that each agent gives sufficient weight to the information received from its neighbors. As a result, the information

Algorithm 4 PP-DKF algorithm

```
For each agent $i \in \mathcal{N}$
Initialize: $\hat{\mathbf{x}}_{i,0|0}$ and $\mathbf{P}_{i,0|0}$
$\hat{\mathbf{x}}_{i,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$
$\mathbf{P}_{i,n|n-1} = \mathbf{A}\mathbf{P}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$
$\mathbf{\Gamma}_{i,n} = \mathbf{P}^{-1}_{i,n|n-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i$
$\mathbf{P}^{-1}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \mathbf{\Gamma}_{j,n}\}$
$\boldsymbol{r}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + N\mathbf{P}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)$
Privacy-Preserving Mechanism:
Select $\boldsymbol{\alpha}_{i,n}(0)$, and set $\boldsymbol{\beta}_{i,n}(0) = 2\boldsymbol{r}_{i,n} - \boldsymbol{\alpha}_{i,n}(0)$
Select the interaction and coupling weights $w_{ij}(k), \mathbf{u}_i(k)$
Share the interaction weights $w_{ij}(k)$ with neighbors
Generate $\{\boldsymbol{\omega}_i(k), k = 0, 1, \cdots, K\}$ based on (2.15)
Share $\tilde{\boldsymbol{\alpha}}_{i,n}(0) = \boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\omega}_i(0)$
for $k = 1$ to $K$ do
    Receive $\tilde{\boldsymbol{\alpha}}_{j,n}(k-1)$, $\forall j \in \mathcal{N}_i$
    Update $\boldsymbol{\alpha}_{i,n}(k)$ and $\boldsymbol{\beta}_{i,n}(k)$, as given in (4.1)
    Share $\tilde{\boldsymbol{\alpha}}_{i,n}(k) = \boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\omega}_i(k)$
end for
$\hat{\mathbf{x}}_{i,n|n} = \boldsymbol{\alpha}_{i,n}(K)$
```

from each agent continuously affects the information of other agents over time. Similarly, for $\mathbf{u}_i(k)$, the elements of $\mathbf{u}_i(0)$ are independently chosen from the set of all real numbers, while, for $k > 0$, they are limited to $[\eta, 1)$ to ensures that each agent gives sufficient weight to the private substates of the extended graph in Figure 4.1. Based on (4.1), the public substate is the only parameter that requires information from neighbors to update, whereas the private substate is updated only with information from the agent itself.

In the subsequent analysis, we assume that the interaction and coupling weights are arbitrarily chosen at $k = 0$ and remain fixed for $k > 0$ while satisfying the weighting mechanism in [50]. For notational convenience, the interaction weights of the entire network are collected into matrix $\mathbf{W} \triangleq [w_{ij}]$ for $k \geq 1$. In **P3**, we have shown that both private and public substates in (4.1) converge to the exact value of the average consensus, asymptotically, i.e.,

$$\lim_{k\to\infty} \boldsymbol{\alpha}_{i,n}(k) = \lim_{k\to\infty} \boldsymbol{\beta}_{i,n}(k) = \frac{1}{N}\sum_{i-1}^{N} \boldsymbol{r}_{i,n}.$$

Thus, in practice, after iterating the steps in (4.1) for sufficient number of iterations, say K, the local state estimate, $\hat{\mathbf{x}}_{i,n|n}$, is updated as

$$\hat{\mathbf{x}}_{i,n|n} = \boldsymbol{\alpha}_{i,n}(K) \;\; \forall i \in \mathcal{N}.$$

The steps of the proposed PP-DKF at each agent are summarized in Algorithm 4. In practice, the number of consensus iterations is always finite; hence, questions arise concerning its consequences on filtering performance, convergence behavior, and resulting privacy. Therefore, it is imperative to examine the effect of injected noise and state decomposition on the PP-DKF accuracy and privacy with a finite number of consensus iterations.

The detailed analysis of the mean and mean-square performance of Algorithm 4 can be found in **P3**, and here, only its results are discussed. In **P3**, we have shown that the proposed PP-DKF converges to the performance of the traditional DKF asymptotically. We also presented how a limited number of average consensus iterations affects the MSE performance. The MSE performance of Algorithm 4 with k consensus iterations is degraded compared to the non-private DKF. This indicates a performance-privacy tradeoff, which will be explored in greater detail in the following sections. In addition, in Section 4.3, numerical simulations will be provided to investigate the validity of the theoretical findings in this section.

4.2 Privacy Analysis

This section provides a comprehensive privacy analysis of the PP-DKF for two different adversaries: an EE and an HBC agent. The local estimate $\mathbf{r}_{j,n}$ is considered private since it corresponds to the local *a posteriori* estimate and includes more node-specific information than the global *a posteriori* state estimate $\hat{\mathbf{x}}_{j,n|n}$. As an output of the ACF, the *a posteriori* state estimate $\hat{\mathbf{x}}_{j,n|n}$ has the same value among agents; therefore, it contains less local information about agents. Similar to [48, 74], we assume that the adversary employs an estimator to infer the local estimate of agents at time n, i.e., $\boldsymbol{r}_{j,n}$ for $j \in \mathcal{N}$. We consider the MSE of the estimator at the adversary as the privacy metric. The MSE metric is used here to measure how accurately the adversary can estimate the exact value of the local *a posteriori* state estimates. Assume that $\hat{\boldsymbol{r}}_{j,n}(k)$ is the estimate of the private information at agent j at time n and after k consensus iterations, then, the corresponding privacy loss $\mathcal{E}_{j,n}(k)$ is the MSE given by

$$\mathcal{E}_{j,n}(k) \triangleq \operatorname{tr}\left(\mathbb{E}\{(\boldsymbol{r}_{j,n} - \hat{\boldsymbol{r}}_{j,n}(k))\,(\boldsymbol{r}_{j,n} - \hat{\boldsymbol{r}}_{j,n}(k))^{\mathrm{T}}\}\right). \tag{4.2}$$

4.2.1 Privacy in the Presence of an EE

The EE knows the network topology and can access all information exchanged among agents. Based on Algorithm 4, the information available at the EE after

k consensus iterations is $\mathcal{I}_{\mathrm{EE}}(k) = \{\tilde{\boldsymbol{\alpha}}_{j,n}(l), w_{ij}(l), \forall i, j \in \mathcal{N}\}_{l=0}^{k}$ where $\tilde{\boldsymbol{\alpha}}_{j,n}(l)$ is the perturbed substate and $w_{ij}(l)$ is the interaction weights exchanged with the neighbors. Employing the information set $\mathcal{I}_{\mathrm{EE}}(k)$, the EE estimates the local state of agents, i.e., $\boldsymbol{r}_{j,n}(k)$ for $j \in \mathcal{N}$, by constructing an observer at each consensus iteration. This results in the following theorem, which states the characterized privacy leakage for agent j.

Theorem 4.1. If the EE can only access messages shared by the agents, Algorithm 4 is privacy-preserving, and the privacy leakage for agent j is given by

$$\mathcal{E}_j = \lim_{n\to\infty} \lim_{k\to\infty} \mathcal{E}_{j,n}(k) = \mathsf{tr}\left((\mathbf{e}_j^{\mathrm{T}} \otimes \mathbf{I}_m)\, \tilde{\boldsymbol{\mathcal{L}}} \tilde{\boldsymbol{\Sigma}} \tilde{\boldsymbol{\mathcal{L}}}^{\mathrm{T}} (\mathbf{e}_j \otimes \mathbf{I}_m) \right) \tag{4.3}$$

where $\mathbf{e}_j \in \mathbb{R}^N$ is a vector with 1 in the jth entry and zeros elsewhere, $\tilde{\boldsymbol{\Sigma}}$ is the network-wide stabilizing error covariance of the filtering operation,

$$\tilde{\boldsymbol{\mathcal{L}}} = \frac{1}{2}\boldsymbol{\mathcal{L}} - \varepsilon \mathbf{U} \boldsymbol{\mathcal{L}} \boldsymbol{\Theta}\, \mathrm{diag}(\frac{1}{1-\lambda_1}, \cdots, \frac{1}{1-\lambda_{2Nm-m}}, 1, \cdots, 1)\boldsymbol{\Theta}^{\mathrm{T}} \tag{4.4}$$

with $\boldsymbol{\mathcal{L}} = [-\mathbf{I}, \mathbf{I}]$, $\lambda_1 < \cdots < \lambda_{2Nm-m} < 1$ are eigenvalues and $\boldsymbol{\Theta}$ is the matrix of eigenvectors of doubly stochastic matrix $\mathbf{G}$ given by

$$\mathbf{G} = \begin{bmatrix} \mathbf{M} & \varepsilon\mathbf{U} \\ \varepsilon\mathbf{U} & \mathbf{I} - \varepsilon\mathbf{U} \end{bmatrix} \tag{4.5}$$

with $\mathbf{M} \triangleq (\mathbf{I} - \varepsilon(\mathbf{D} - \mathbf{W})) \otimes \mathbf{I}_m - \varepsilon\mathbf{U}$, $\mathbf{U} = \mathsf{Blockdiag}(\{\mathbf{U}_i\}_{i=1}^{N})$, and $\mathbf{D} \triangleq \mathsf{diag}(\{\sum_{j\in\mathcal{N}_i} w_{ij}\}_{i=1}^{N})$.

Proof. The proof is given in [**P3**, Appendix B]. □

Here, agents communicate interaction weights with their neighbors so that the interaction weights remain symmetric, and thus the adversary can acquire $w_{ij}(l)$. However, if the EE does not know the initial interaction weights $w_{ij}(0)$, then the state of agents remains private with no information leakage, [50], and we can guarantee stronger privacy.

4.2.2 Privacy in the Presence of an HBC Agent

Without loss of generality, we assume that agent N is the HBC agent that uses its local information and the information received from its neighbors to estimate the private information of other agents. From Algorithm 4, we can see that the information available at the HBC agent N after k consensus iteration is given by

$$\mathcal{I}_{\mathrm{HBC}}(k) = \{\boldsymbol{\alpha}_{N,n}(l), \boldsymbol{\beta}_{N,n}(l), \boldsymbol{\omega}_N(l), \mathbf{u}_N(l), w_{Nj}(l), \tilde{\boldsymbol{\alpha}}_{j,n}(l) : \forall j \in \mathcal{N}_N\}_{l=0}^{k}$$

We can observe that agent privacy depends on the availability of the interaction and coupling weights at the adversary. Therefore, in order to analyze the worst-case scenario of an attack, we consider the case where the HBC agent has access to the entire weight matrix $\mathbf{W}$ and an estimate of the coupling weight matrix $\mathbf{U}$. This information set at the adversary can be represented as $\tilde{\mathcal{I}}_{\text{HBC}}(k) = \mathcal{I}_{\text{HBC}}(k) \cup \{\mathbf{W}(l), \hat{\mathbf{U}}(l)\}_{l=0}^{k}$ where $\hat{\mathbf{U}}$ denotes the estimate of the coupling weight matrix $\mathbf{U}$ at the adversary. Under these assumptions, the HBC agent can estimate the initial substate of agents, i.e., $\boldsymbol{\alpha}_{j,n}(0), \boldsymbol{\beta}_{j,n}(0)$ for all $j \in \mathcal{N}$. To this end, we define an observation vector that includes the shared information of neighbors and the information of the HBC agent itself at each time instant k given by

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{z}_n(k) + \mathbf{C}_\alpha \boldsymbol{\omega}(k), \tag{4.6}$$

where $\mathbf{z}_n(k) = [\boldsymbol{\alpha}_n^{\text{T}}(k), \boldsymbol{\beta}_n^{\text{T}}(k)]^{\text{T}}$ with $\boldsymbol{\alpha}_n(k) = [\boldsymbol{\alpha}_{1,n}^{\text{T}}(k), \cdots, \boldsymbol{\alpha}_{N,n}^{\text{T}}(k)]^{\text{T}}$ and $\boldsymbol{\beta}_n(k) = [\boldsymbol{\beta}_{1,n}^{\text{T}}(k), \cdots, \boldsymbol{\beta}_{N,n}^{\text{T}}(k)]^{\text{T}}$. In order to capture the relevant set of information, we define $\mathbf{C} \triangleq [\mathbf{C}_\alpha, \mathbf{C}_\beta]$ with $\mathbf{C}_\beta = [\mathbf{0}, \mathbf{e}_N]^{\text{T}} \otimes \mathbf{I}_m$ that captures the private substates of the HBC agent itself and $\mathbf{C}_\alpha = \left[\mathbf{e}_{j_1}, \cdots, \mathbf{e}_{j_{N_N}}, \mathbf{e}_N\right]^{\text{T}} \otimes \mathbf{I}_m$ that captures the public substate of neighbors and the HBC agent itself. The vector $\mathbf{e}_j \in \mathbb{R}^N$ is a vector with 1 in the jth entry and zeros elsewhere, $\mathcal{N}_N = \{j_1, \cdots, j_{N_N}\}$ is the adjacency set of the HBC agent and N_N denotes the cardinality of the adjacency set. As a result, the HBC agent infers the local state information of all agents by estimating $\boldsymbol{r}_n = 0.5(\boldsymbol{\alpha}_n(0) + \boldsymbol{\beta}_n(0))$.

Substituting the network-wide substate update equations in (4.1) into (4.6) gives

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{G}^k \mathbf{z}_n(0) + \mathbf{C}_\alpha \left(\sum_{t=0}^{k-1} \boldsymbol{\mathcal{C}}_{k-1-t} \boldsymbol{\mathcal{B}} \boldsymbol{\omega}(t) + \boldsymbol{\omega}(k) \right) \tag{4.7}$$

where $\boldsymbol{\mathcal{C}}_k = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \mathbf{G}^k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}^{\text{T}}$, $\boldsymbol{\omega}(k) = [\boldsymbol{\omega}_1^{\text{T}}(k), \cdots, \boldsymbol{\omega}_N^{\text{T}}(k)]^{\text{T}}$, and $\boldsymbol{\mathcal{B}} = \varepsilon(\mathbf{W} \otimes \mathbf{I}_m)$. Since $\boldsymbol{\nu}(k) = [\boldsymbol{\nu}_1^{\text{T}}(k), \cdots, \boldsymbol{\nu}_N^{\text{T}}(k)]^{\text{T}}$ is a zero-mean i.i.d. sequence, stacking all the available accumulated observations at each consensus iteration k in a vector gives

$$\begin{bmatrix} \frac{\sum_{t=0}^{0} \mathbf{y}_n(t)}{\phi^0} \\ \frac{\sum_{t=0}^{1} \mathbf{y}_n(t)}{\phi^1} \\ \vdots \\ \frac{\sum_{t=0}^{k} \mathbf{y}_n(t)}{\phi^k} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\mathbf{C}}{\phi^0} \\ \frac{\mathbf{C}(\mathbf{I}+\mathbf{G})}{\phi^1} \\ \vdots \\ \frac{\mathbf{C}(\mathbf{I}+\sum_{t=1}^{k} \mathbf{G}^t)}{\phi^k} \end{bmatrix}}_{\mathbf{H}(k)} \mathbf{z}_n(0) + \underbrace{\begin{bmatrix} \hat{\mathbf{F}}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \hat{\mathbf{F}}_1 & \hat{\mathbf{F}}_0 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{F}}_{k-1} & \hat{\mathbf{F}}_{k-2} & \cdots & \hat{\mathbf{F}}_0 \end{bmatrix}}_{\mathbf{F}(k)} \begin{bmatrix} \boldsymbol{\nu}(0) \\ \boldsymbol{\nu}(1) \\ \vdots \\ \boldsymbol{\nu}(k) \end{bmatrix} \tag{4.8}$$

where $\hat{\mathbf{F}}_0 = \mathbf{C}_\alpha$ and $\hat{\mathbf{F}}_k = \frac{\varepsilon}{\phi^{k+1}}\mathbf{C}_\alpha\boldsymbol{\mathcal{C}}_k(\mathbf{W}\otimes\mathbf{I}_m)$ for $k \geq 1$. Assuming the estimate of the coupling weight matrix $\mathbf{U}$ at the adversary as $\hat{\mathbf{U}} = \mathbf{U} + \boldsymbol{\Delta}_\mathbf{U}$, where $\boldsymbol{\Delta}_\mathbf{U}$ denotes the uncertainty in the estimate of the adversary, we quantify the privacy guarantee in the following results.

Theorem 4.2. If an HBC agent has access to the information $\{\mathbf{W}(l)\}_{l=0}^{k}$, the messages shared by its neighbors, and an estimate of the coupling weight matrix as $\hat{\mathbf{U}}$, then the error covariance at the HBC agent corresponding to estimate the initial substates $[\boldsymbol{\alpha}_n^\mathrm{T}(0), \boldsymbol{\beta}_n^\mathrm{T}(0)]^\mathrm{T}$ is given by

$$\tilde{\mathbf{P}}_n(k) = \bar{\mathbf{P}}_\mathrm{n}(k) + \mathbb{E}_\mathbf{U}\{\varepsilon^2\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_\mathbf{H}(k)\tilde{\boldsymbol{\Pi}}_n\boldsymbol{\Delta}_\mathbf{H}^\mathrm{T}(k)(\mathbf{H}^\dagger(k))^\mathrm{T}\} \tag{4.9}$$

where $\tilde{\boldsymbol{\Pi}}_n = \mathbf{1}_{2N}\mathbf{1}_{2N}^\mathrm{T}\otimes\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^\mathrm{T}\}$ with $\mathbf{x}_n$ as the state vector,

$$\begin{aligned}\bar{\mathbf{P}}_\mathrm{n}(k) = \mathbb{E}_\mathbf{U}\{&\varepsilon^2\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_\mathbf{H}(k)\tilde{\boldsymbol{\Sigma}}_n\boldsymbol{\Delta}_\mathbf{H}^\mathrm{T}(k)(\mathbf{H}^\dagger(k))^\mathrm{T}\\ &+\sigma^2(\mathbf{I}-\varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_\mathbf{H}(k))\mathbf{H}^\dagger(k)\mathbf{F}(k)\mathbf{F}^\mathrm{T}(k)(\mathbf{H}^\dagger(k))^\mathrm{T}\\ &(\mathbf{I}-\varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_\mathbf{H}(k))^\mathrm{T}\}\end{aligned} \tag{4.10}$$

with $\tilde{\boldsymbol{\Sigma}}_n$ as the error covariance of the filtering algorithm, $\mathbf{H}(k)$ and $\mathbf{F}(k)$ are defined in (4.8), and

$$\boldsymbol{\Delta}_\mathbf{H}(k) = \begin{bmatrix}\mathbf{0}\\ \phi^{-1}\mathbf{C}\boldsymbol{\Delta}_{\mathbf{G}_1}\\ \vdots\\ \phi^{-k}\mathbf{C}\sum_{t=1}^{k}\boldsymbol{\Delta}_{\mathbf{G}_t}\end{bmatrix}$$

with $\boldsymbol{\Delta}_{\mathbf{G}_k} = \sum_{t=1}^{k}\frac{k!\varepsilon^{t-1}}{(k-t)!t!}\mathbf{G}^{k-t}\boldsymbol{\Delta}_{\mathbf{G}_1}^{t}$, $\boldsymbol{\Delta}_{\mathbf{G}_1} = -\boldsymbol{\mathcal{L}}^\mathrm{T}\boldsymbol{\Delta}_\mathbf{U}\boldsymbol{\mathcal{L}}$, and $\boldsymbol{\mathcal{L}} = [-\mathbf{I}, \mathbf{I}]$.

Proof. The proof is given in [**P3**, Appendix C]. □

From Theorem 4.2, we can show that the first term in (4.9) converges to the fixed matrix $\bar{\mathbf{P}}_\mathrm{LB}(k) = \lim_{n\to\infty}\bar{\mathbf{P}}_\mathrm{n}(k)$ as $\lim_{n\to\infty}\tilde{\boldsymbol{\Sigma}}_n = \tilde{\boldsymbol{\Sigma}}$ and the second term diverges as $\lim_{n\to\infty}\mathrm{tr}\left(\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^\mathrm{T}\}\right) = \infty$. Therefore, a lower bound of the privacy leakage at agent j after k consensus iterations is given by

$$\bar{\mathcal{E}}_j(k) = \mathrm{tr}\left((\mathbf{e}_j^\mathrm{T}\otimes\mathbf{I}_m)\mathbf{P}(k)(\mathbf{e}_j\otimes\mathbf{I}_m)\right) \tag{4.11}$$

where $\mathbf{P}(k) = \frac{1}{4}\begin{bmatrix}\mathbf{I} & \mathbf{I}\end{bmatrix}\bar{\mathbf{P}}_\mathrm{LB}(k)\begin{bmatrix}\mathbf{I} & \mathbf{I}\end{bmatrix}^\mathrm{T}$. Additionally, for the worst-case scenario, when the HBC agent knows the exact coupling weights $\mathbf{U}$, i.e., $\boldsymbol{\Delta}_\mathbf{U} = \mathbf{0}$, then the error covariance $\tilde{\mathbf{P}}_n(k)$ in (4.9) is independent of n and is reduced as

$$\tilde{\mathbf{P}}(k) = \sigma^2\left(\mathbf{H}^\mathrm{T}(k)\left(\mathbf{F}(k)\mathbf{F}^\mathrm{T}(k)\right)^{-1}\mathbf{H}(k)\right)^{-1}. \tag{4.12}$$

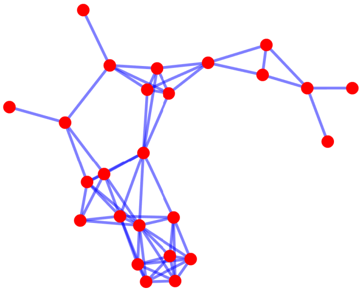


Figure 4.2: Network topology with $N = 25$ agents.

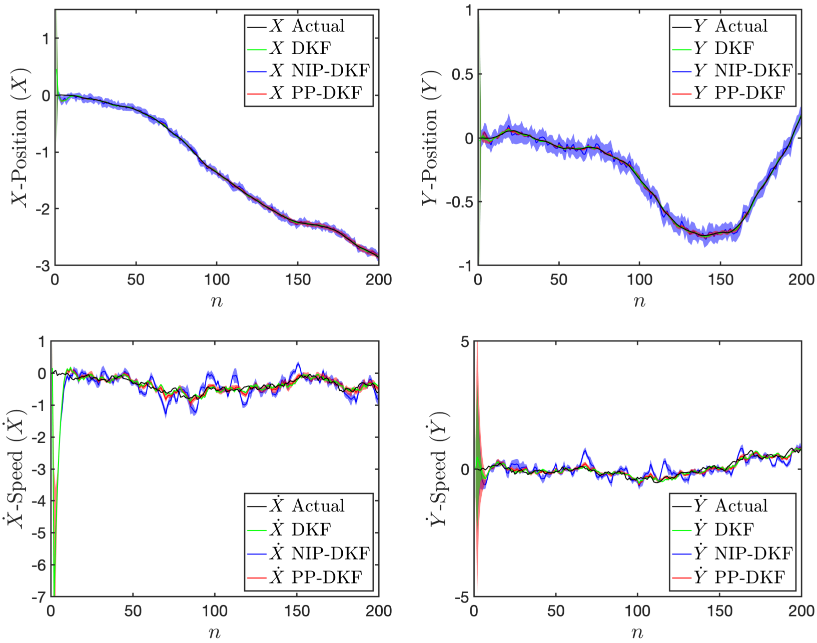


Figure 4.3: The DKF tracking performance for all agents (shaded color) and their average as a solid line with noise variance $\sigma^2 = 4$.

4.3 Numerical Results

To illustrate the performance of the PP-DKF algorithm, we consider the undirected connected network with $N = 25$ agents shown in Figure 4.2. The PP-DKF is used to collaboratively track the speed and position of a target moving in two dimensions where the state vector $\mathbf{x}_n = [X_n, Y_n, \dot{X}_n, \dot{Y}_n]^{\mathrm{T}}$ consists of the positions $\{X_n, Y_n\}$ and velocities $\{\dot{X}_n, \dot{Y}_n\}$ in the horizontal and vertical directions, respectively. The state evolution of such a dynamic system is given by

$$\mathbf{x}_n = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{n-1} + \begin{bmatrix} \frac{1}{2}(\Delta T)^2 & 0 \\ 0 & \frac{1}{2}(\Delta T)^2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} \mathbf{w}_n$$

where $\mathbf{w}_n = [\ddot{X}_n, \ddot{Y}_n]^{\mathrm{T}}$ denotes the unknown acceleration in horizontal and vertical directions and $\Delta T = 0.04$ is the sampling interval. The acceleration is modeled as zero-mean Gaussian process with covariance matrix of $\mathbb{E}\{\mathbf{w}_n \mathbf{w}_n^{\mathrm{T}}\} = 1.44\, \mathbf{I}_2$ while the observation parameters are considered as

$$\mathbf{H}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{R}_i = \begin{bmatrix} 0.0416 & 0.008 \\ 0.008 & 0.04 \end{bmatrix}$$

for each agents $i \in \mathcal{N}$. For comparison purposes, we introduce a DKF that employs the conventional noise-injection-based average consensus technique proposed in [48], with the injected noise following (2.15). This algorithm is hereafter referred to as the noise-injection-based privacy-preserving DKF (NIP-DKF). The consensus and noise parameters are selected as $\varepsilon = 1/4$ and $\phi = 0.9$, respectively. We considered the interaction weights given in [50], which is $\mathbf{W} = 0.75\mathbf{E}$ where $\mathbf{E}$ denotes the adjacency matrix. The elements of the coupling weight $\mathbf{u}_i$ are chosen independently with distribution $\mathcal{U}(\eta, 1)$ with $\eta = 0.4$ and the average consensus steps are iterated $K = 30$ times throughout the experiment.

4.3.1 Filtering Performance

Figure 4.3 shows the tracking performance of the proposed PP-DKF compared to the conventional DKF [67] and the NIP-DKF. We see that the PP-DKF performs as well as the conventional DKF which demonstrates the robustness of the PP-DKF to noise injection and state decomposition. Figure 4.4 shows how the perturbation noise variance σ^2 affects the average MSE of Kalman filtering algorithms. We see that the perturbation noise deteriorates the MSE performance compared to the conventional DKF [67], meaning that increasing the perturbation noise variance increases the MSE. Moreover, the slower growth rate of the MSE in PP-DKF compared to the NIP-DKF implies its improved robustness to the injected noise. Figure 4.4 additionally shows that the theoretical predictions for NIP-DKF and PP-DKF match the simulation results perfectly. [1]

Figure 4.5 shows the MSE of the PP-DKF and the NIP-DKF versus the number of consensus iterations. We see that increasing the number of consensus iterations

[1] To compute the filtering state vector estimation error for the NIP-DKF, we follow a similar approach to the PP-DKF in **P3**; the detailed derivation is provided in [**P3**, Appendix E].

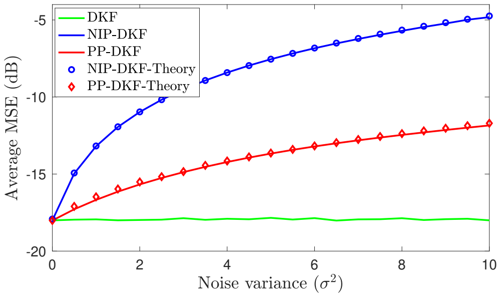


Figure 4.4: Average MSE of the DKF versus noise variance σ^2 for theory and simulation.

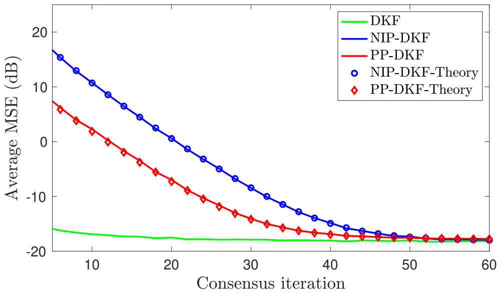


Figure 4.5: Average MSE of the DKF versus the number of consensus iterations with noise variance $\sigma^2 = 4$.

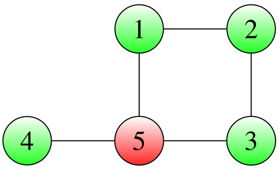


Figure 4.6: Network topology with $N = 5$ agents.

improves performance by reducing MSE. For a sufficiently large number of iterations, the filtering performance of the PP-DKF and the NIP-DKF converges to the conventional DKF [67]. Also, it can be seen that the theoretical predictions for a finite number of consensus iterations match the simulation results.

4.3.2 Privacy Analysis

To investigate the privacy performance of the PP-DKF algorithm, we need to focus more on the network and the effect of adversaries on each individual agent. We, therefore, consider a smaller undirected connected network with $N = 5$ agents

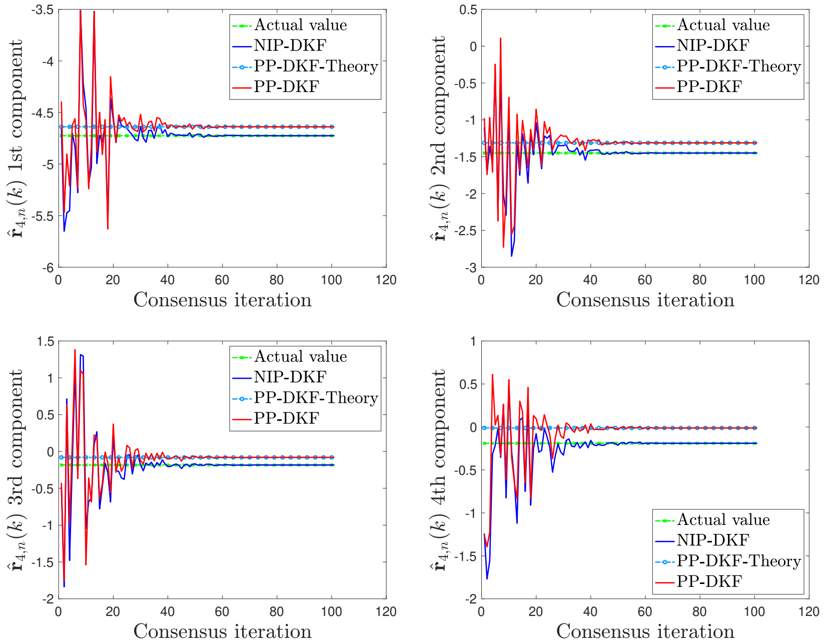


Figure 4.7: The observer of the EE to estimate components of the initial state $\boldsymbol{r}_{4,n}(0)$, i.e., $\hat{\boldsymbol{r}}_{4,n}(k)$, given the noise variance $\sigma^2 = 4$.

shown in Figure 4.6. We assume the EE follows the same approach for constructing observers to estimate the state of agents under NIP-DKF and PP-DKF. Figure 4.7 shows the state estimate of the EE versus the number of consensus iterations. It shows that whenever the NIP-DKF is employed, the eavesdropper can estimate the initial state with great accuracy. However, the PP-DKF prevents the initial state of the agents from being correctly estimated, as predicted by Theorem 4.1. Figure 4.7 also represents that the predicted estimation bias at the EE under the PP-DKF matches the simulation.

Figure 4.8 shows the average MSE obtained by the EE, i.e., $\frac{1}{N}\sum_{j=1}^{N}\mathcal{E}_j(k)$ with $\mathcal{E}_j(k)$ in (4.2), versus consensus iterations. Regarding the definition of privacy in (4.2), the larger the MSE at the EE, the better the privacy for network agents. Under the NIP-DKF, the average MSE at the EE decreases with increasing consensus iterations, which means the EE can determine the initial *a posteriori* state of the agents asymptotically. In contrast, under the PP-DKF, the achievable MSE at the EE is bounded as in (4.3), and, therefore, the estimation cannot be improved by extending the number of consensus iterations. Figure 4.8 also shows that the predicted bound of the privacy leakage in Theorem 4.1 matches the simulation.

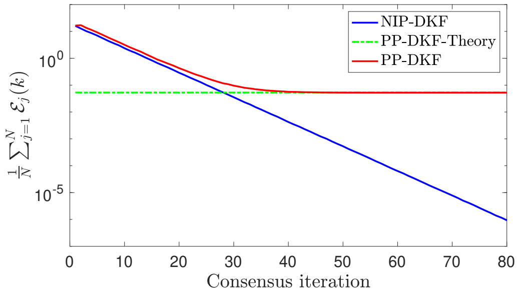


Figure 4.8: Average privacy $\frac{1}{N}\sum_{j=1}^{N}\mathcal{E}_j(k)$ versus the number of consensus iterations in the presence of the EE.

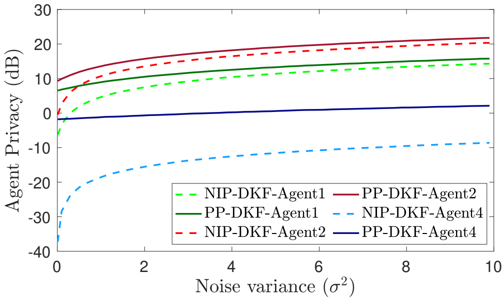


Figure 4.9: Agent privacy versus noise variance (σ^2). Due to the symmetric topology, agents 1 and 3 obtain same privacy and only the result of agent 1 is shown in the figure.

Here, we investigate the case when an HBC agent attempts to estimate the initial state of the network agents, considering the 5th agent in Figure 4.6 to be an HBC agent. The HBC agent has no access to the coupling weights of other agents, while a legitimate agent of the network knows the parameter η. Based on the assumption of the coupling weights distribution, the HBC agent uses an average value $\bar{\mathbf{U}}$, with uncertainty $\mathbf{\Delta_U} = \mathbf{U} - \bar{\mathbf{U}}$, to estimate the initial states of the other agents. Figure 4.9 shows the lower bound of the agent privacy in (4.11) versus the injected noise variance σ^2. We see that by employing the NIP-DKF, the privacy of agent 4 is breached due to the lack of neighbors other than the HBC agent. Consequently, the HBC agent can estimate the initial state of the 4th agent with negligible error. In contrast, the PP-DKF significantly improves the privacy of all agents even with a low amount of injected noise.

The tradeoff between filtering accuracy and the average privacy $\sum_{j=1}^{4}\bar{\mathcal{E}}_j(k)/4$ is shown in Figure 4.10. It illustrates the privacy-MSE tradeoff for different values of the injected noise variance σ^2. For both PP-DKF and NIP-DKF, we see that

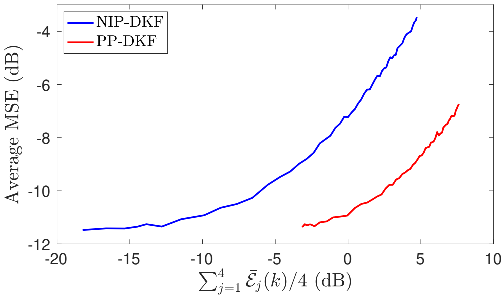


Figure 4.10: The tradeoff between Kalman filtering accuracy and average privacy $\sum_{j=1}^{4} \bar{\mathcal{E}}_j(k)/4$ for different values of the injected noise variance σ^2.

obtaining larger privacy guarantee reduces the filtering accuracy, which is reflected in a higher MSE. In addition, we see that a fixed privacy guarantee is ensured with lower MSE under the PP-DKF compared to the NIP-DKF. This is because the NIP-DKF perturbs the entire state estimate, whereas the PP-DKF perturbs only the public substate and keeps the private substate noise-free.

4.4 Summary

This chapter investigated possible privacy leaks in distributed Kalman filtering settings and developed an algorithm that protects private information from adversaries. It proposed the PP-DKF algorithm that protects sensitive data using state decomposition and noise injection techniques. Moreover, it analyzed the mean and mean-square convergence of the PP-DKF algorithm and provided a closed-form expression that captures agent privacy. In particular, it provided lower bounds on achieved privacy for various practical scenarios in the presence of an EE and an HBC adversary. Lastly, it demonstrated several simulations that corroborated the theoretical findings. Following the achievement of agent privacy, in the next chapter, we focus on improving the practical issue of communication efficiency. As a proof of concept and to demonstrate the adaptability of our proposed privacy-preserving algorithm, we focus on a simpler distributed learning system model, i.e., D-LMS. As a result, we propose a DL strategy that provides communication efficiency and privacy by limiting agents to only sharing a perturbed fraction of their private data.

Chapter 5

Privacy-Preserving and Communication-Efficient Distributed Learning

In this chapter, we present the results of publications **P7** and **P8** in which we examine how to reduce the load of local communications in distributed learning (DL) scenarios while protecting private information. The main focus of this chapter is on **P8** which also covers the contributions of **P7**. Focusing on communication efficiency is necessary, since local collaborations are realized through radio communications that consume considerable power and bandwidth. Consequently, a DL procedure that reduces communication load as much as possible without compromising agent privacy and network performance is always desirable.

Considering distributed least mean square (D-LMS) settings, the work in **P8** proposes two strategies that are both communication-efficient and privacy-preserving. The proposed partial sharing and privacy-preserving distributed learning (PPDL) algorithms achieve communication efficiency by allowing agents to share a fraction of information at each time instant and obtain privacy by noise injection and state decomposition average consensus techniques. The noise injection-based PPDL (NI-PPDL) algorithm enables agents to collaborate locally by sharing only a fraction of their perturbed information, thereby reducing resource consumption while maintaining privacy. On the other hand, decomposition and noise injection-based PPDL (DNI-PPDL) decomposes private information into public and private substates and allows agents to communicate only by sharing a perturbed fraction of their public substate. The chapter also includes the characterization of agent privacy in the presence of an HBC adversary and analyzes the impact of the privacy and the partial sharing of information on the overall performance of the LMS algorithm.

The remainder of this chapter is organized as follows. Section 5.1 examines the background information and system model of the D-LMS setting in a parameter estimation scenario. Section 5.2 proposes the NI-PPDL and DNI-PPDL algorithms and investigates their convergence and stability conditions. The privacy of agents in the presence of an HBC adversary is characterized in Section 5.3, while Section 5.4 provides numerical simulations to validate the theoretical findings. Lastly, Section 5.5 summarizes the chapter.

5.1 Background and Problem Formulation

Numerous studies have developed algorithms for improving communication efficiency in DL settings, but these studies have not been investigated in adversarial environments [94, 95]. In [96], a federated learning (FL) framework is employed to address privacy concerns, while [97] combines encryption and differential privacy, techniques to improve privacy in an FL scenario. Additionally, several studies have been conducted on simultaneously enhancing privacy and communication efficiency in DL settings [98–102] In these works, differential privacy [98–100] and homomorphic encryption [101, 102] methods are used for providing privacy, while partial participation and less frequent information exchange are employed to enhance communication efficiency [98–100]. According to the literature, an efficient method that improves privacy and communication efficiency in a distributed framework without imposing additional burdens on agents is still lacking. Additionally, previous studies mostly assumed a centralized processing unit aggregating information from multiple sources, while in this chapter, we consider a fully distributed architecture. Therefore, due to differences in the assumptions underlying network topology and the privacy-preserving measures used, our proposed method cannot be fairly compared with the existing literature. However, the proposed strategies are benchmarked against their noise-free scenarios, which demonstrate the impact of privacy constraints on their performances.

5.1.1 Problem Formulation

We consider a sensor network modeled as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ represents agents and the set of communication links between agents is denoted by $\mathcal{E}$. At each time instant n, agent i has access to an input signal $\mathbf{x}_{i,n} \in \mathbb{R}^m$ and the desired signal $y_{i,n} \in \mathbb{R}$ that is given by

$$y_{i,n} = \mathbf{x}_{i,n}^{\mathrm{T}} \mathbf{w}^{\star} + \epsilon_{i,n}, \tag{5.1}$$

where $\mathbf{w}^{\star} \in \mathbb{R}^m$ is the optimal parameter vector that has to be estimated, $\mathbf{x}_{i,n} \triangleq [x_{i,n}, x_{i,n-1}, \ldots, x_{i,n-m+1}]^{\mathrm{T}}$ is the input signal vector, and the observation noise $\epsilon_{i,n}$ is a zero-mean Gaussian random sequence with variance $\sigma^2_{\epsilon_i}$. The system parameter vector $\mathbf{w}^{\star}$ is estimated at each time instant n, i.e., $\mathbf{w}_n$, minimizing

$$\mathcal{J}_n \triangleq \frac{1}{N}\sum_{i\in\mathcal{N}} \mathbb{E}\{e^2_{i,n}\}, \tag{5.2}$$

where $e_{i,n} \triangleq y_{i,n} - \hat{y}_{i,n}$ with $\hat{y}_{i,n}$ as the estimated filter output at agent i. At each time instant n, $\mathbf{w}_n$ can be recursively updated in a steepest descent manner as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\eta}{2}\nabla \mathcal{J}_n, \tag{5.3}$$

where ∇ denotes the gradient operator, and η is the positive real-valued gain. Using an instantaneous approximation of the gradient, the learning update for $\mathbf{w}_n$ becomes

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \eta \sum_{i\in\mathcal{N}} \mathbf{x}_{i,n} e_{i,n} = \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{\psi}_{i,n+1}, \tag{5.4}$$

where $\boldsymbol{\psi}_{i,n+1}$ defined as

$$\boldsymbol{\psi}_{i,n+1} = \mathbf{w}_n + \mu \mathbf{x}_{i,n} e_{i,n}, \tag{5.5}$$

is an intermediate estimate of $\mathbf{w}^{\star}$ at agent i and time instant n with $\mu = \eta N$ as the step size. The average of the intermediate estimates $\boldsymbol{\psi}_{i,n+1}$ across the entire network can be evaluated in a distributed manner using an ACF [65, 103, 104], as stated in (2.12) and (2.13). To complete the estimation task, we assign the initial state of the ACF operations as $\mathbf{h}_i(0) = \boldsymbol{\psi}_{i,n+1}$ and run the ACF for k iteration to obtain the average of the $\boldsymbol{\psi}_{i,n+1}$ among agents. Then, at each agents i, the estimate of parameter vector is updated as $\mathbf{w}_{i,n+1} = \mathbf{h}_i(k)$.

Agents exchange local information $\boldsymbol{\psi}_{i,n+1}$ with their neighbors to obtain the average consensus. Potential adversaries attempt to access the node-sensitive information by exploiting the shared information. Thus, to safeguard node-sensitive data from being inferred by adversaries, agents must protect shared information when performing distributed consensus operations.

5.2 Distributed Learning Algorithms

As seen from Section 5.1, the collaboration between agents is vital for distributed learning. Although collaboration among agents improves learning accuracy, it is resource-intensive and exposes private information to adversaries. As stated in the previous section, agents only share their intermediate local estimates, as in (5.5), during the average consensus steps. Therefore, we propose the following PPDL algorithms that protect information exchange during consensus operations.

5.2.1 Noise Injection-based PPDL

In this section, we propose a PPDL algorithm that provides communication efficiency and privacy for distributed learning operations in (5.1) to (5.5). Without loss of generality, the intermediate estimate in (5.5) is considered private data that needs to be protected during interactions with neighbors. To provide privacy during the ACF process, each agent i perturbs its local information before sharing with neighbors, i.e., $\mathbf{h}_i(k) + \boldsymbol{\omega}_i(k)$ where $\boldsymbol{\omega}_i(k)$ is the perturbation sequence given in (2.15). Moreover, each agent only shares a fraction of the private information with neighbors (i.e., l entries of $\mathbf{h}_i(k)$, with $l \leq m$) to reduce the inter-node communication overhead. At each agent i and time instant n, the entry selection procedure at consensus iteration k is characterized by a diagonal selection matrix $\mathbf{S}_{i,n}(k)$ that consists of l numbers of ones and $m - l$ numbers of zeros on its main diagonal, same as in Section 3.2.1.

To keep the selection procedure simple, we adopt the coordinated partial-sharing, which is a special case of sequential partial-sharing-based communication method [57]. In coordinated partial-sharing, all agents are initialized with the same selection matrices, i.e., $\mathbf{S}_{1,0}(0) = \cdots = \mathbf{S}_{N,0}(0) = \mathbf{S}_0(0)$. This implies every agent in the network shares the same elements of the perturbed private information at the beginning of the process. The selection matrix at the next consensus iteration can be obtained by performing τ right-circular shift operations on the main diagonal elements of the entry selection matrix used in the current consensus iteration, as in Section 3.2.1. Since every agent in the network uses the same selection matrix at each time instance n and consensus iteration k, we drop node index in $\mathbf{S}_{i,n}(k)$ and continue with $\mathbf{S}_n(k)$. Using k consensus iterations at the ACF, the selection matrix at the next time instant is also updated as $\mathbf{S}_{i,n}(0) = \mathbf{S}_{i,n-1}(k)$. In the coordinated partial sharing, the frequency of each entry being shared is equal to $p_e = \frac{l}{m}$.

As a result, at each agent i, the ACF steps to obtain the average consensus of the intermediate local estimate in (5.5) can be expressed alternatively as

$$\mathbf{h}_i(k+1) = b_{ii}\tilde{\mathbf{h}}_i(k) + \sum_{j \in \mathcal{N}_i} b_{ji} \left(\mathbf{S}_n(k)\tilde{\mathbf{h}}_j(k) + (\mathbf{I} - \mathbf{S}_n(k))\tilde{\mathbf{h}}_j(k) \right),$$

where $\tilde{\mathbf{h}}_j(k) = \mathbf{h}_j(k) + \boldsymbol{\omega}_j(k)$. Due to the partial-sharing of information, agents substitute the missing entries of the received information, i.e., $(\mathbf{I} - \mathbf{S}_n(k))\tilde{\mathbf{h}}_j(k)$ with their local entries $(\mathbf{I} - \mathbf{S}_n(k))\tilde{\mathbf{h}}_i(k)$, resulting in

$$\mathbf{h}_i(k+1) = b_{ii}\tilde{\mathbf{h}}_i(k) + \sum_{j \in \mathcal{N}_i} b_{ji} \left(\mathbf{S}_n(k)\tilde{\mathbf{h}}_j(k) + (\mathbf{I} - \mathbf{S}_n(k))\tilde{\mathbf{h}}_i(k) \right) \cdot \tag{5.6}$$

Finally, after K consensus iterations, we update the local estimation as $\mathbf{w}_{i,n+1} = \mathbf{h}_i(K)$. The workflow of the NI-PPDL procedure is summarized in Algorithm 5.

Algorithm 5 NI-PPDL algorithm

```
At each time instant $n$ and agent $i$
Initialize: $\mathbf{S}_n(0) = \mathsf{diag}(\mathsf{s}_n(0))$ and $\tau$
$\hat{y}_{i,n} = \mathbf{x}_{i,n}^{\mathrm{T}} \mathbf{w}_{i,n}$
$e_{i,n} = y_{i,n} - \hat{y}_{i,n}$
$\boldsymbol{\psi}_{i,n+1} = \mathbf{w}_{i,n} + \mu \mathbf{x}_{i,n} e_{i,n}$
Set $\mathbf{h}_i(0) = \boldsymbol{\psi}_{i,n+1}$
for $k = 0$ to $K - 1$ do
    Share $\mathbf{S}_n(k)\tilde{\mathbf{h}}_i(k)$
    Receive $\left\{\mathbf{S}_n(k)\tilde{\mathbf{h}}_j(k) : \forall j \in \mathcal{N}_i\right\}$
    Update $\mathbf{h}_i(k+1)$ as given in (5.6)
    $\mathsf{s}_n(k+1) = \mathsf{right\text{-}circularshift}\{\mathsf{s}_n(k), \tau\}$
    $\mathbf{S}_n(k+1) = \mathsf{diag}(\mathsf{s}_n(k))$
end for
$\mathbf{w}_{i,n+1} = \mathbf{h}_i(K)$
```

5.2.2 Decomposition and Noise Injection-based PPDL

In this section, we propose a PPDL algorithm that offers privacy by employing noise injection and state decomposition and provides communication efficiency using the partial-sharing technique. Similar to the proposed NI-PPDL approach, the DNI-PPDL algorithm also modifies the ACF steps. The average consensus procedure begins by each agent i decomposing its local information $\mathbf{h}_i(0)$ into public and private substates. The public substate is exchanged with neighbors while the private substate is updated internally and will not be observed by neighbors [50]. Although the private substate is invisible to neighbors, it contributes directly to the evolution of the public substate. To provide the initial decomposition, each agent i chooses the initial public and private substates $\boldsymbol{\alpha}_i(0)$ and $\boldsymbol{\beta}_i(0)$ randomly from the set of all real numbers such that

$$\boldsymbol{\alpha}_i(0) + \boldsymbol{\beta}_i(0) = 2\mathbf{h}_i(0). \tag{5.7}$$

To simplify the mathematical derivations, we set the private substate as $\boldsymbol{\beta}_i(0) = \gamma \mathbf{h}_i(0)$, where γ is randomly chosen from the uniform distribution $\mathcal{U}(0, 1)$. This simplification subsequently results $\boldsymbol{\alpha}_i(0) = (2 - \gamma)\mathbf{h}_i(0)$. Agents further protect node-sensitive information by perturbing their public substates with random sequences, as in (2.15), at each consensus iteration k. Without considering communication efficiency, agents update their local substates as in (4.1) and obtain the average consensus while protecting private information. However, in the DNI-PPDL, during each consensus iteration k, every agent only shares a fraction of the perturbed public substate with neighbors (i.e., l entries of $\tilde{\boldsymbol{\alpha}}_i(k)$, with $l \leq m$) to reduce the inter-node communication overhead.

With the help of selection matrices in Section 5.2.1 and updating equations in (4.1), at each agent i, the public and private substates are alternatively updated as

$$\begin{cases} \boldsymbol{\alpha}_i(k+1) = \boldsymbol{\alpha}_i(k) + \varepsilon \mathbf{U}_i \left(\boldsymbol{\beta}_i(k) - \boldsymbol{\alpha}_i(k)\right) \\ \qquad + \varepsilon \displaystyle\sum_{l \in \mathcal{N}_i} w_{ji} \Big(\mathbf{S}_n(k) \tilde{\boldsymbol{\alpha}}_j(k) + (\mathbf{I} - \mathbf{S}_n(k)) \tilde{\boldsymbol{\alpha}}_j(k) - \boldsymbol{\alpha}_i(k) \Big), \\ \boldsymbol{\beta}_i(k+1) = \boldsymbol{\beta}_i(k) + \varepsilon \mathbf{U}_i \left(\boldsymbol{\alpha}_i(k) - \boldsymbol{\beta}_i(k)\right). \end{cases} \tag{5.8}$$

Agent i substitutes its local information $(\mathbf{I} - \mathbf{S}_n(k))\tilde{\boldsymbol{\alpha}}_i(k)$ with the missing entries of the perturbed public substate from neighbors $(\mathbf{I} - \mathbf{S}_n(k))\tilde{\boldsymbol{\alpha}}_j(k)$, which results in

$$\begin{cases} \boldsymbol{\alpha}_i(k+1) = \boldsymbol{\alpha}_i(k) + \varepsilon \mathbf{U}_i \left(\boldsymbol{\beta}_i(k) - \boldsymbol{\alpha}_i(k)\right) \\ \qquad + \varepsilon \displaystyle\sum_{j \in \mathcal{N}_i} w_{ji} \Big(\mathbf{S}_n(k) \tilde{\boldsymbol{\alpha}}_j(k) + (\mathbf{I} - \mathbf{S}_n(k)) \tilde{\boldsymbol{\alpha}}_i(k) - \boldsymbol{\alpha}_i(k) \Big), \\ \boldsymbol{\beta}_i(k+1) = \boldsymbol{\beta}_i(k) + \varepsilon \mathbf{U}_i \left(\boldsymbol{\alpha}_i(k) - \boldsymbol{\beta}_i(k)\right). \end{cases} \tag{5.9}$$

Iterating operations in (5.9) forces the public and private substates to reach consensus on the exact value of $\frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i(0)$, asymptotically. Thus, after K consensus iterations, we update the local estimate as $\mathbf{w}_{i,n+1} = \boldsymbol{\alpha}_i(K)$. The workflow of the proposed DNI-PPDL is summarized in Algorithm 6.

5.2.3 Learning Performance Analysis

Throughout this section, we examine the convergence behavior of the proposed NI-PPDL and DNI-PPDL strategies. In particular, we study the impact of partial-sharing-based communication and privacy constraints on the convergence of distributed learning algorithms. For establishing the convergence conditions and obtaining the closed-form expressions for network-level mean and mean squared error of the proposed PPDL strategies, we make the following assumptions:

- **A1**: For all $i \in \mathcal{N}$, the input signal vector $\mathbf{x}_{i,n}$ is drawn from a WSS multivariate random sequence with correlation matrix $\mathbf{R}_i \triangleq \mathrm{E}\{\mathbf{x}_{i,n}\mathbf{x}_{i,n}^{\mathrm{T}}\}$. Furthermore, the input signal vectors $\mathbf{x}_{i,n}$ and $\mathbf{x}_{j,s}$ are independent for all $i \neq j$ and $n \neq s$.

- **A2**: The observation noise process $\epsilon_{i,n}$ is assumed to be zero-mean i.i.d. and independent of other quantities.

Algorithm 6 DNI-PPDL algorithm

```
At each time instant $n$ and agent $i$
Initialize: $\mathbf{S}_n(0) = \mathsf{diag}(\mathsf{s}_n(0))$ and $\tau$
$\hat{y}_{i,n} = \mathbf{x}_{i,n}^{\mathrm{T}}\mathbf{w}_{i,n}$
$e_{i,n} = y_{i,n} - \hat{y}_{i,n}$
$\boldsymbol{\psi}_{i,n+1} = \mathbf{w}_{i,n} + \mu\mathbf{x}_{i,n}e_{i,n}$
Set $\mathbf{h}_i(0) = \boldsymbol{\psi}_{i,n+1}$
Private substate: $\boldsymbol{\beta}_i(0) = \gamma\mathbf{h}_i(0)$
Public substate: $\boldsymbol{\alpha}_i(0) = (2-\gamma)\mathbf{h}_i(0)$
for $k = 0$ to $K-1$ do
    Share $\mathbf{S}_n(k)\tilde{\boldsymbol{\alpha}}_i(k)$
    Receive $\{\mathbf{S}_n(k)\tilde{\boldsymbol{\alpha}}_j(k) : \forall j \in \mathcal{N}_i\}$
    Update substates $\boldsymbol{\alpha}_i(k+1)$ and $\boldsymbol{\beta}_i(k+1)$ as given in (5.9)
    $\mathsf{s}_n(k+1) = \mathsf{right\text{-}circularshift}\{\mathsf{s}_n(k), \tau\}$
    $\mathbf{S}_n(k+1) = \mathsf{diag}(\mathsf{s}_n(k))$
end for
$\mathbf{w}_{i,n+1} = \mathbf{h}_i(K)$
```

- **A3**: For all $i \in \mathcal{N}$, the selection matrix $\mathbf{S}_n(k)$ is independent of any other data. Additionally, the selection matrices $\mathbf{S}_n(k)$ and $\mathbf{S}_s(q)$ are independent for all $n \neq s$ and $k \neq q$.

- **A4**: For a sufficiently small learning rate μ, the terms involving higher-order powers of μ can be neglected.

Accordingly, the following theorem establishes the necessary conditions for the mean convergence of the proposed PPDL strategies.

Theorem 5.1. Let **A1**-**A3** hold true; then, the NI-PPDL and DNI-PPDL algorithms converge in the mean if and only if

$$0 < \mu < \frac{2}{\max\limits_{\forall p,i}\{\lambda_p(\mathbf{R}_i)\}} \tag{5.10}$$

where $\mathbf{R}_i = \mathrm{E}\{\mathbf{x}_{i,n}\mathbf{x}_{i,n}^{\mathrm{T}}\}$ and $\lambda_p(\cdot)$ is the pth eigenvalue of the argument matrix.

Proof. The approach for proof in algorithms NI-PPDL and DNI-PPDL are different, and the detailed proofs are given in **P8**. □

The following theorem characterizes the stability conditions for the mean-square convergence of the NI-PPDL and DNI-PPDL algorithms.

Theorem 5.2. Let **A1**-**A4** hold true; then the mean-square dynamics of the NI-PPDL and DNI-PPDL algorithms are stable if

$$0 < \mu < \frac{1}{\max\limits_{\forall p,i}\{\lambda_p(\mathbf{R}_i)\}}. \tag{5.11}$$

where $\mathbf{R}_i = \mathrm{E}\{\mathbf{x}_{i,n}\mathbf{x}_{i,n}^{\mathrm{T}}\}$ and $\lambda_p(\cdot)$ is the pth eigenvalue of the argument matrix.

Proof. The approach for proof in algorithms NI-PPDL and DNI-PPDL are different, and the detailed proofs are given in **P8**. □

Furthermore, **P8** demonstrates that the steady-state MSE of NI-PPDL and DNI-PPDL algorithms depends on the number of consensus iterations and is degraded by noise injection.

5.3 Privacy Analysis

This section examines the impact of partial sharing of information and noise injection on the privacy of agents. To this end, we analyze the privacy of agents in the presence of both HBC agents and external eavesdroppers. Similar to Section 4.2, the privacy of each agent j is defined as the mean squared estimation error at the adversary attempting to infer the private information $\mathbf{h}_j(0)$, i.e.,

$$\mathcal{E}_j(k) \triangleq \text{trace}\left(\mathbb{E}\left\{(\hat{\mathbf{h}}_j(k) - \mathbf{h}_j(0))(\hat{\mathbf{h}}_j(k) - \mathbf{h}_j(0))^{\mathrm{T}}\right\}\right), \tag{5.12}$$

where $\hat{\mathbf{h}}_j(k)$ denotes the estimate of the private information $\mathbf{h}_j(0)$ after k consensus iterations. The MSE metric used here measures how accurately the adversary can estimate private information.

5.3.1 Honest-but-Curious (HBC) Agent

Without loss of generality, we assume agent N is the HBC adversary, similar to Section 4.2.2. The HBC agent constructs a maximum likelihood (ML) estimator to estimate $\mathbf{h}_j(0)$ for $j \in \mathcal{N}\backslash\{N\}$, and characterizes the agent privacy at each agent j for the NI-PPDL strategy as

$$\mathcal{E}_j^{\mathrm{NI}}(k) = \text{trace}\left((\mathbf{e}_j^{\mathrm{T}} \otimes \mathbf{I}_m)\mathbf{P}^{\mathrm{NI}}(k)(\mathbf{e}_j \otimes \mathbf{I}_m)\right), \tag{5.13}$$

where $\mathbf{P}^{\mathrm{NI}}(k)$ is the ML estimator associated error covariance after k consensus iterations. The canonical vector corresponding to agent j is denoted by $\mathbf{e}_j$ with 1 at jth element and zeros elsewhere. Following a similar approach results in the privacy of the DNI-PPDL strategy at agents j and after k consensus iterations as

$$\mathcal{E}_j^{\mathrm{DNI}}(k) = \text{trace}\left((\mathbf{e}_j^{\mathrm{T}} \otimes \mathbf{I}_m)\mathbf{P}^{\mathrm{DNI}}(k)(\mathbf{e}_j \otimes \mathbf{I}_m)\right), \tag{5.14}$$

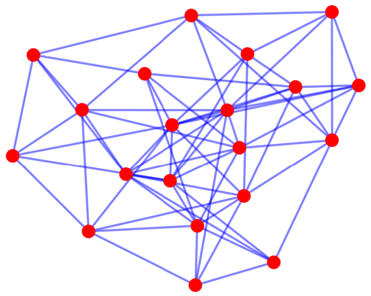


Figure 5.1: Distributed network with 20 agents.

where $\mathbf{P}^{\text{DNI}}(k)$ is the ML estimator associated error covariance after k consensus iterations. The detailed derivation of the privacy measure for both NI-PPDL and DNI-PPDL algorithms is given in **P8**.

5.3.2 External Eavesdropper

Regarding the definition of the EE in Section 2.4, in the proposed PPDL strategies, the selection matrices and circular shift variables are invisible to the EE since they are initialized during network establishment and are never shared during collaborations. Further, due to partial information sharing, an external eavesdropper can only access a fraction of perturbed entries during each consensus iteration k while being unable to determine their position and the size of the private information. Thus, the proposed PPDL strategies are resilient against external eavesdroppers with no information leakage.

5.4 Numerical Simulations

To demonstrate the effectiveness of NI-PPDL and DNI-PPDL algorithms, we conducted a series of simulations in the context of system identification in a random network of $N = 20$ agents with topology shown in Figure 5.1. Agents aim to estimate parameters of an unknown system of length $m = 32$. The input signal $x_{i,n}$ and observation noise sequence $\epsilon_{i,n}$, were drawn from zero-mean Gaussian distribution with variance $\sigma^2_{x_i} = 1$ and $\sigma^2_{\epsilon_i} \in \mathcal{U}(0.008, 0.03)$, respectively. The non-negative coefficients b_{ij} in the ACF steps of the NI-PPDL were obtained from the Metropolis rule in [65]. The interaction weights of the decomposition-based method were set as $\mathbf{W} = 0.8\mathbf{E}$ where $\mathbf{E}$ denotes the adjacency matrix of the network. The elements of the coupling weight $\mathbf{U}_i$ were chosen independently from a uniform distribution $\mathcal{U}(\eta, 1)$ where $\eta = 0.8$ and we set $\phi = 0.9$. The ACFs are iterated $K = 40$ times, and the perturbation noise sequence at each agent follows (2.15). The PPDL strategies were simulated under a coordinated partial shar-

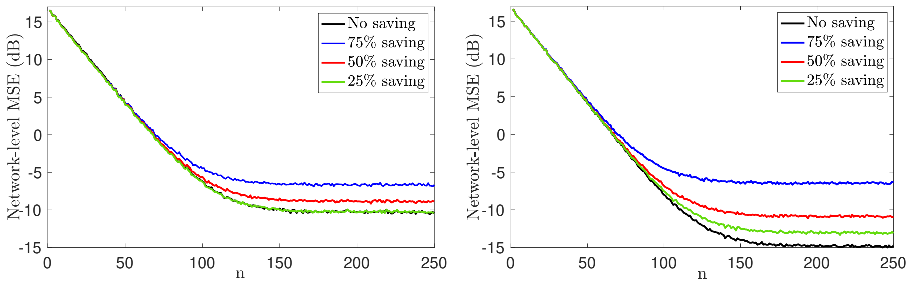


Figure 5.2: Learning curves of the proposed communication-efficient and privacy-preserving distributed learning strategies: (a). NI-PPDL. (b). DNI-PPDL.

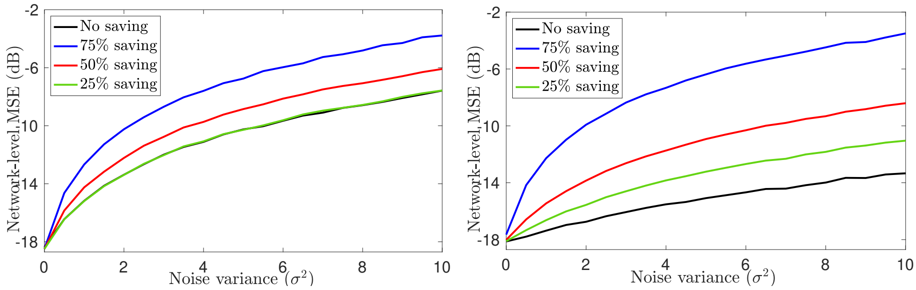


Figure 5.3: Steady-state network-level MSE vs perturbation noise variance: (a). NI-PPDL. (b). DNI-PPDL.

ing scheme for different values of l (say 32, 24, 16, 8, implying no saving, 25%, 50% and 75% communication-saving). The network-level MSE, which is given by $\frac{1}{N}\mathbb{E}\{\mathbf{e}_n^{\mathrm{T}}\mathbf{e}_n\}$ with $\mathbf{e}_n = [\mathbf{e}_{1,n}^{\mathrm{T}}, \cdots, \mathbf{e}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}$, is considered as the performance metric.

Figure 5.2 demonstrates the learning curves, i.e., network MSE in dB versus iteration index n, for $\sigma^2 = 5$. From Figure 5.2, we see that the proposed distributed learning strategies provide communication efficiency and privacy at the cost of slightly degrading performance. Increasing the communication-saving results in performance degradation, while even by saving 50% communications, the PPDL algorithms can achieve comparable performance with the case of no communication-saving. It is also evident from Figure 5.2 that the DNI-PPDL exhibits better estimation performance than the NI-PPDL because it injects perturbation noise only into a fraction of the public state, thus minimizing the overall contamination of the private information.

Figure 5.3 plots the steady-state network MSE of PPDL algorithms versus the perturbation noise variance σ^2, to examine the robustness of the proposed strategies to noise injection. As the variance of the perturbation noise increases, the per-

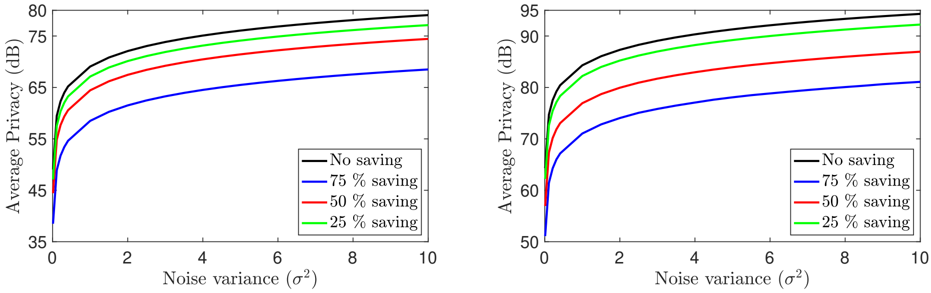


Figure 5.4: Average privacy versus perturbation noise variance: (a). NI-PPDL. (b). DNI-PPDL.

formance of the PPDL strategies reduces, but the reduction for DNI-PPDL is limited when compared to NI-PPDL. This behavior is due to the contribution of the noise-free private substate in its update equations. The injected noise with a higher variance impairs the learning performance regardless of the level of communication efficiency. However, the performance reduction is more pronounced when the communication savings are high. In other words, PPDL strategies become more sensitive to the perturbation noise variance when a smaller fraction of the information is shared at each instant.

Furthermore, we conducted experiments to investigate the impact of communication savings on agent privacy. Figure 5.4 demonstrates the average privacy of network agents, defined as $\bar{\mathcal{E}} \triangleq \frac{1}{N}\sum_{j=1}^{N-1}\mathcal{E}_j(k)$ with $\mathcal{E}_j(k)$ in (5.12), versus the perturbation noise variance σ^2. From Figure 5.4, we see that, in both NI-PPDL and DNI-PPDL methods, increasing the variance of the perturbation noise increases the average privacy regardless of the level of communication savings. It can be seen that sharing a smaller fraction of information at each time results in a lower level of average privacy in the network. In other words, by sharing a larger fraction of information at each iteration, cumulative noise in the elements of the private information increases, resulting in a higher estimation error at the HBC agent and a higher privacy guarantee. From Figure 5.4 (b), it is also evident that the DNI-PPDL offers better privacy than the NI-PPDL for a given value of σ^2. Although privacy decreases with an increase in communication efficiency, the privacy achieved by the DNI-PPDL with 75% communication savings is higher than the privacy obtained by the NI-PPDL approach without communication savings.

5.5 Summary

This chapter proposed two PPDL algorithms that offer communication efficiency while preserving privacy. We proposed the NI-PPDL algorithm that allows agents to share only a fraction of their perturbed private information with their neighbors. The DNI-PPDL algorithm, on the other hand, randomly decomposes the private information into public and private substates and only shares a perturbed version of the public substates. Mean and mean-square convergence analyses were conducted to determine the impact of partial information sharing and privacy constraints on the performance of the PPDL algorithms. The privacy was also characterized in the presence of an HBC adversary. The proposed algorithms achieved communication efficiency with the cost of performance and privacy. However, numerical simulations showed that the DNI-PPDL with 50% communication savings achieved nearly the same learning performance and significantly improved privacy compared to the NI-PPDL without communication savings. We mainly focused on protecting distributed algorithms against HBCs and EEs in the last two chapters. However, the development of an algorithm that can perform robustly in the presence of Byzantine adversaries remains a concern. To this end, in the next chapter, we investigate a DKF strategy that limits the impact of Byzantine adversaries on the network without a significant increase in agent local computation.

Chapter 6

Distributed Kalman Filter with Enhanced Resilience to Byzantine Attacks

In this chapter, we present the results of publication **P6** where we propose a DKF based on an optimization framework to provide resilience to Byzantine attacks. First, we design the filtering algorithm by adapting the framework proposed in [105] to model the Kalman filtering algorithm as a solution to an optimization problem. Then, we use the total variation (TV)-norm penalty in the objective function to enforce resilience to data falsification attacks [106–108]. We solve the TV-norm-penalized optimization problem using a distributed subgradient algorithm that updates the state estimate for all agents through local collaborations. We show that the proposed TV-norm penalized optimization problem corresponding to the state estimate update results in the same solution as the centralized Kalman filter (CKF), and it converges to a bounded neighborhood of the optimal solution when Byzantine agents are present.

The rest of this chapter is organized as follows. Section 6.1 proposes the Byzantine-resilient distributed Kalman filter (BR-DKF) as a solution to a TV-norm-penalized optimization problem. The BR-DKF is studied for convergence and stability under Byzantine attacks in Section 6.2, and Section 6.3 provides numerical simulations to corroborate the theoretical findings. Finally, Section 6.4 summarizes the chapter.

6.1 Byzantine Robust Distributed Kalman Filter

Many studies in the literature were proposed to provide robustness against Byzantine agents using statistical approaches [25–28]. The methods mainly involve assigning adaptive weights to received measurements from neighbors. They assign smaller weights to measurements that are most likely to originate from a Byzantine agent, which results in a smaller impact on state estimates. To provide robustness against adversaries, homomorphic encryption-based schemes [31–33, 61], randomization-based methods [34], and redundancy-based approaches [35–38] have also been proposed in the literature. However, these approaches require more local computations and information transfer in the network, which is undesirable in resource-constrained situations. Thus, this chapter focuses on developing a distributed filtering algorithm that can achieve robustness against Byzantine attacks without requiring extra computations on agents. Although the Kalman filtering algorithm has been modeled as an optimization problem in the literature [105], it has not been analyzed in adversarial situations or adapted for robustness in the presence of Byzantine agents. Consequently, the literature does not cover optimization-based DKF algorithms with different attack-resilient methods that can be compared to our proposed algorithm in this chapter. However, in the future, it may be possible to compare our strategy with a similar DKF strategy that employs a different penalty term in the objective function.

6.1.1 System Model

We consider a network of N agents that exchange information with their neighbors to develop their optimal estimates. The state-space model characterizes the state vector evolution, and observation vectors are given in (2.1) and (2.2). Each agent $i \in \mathcal{N}$ updates its local estimate by using information from its neighbors.

We revisit the DKF algorithm modeled as an ML estimation problem that essentially represents the relationship between a KF algorithm [7] and an optimization problem [105]. Similar to the centralized case in [105], the modeling of the DKF also requires two steps of prediction and correction, where for each agent i and time instant n, the prediction updates are stated as

$$\begin{aligned} \hat{\mathbf{x}}_{i,n|n-1} &= \mathbf{A}\hat{\mathbf{x}}_{i,n-1} \\ \mathbf{P}_{i,n|n-1} &= \mathbf{A}\mathbf{P}_{i,n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q} \end{aligned} \tag{6.1}$$

with $\hat{\mathbf{x}}_{i,n-1}$ and $\mathbf{P}_{i,n-1} = \mathbb{E}\{\mathbf{e}_{i,n-1}\mathbf{e}_{i,n-1}^{\mathrm{T}}\}$ being the estimate and error covariance matrix at time instant $n-1$, and $\mathbf{e}_{i,n-1} = \mathbf{x}_{n-1} - \hat{\mathbf{x}}_{i,n-1}$. The intermediate *a priori* state estimate and error covariance are denoted by $\hat{\mathbf{x}}_{i,n|n-1}$ and $\mathbf{P}_{i,n|n-1} = \mathbb{E}\{\mathbf{e}_{i,n|n-1}\mathbf{e}_{i,n|n-1}^{\mathrm{T}}\}$, respectively, with $\mathbf{e}_{i,n|n-1} = \mathbf{x}_n - \hat{\mathbf{x}}_{i,n|n-1}$. Accordingly, the correction steps of the DKF can be modeled as the solution of a constrained optimization problem [105]; in particular, the *a posteriori* state estimates can be obtained by solving the optimization problem

$$\begin{aligned} &\min_{\{\mathbf{x}_{i,n}\}_{i=1}^{N}} \sum_{i=1}^{N} f_i(\mathbf{x}_{i,n}) \\ &\text{s. t. } \mathbf{x}_{i,n} = \mathbf{x}_{j,n}, \ \ \forall j \in \mathcal{N}_i, i \in \mathcal{N} \end{aligned} \tag{6.2}$$

where the local objective function $f_i(\mathbf{x}_{i,n})$ is given by

$$\begin{aligned} f_i(\mathbf{x}_{i,n}) =& \frac{1}{2}\Bigg((\mathbf{y}_{i,n} - \mathbf{H}_i\mathbf{x}_{i,n})^{\mathrm{T}}\mathbf{R}_i^{-1}(\mathbf{y}_{i,n} - \mathbf{H}_i\mathbf{x}_{i,n}) \\ &+ \frac{1}{N}(\mathbf{x}_{i,n} - \hat{\mathbf{x}}_{i,n|n-1})^{\mathrm{T}}\mathbf{P}_{i,n|n-1}^{-1}(\mathbf{x}_{i,n} - \hat{\mathbf{x}}_{i,n|n-1})\Bigg) \end{aligned} \tag{6.3}$$

and the constraints enforce consensus across all the agents in the network. The distributed Kalman filtering problem can be solved by any distributed algorithm that finds the optimal solutions in (6.2), i.e., $\mathbf{x}_{i,n}^*$ for each $i \in \mathcal{N}$. Subsequently, the *a posteriori* state estimates of agents are updated as $\hat{\mathbf{x}}_n = [\hat{\mathbf{x}}_{1,n}^{\mathrm{T}}, \cdots, \hat{\mathbf{x}}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}$ where $\hat{\mathbf{x}}_{i,n} = \mathbf{x}_{i,n}^*$.

Motivated by [106, 107], the constraints in (6.2) can be approximated by a TV-norm penalty which also endows robustness to data falsification attacks. Thus, the optimization problem in (6.2) can be can be formulated as TV-norm-penalized problem given by

$$\bar{\mathbf{x}}_n^* = \min_{\{\mathbf{x}_{i,n}\}_{i=1}^{N}} \sum_{i=1}^{N} \left(f(\mathbf{x}_{i,n}) + \frac{\lambda_{\text{tv}}}{2} \sum_{j\in\mathcal{N}_i} \|\mathbf{x}_{i,n} - \mathbf{x}_{j,n}\|_1 \right) \tag{6.4}$$

where $\bar{\mathbf{x}}_n^* = [(\mathbf{x}_{1,n}^*)^{\mathrm{T}}, \cdots, (\mathbf{x}_{N,n}^*)^{\mathrm{T}}]^{\mathrm{T}}$ and λ_{tv} is a penalty parameter. Due to the penalty parameter λ_{tv}, estimates $\mathbf{x}_{i,n}$ and $\mathbf{x}_{j,n}$ are forced to be close. The larger the λ_{tv}, the closer $\mathbf{x}_{i,n}$ and $\mathbf{x}_{j,n}$ become. However, the TV-norm penalty allows for some pairs of $\mathbf{x}_{i,n}$ and $\mathbf{x}_{j,n}$ to be different, which is crucial when Byzantine agents are present in the network.

We solve the optimization problem in (6.4) with a subgradient method [107], and derive the state estimate update at each agent $i \in \mathcal{N}$ as

$$\mathbf{x}_{i,n}^{l+1} = \mathbf{x}_{i,n}^{l} - \alpha_n \left(\nabla_{\mathbf{x}_{i,n}} f(\mathbf{x}_{i,n}^{l}) + \lambda_{\text{tv}} \sum_{j\in\mathcal{N}_i} \text{sign}(\mathbf{x}_{i,n}^{l} - \mathbf{x}_{j,n}^{l}) \right) \tag{6.5}$$

where $\alpha_n > 0$ denotes the step size and $\mathbf{x}_{i,n}^{l}$ is the state estimate of the subgradient method at agent i and iteration l. The element-wise sign function is represented by sign$(\cdot)$ where given $x > 0$, sign$(x) = 1$ and sign$(x) = -1$ when $x < 0$. In case of $x = 0$, the value of sign(x) can be any arbitrary value within $[-1, 1]$. Assuming that a group of agents, i.e., $\mathcal{B} \subset \mathcal{N}$, is conducting Byzantine attacks, as in Section 2.4.2, and by substituting the gradient $\nabla_{\mathbf{x}_{i,n}} f(\mathbf{x}_{i,n}^l)$, we obtain

$$\begin{aligned}\mathbf{x}_{i,n}^{l+1} = \mathbf{x}_{i,n}^l - \alpha_n \bigg(& \boldsymbol{\Omega}_{i,n}\mathbf{x}_{i,n}^l - \boldsymbol{\theta}_{i,n} + \lambda_{\text{tv}} \sum_{j\in\mathcal{R}_i} \text{sign}(\mathbf{x}_{i,n}^l - \mathbf{x}_{j,n}^l) \\ & + \lambda_{\text{tv}} \sum_{j\in\mathcal{B}_i} \text{sign}(\mathbf{x}_{i,n}^l - \tilde{\mathbf{x}}_{j,n}^l) \bigg)\end{aligned} \tag{6.6}$$

where $\tilde{\mathbf{x}}_{j,n}^l = \mathbf{x}_{j,n}^l + \boldsymbol{\delta}_j^l$ is the state estimate received from the jth Byzantine neighbor, $\mathcal{R}_i$ and $\mathcal{B}_i$ include honest and Byzantine members of $\mathcal{N}_i$, and

$$\begin{aligned}\boldsymbol{\Omega}_{i,n} &= \mathbf{H}_i^{\text{T}}\mathbf{R}_i^{-1}\mathbf{H}_i + \frac{1}{N}\mathbf{P}_{i,n|n-1}^{-1} \\ \boldsymbol{\theta}_{i,n} &= \mathbf{H}_i^{\text{T}}\mathbf{R}_i^{-1}\mathbf{y}_{i,n} + \frac{1}{N}\boldsymbol{\Omega}_{i,n|n-1}\hat{\mathbf{x}}_{i,n|n-1}\end{aligned} \tag{6.7}$$

with $\boldsymbol{\Omega}_{i,n|n-1} = \mathbf{P}_{i,n|n-1}^{-1}$. Regardless of the state estimate received from neighbors, the value of sign$(\mathbf{x}_{i,n}^l - \tilde{\mathbf{x}}_{j,n}^l)$ is restricted to $[-1, 1]$. Thus, the last term in (6.6) limits the effect of perturbed data received from a Byzantine agent so that the state estimate update is more robust to Byzantine attacks.

Similarly, the error covariance update also requires designing an optimization problem to obtain the average consensus of the information matrices $N\boldsymbol{\Omega}_{i,n}$ throughout the network. To this end, we propose the following optimization problem that updates the error covariance as

$$\begin{aligned}\min_{\{\boldsymbol{\zeta}_i\}_{i=1}^N} \quad & \sum_{i=1}^{N} \|\boldsymbol{\zeta}_i - \text{vec}_h(N\boldsymbol{\Omega}_{i,n})\|_2^2 \\ \text{s. t.} \quad & \boldsymbol{\zeta}_i = \boldsymbol{\zeta}_j, \ \forall j \in \mathcal{N}_i, i \in \mathcal{N}\cdot\end{aligned} \tag{6.8}$$

The optimal solution of (6.8) is denoted by $\boldsymbol{\zeta}^* = [(\boldsymbol{\zeta}_1^*)^{\text{T}}, \cdots, (\boldsymbol{\zeta}_N^*)^{\text{T}}]^{\text{T}}$ which returns the average of $\text{vec}_h(N\boldsymbol{\Omega}_{i,n})$ throughout the entire network. [1] Subsequently, the error covariance matrix can be updated as $\mathbf{P}_{i,n} = (\text{vec}_h^{-1}(\boldsymbol{\zeta}_i^*))^{-1}$. [2] Motivated

[1] The half vectorization of a symmetric matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$is denoted by $\text{vec}_h(\mathbf{M}) \in \mathbb{R}^{m(m+1)/2}$, where $\text{vec}_h(\mathbf{M}) = [M_{1,1}, \cdots, M_{1,m}, M_{2,2}, \cdots, M_{2,m}, \cdots, M_{m,m}]^{\text{T}}$ with M_{ij} as the ijth element of $\mathbf{M}$.

[2] The operator of $\text{vec}_h^{-1}(\cdot)$ denotes the inverse function of $\text{vec}_h(\cdot)$, i.e., $\text{vec}_h^{-1}(\text{vec}_h(\mathbf{M})) = \mathbf{M}$.

by the TV-norm-penalized optimization problem in (6.4), we modify the optimization problem in (6.8) as

$$\boldsymbol{\zeta}^* = \min_{\{\boldsymbol{\zeta}_i\}_{i=1}^N} \sum_{i=1}^{N} \left(\|\boldsymbol{\zeta}_i - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,n})\|_2^2 + \frac{\lambda_{\text{tv}}}{2} \sum_{j\in\mathcal{N}_i} \|\boldsymbol{\zeta}_i - \boldsymbol{\zeta}_j\|_1 \right) \cdot \tag{6.9}$$

Employing a similar subgradient approach as in (6.5), results in

$$\boldsymbol{\zeta}_i^{l+1} = \boldsymbol{\zeta}_i^l - \gamma_n \left(\boldsymbol{\zeta}_i^l - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,n}) + \lambda_{\text{tv}} \sum_{j\in\mathcal{N}_i} \text{sign}(\boldsymbol{\zeta}_i^l - \boldsymbol{\zeta}_j^l) \right), \tag{6.10}$$

where $\gamma_n > 0$ denotes the step size. After a large enough number of iterations, say l^*, the suboptimal solutions in (6.6) and (6.10) converge to $(\mathbf{x}_{i,n}^{l^*}, \boldsymbol{\zeta}_i^{l^*})$ and the filtering *a posteriori* state estimate and error covariance matrix can be updated as

$$\hat{\mathbf{x}}_{i,n} = \mathbf{x}_{i,n}^{l^*}$$
$$\mathbf{P}_{i,n} = (\mathsf{vec}_h^{-1}(\boldsymbol{\zeta}_i^{l^*}))^{-1}.$$

Assuming that Byzantine agents manipulate only state estimates, i.e., falsify the state estimate $\mathbf{x}_{i,n}^l$ at each iteration l as $\mathbf{x}_{i,n}^l + \boldsymbol{\delta}_i^l$ with $\boldsymbol{\delta}_i^l \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i)$ and $\boldsymbol{\Sigma}_i$ denoting the covariance of perturbation sequences of agent $i \in \mathcal{B}$, Algorithm 7 summarizes detailed steps of the BR-DKF.

6.2 Performance Analysis

In this section, we demonstrate that the TV-norm-penalized problem in (6.4) yields a feasible solution when the penalty parameter λ_{tv} is sufficiently large. We also show that the suboptimal solution in (6.6) converges to a neighborhood of the optimal solution of the problem in (6.4) with a bounded radius when Byzantine agents are in the network. To assist the future calculations, we define $\boldsymbol{\mathcal{A}} = [a_{ij}] \in \mathbb{R}^{N\times|\mathcal{E}|}$ as the node-edge incidence matrix where for each edge $e = (i, j) \in \mathcal{E}$ with $i < j$, we set $a_{ei} = 1$ and $a_{je} = -1$, otherwise, the elements of $\boldsymbol{\mathcal{A}}$ remain zero. In the following theorem, we establish the optimality of the proposed solution in (6.4) to yield the same solution as the centralized Kalman filter solution $\hat{\mathbf{x}}_n^*$ in [105]. We provide a lower bound threshold for the penalty parameter λ_{tv} that guarantees convergence of the solution in (6.4) to the centralized solution in [105].

Theorem 6.1. Given that the network topology is connected, if $\lambda_{\text{tv}} \geq \lambda_0$ where

$$\lambda_0 = \frac{\sqrt{N}}{\sigma_{\min}(\boldsymbol{\mathcal{A}})} \max_{\forall n} \max_{i\in\mathcal{N}} \|\boldsymbol{\Omega}_{i,n}\mathbf{x}_{i,n}^* - \boldsymbol{\theta}_{i,n}\|_\infty \tag{6.11}$$

Algorithm 7 BR-DKF algorithm

```
For each agent $i \in \mathcal{N}$
Initialize: $\hat{\mathbf{x}}_{i,0}$ and $\mathbf{P}_{i,0}$
for all $n > 0$ do
    $\hat{\mathbf{x}}_{i,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{i,n-1}$
    $\mathbf{P}_{i,n|n-1} = \mathbf{A}\mathbf{P}_{i,n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$
    $\boldsymbol{\Omega}_{i,n|n-1} = \mathbf{P}^{-1}_{i,n|n-1}$
    $\boldsymbol{\Omega}_{i,n} = \mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i + \frac{1}{N}\boldsymbol{\Omega}_{i,n|n-1}$
    $\boldsymbol{\theta}_{i,n} = \mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{y}_{i,n} + \frac{1}{N}\boldsymbol{\Omega}_{i,n|n-1}\hat{\mathbf{x}}_{i,n|n-1}$
    Set $\mathbf{x}^1_{i,n} = \mathbf{0}$ and $\boldsymbol{\zeta}_i^1 = \mathbf{0}$
    for $l = 1$ to $l^*$ do
    Share $\mathbf{x}^l_{i,n} + \boldsymbol{\delta}_i^l$ with neighbors if $i \in \mathcal{B}$
    $\mathbf{x}^{l+1}_{i,n} = \mathbf{x}^l_{i,n} - \alpha_n \left(\boldsymbol{\Omega}_{i,n}\mathbf{x}^l_{i,n} - \boldsymbol{\theta}_{i,n} + \lambda_{\text{tv}} \sum_{j\in\mathcal{N}_i} \text{sign}(\mathbf{x}^l_{i,n} - \tilde{\mathbf{x}}^l_{j,n})\right)$
    $\boldsymbol{\zeta}_i^{l+1} = \boldsymbol{\zeta}_i^l - \gamma_n \left(\boldsymbol{\zeta}_i^l - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,n}) + \lambda_{\text{tv}} \sum_{j\in\mathcal{N}_i} \text{sign}(\boldsymbol{\zeta}_i^l - \boldsymbol{\zeta}_j^l)\right)$
    end for
    $\hat{\mathbf{x}}_{i,n} = \mathbf{x}^{l^*}_{i,n}$
    $\mathbf{P}_{i,n} = (\mathsf{vec}_h^{-1}(\boldsymbol{\zeta}_i^{l^*}))^{-1}$
end for
```

with $\sigma_{\min}(\boldsymbol{\mathcal{A}})$ being the minimum non-zero singular value of $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\Omega}_{i,n}$ and $\boldsymbol{\theta}_{i,n}$ defined in (6.7); then, for the optimal solution $\bar{\mathbf{x}}^*_n$ in (6.4) and the optimal solution of the CKF problem $\hat{\mathbf{x}}^*_n$ in [105], we have $\bar{\mathbf{x}}^*_n = [\hat{\mathbf{x}}^*_n]_{i=1}^N$.[3]

Proof. The detailed proof is given in **P6**. □

The next step is to theoretically analyze the performance of the proposed solution in the presence of Byzantine agents. To this end, the following theorem characterizes the performance of the solution in (6.6) when Byzantine agents are present.

Theorem 6.2. Given the assumptions in Theorem 6.1 and $\lambda_{\text{tv}} \geq \lambda_0$, at each agent $i \in \mathcal{N}$ and the presence of Byzantine agents, the solution proposed in (6.6) stays in the neighborhood of the optimal solution $\bar{\mathbf{x}}^*_n = [\mathbf{x}^*_{i,n}]_{i=1}^N$ in (6.4) with radius

$$\lim_{l\to\infty} \mathbb{E}_l\{\|\mathbf{x}^{l+1}_{i,n} - \mathbf{x}^*_{i,n}\|^2\} \leq \frac{\Delta_0}{1 - \|\boldsymbol{\Delta}\|} \tag{6.12}$$

[3] The stacked vector $\mathbf{x} = [\mathbf{a}]_{i=1}^N \in \mathbb{R}^{Nm}$ corresponds to N times stacking the smaller vector $\mathbf{a} \in \mathbb{R}^m$ together.

where

$$\boldsymbol{\Delta} = \left(1 + 2\alpha_n^2 \|\boldsymbol{\Omega}_{i,n}\|^2 + 2\varepsilon\alpha_n\right)\mathbf{I} - 2\alpha_n \boldsymbol{\Omega}_{i,n}$$
$$\Delta_0 = \lambda_{\text{tv}}^2 \alpha_n (4\alpha_n + \frac{1}{\varepsilon})(4|\mathcal{R}_i|^2 + |\mathcal{B}_i|^2)m$$

with $0 \leq \varepsilon \leq \lambda_{\min}(\boldsymbol{\Omega}_{i,n})$, $\mathcal{R}_i$ and $\mathcal{B}_i$ denote the set of honest and Byzantine members of $\mathcal{N}_i$, and the step size α_n satisfies

$$\alpha_n \leq \min_{i \in \mathcal{N}} \left\{ \frac{\lambda_{\min}(\boldsymbol{\Omega}_{i,n}) - \varepsilon}{\|\boldsymbol{\Omega}_{i,n}\|^2} \right\}. \tag{6.13}$$

Proof. The detailed proof is given in **P6**. □

Remark 6.1. The error gap in (6.12) illustrates that the BR-DKF restricts the impact of attack amplitude entirely due to the sign$(\cdot)$ terms; however, the number of Byzantine agents still affects the error bound in (6.12) by altering Δ_0.

6.3 Numerical Simulations

The performance of the BR-DKF algorithm is illustrated by considering two network topologies, including a network of $N = 5$ agents as shown in Figure 6.1, and a randomly generated undirected connected network with $N = 25$ agents with the topology shown in Figure 6.6. The discrete-time system and agent parameters are considered similar to the work in [105], given by

$$\mathbf{x}_{n+1} = \begin{bmatrix} 0.4 & 0.9 & 0 & 0 \\ -0.9 & 0.4 & 0 & 0 \\ 0 & 0 & 0.5 & 0.8 \\ 0 & 0 & -0.8 & 0.5 \end{bmatrix} \mathbf{x}_n + \mathbf{w}_n,$$
$$\mathbf{y}_{i,n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_n + \mathbf{v}_{i,n},$$

where the state noise covariance $\mathbf{Q} = 0.1\mathbf{I}$, and the observation noise covariance $\mathbf{R}_i = \mathsf{diag}(0.1, 0.2, 0.3, 0.1)$. To benchmark our proposed algorithm, we evaluate the following scenarios: the centralized Kalman filter as CKF, distributed Kalman filter as DKF [105], DKF subject to Byzantine attack as B-DKF, and the BR-DKF subject to Byzantine attack. The subgradient solution for the state and error covariance are iterated for $l^* = 25$ iterations, and the results are averaged over 500 Monte Carlo experiments.

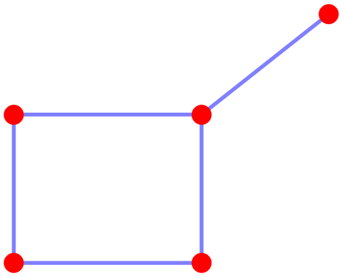


Figure 6.1: Network topology with $N = 5$ agents.

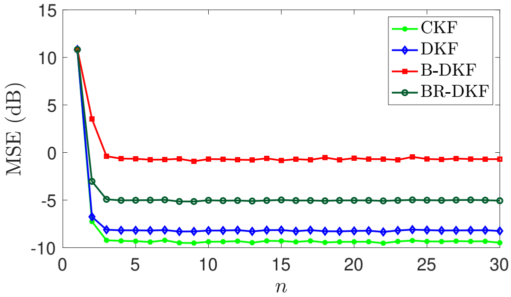


Figure 6.2: MSE versus filtering time index n in the network with $N = 5$ agents.

In the first scenario, we consider the network in Figure 6.1 comprising $N = 5$ agents, of which $B = 2$ are Byzantine agents, taken as the agents with the highest node degree. We plot the average MSE across agents, i.e.,

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_n - \hat{\mathbf{x}}_{i,n})^{\mathrm{T}}(\mathbf{x}_n - \hat{\mathbf{x}}_{i,n}), \tag{6.14}$$

as a performance measure. In the absence of Byzantines, the parameters of α_n, γ_n, and λ_{tv} of the BR-DKF are tuned to obtain the nearest possible MSE to the DKF algorithm. Even without a Byzantine attack, the BR-DKF does not reach the same performance as the DKF method; this is because the $\text{sign}(\cdot)$ terms in the updating process restrict the actual values of the state estimate. Here, Byzantine agents conduct a coordinated data falsification attack where $\boldsymbol{\Sigma}_i$ denotes the covariance matrix of perturbation sequences of agent $i \in \mathcal{B}$.

Figure 6.2 shows the MSE in (6.14) versus the filtering time index n in a network of $N = 5$ agents. The BR-DKF achieves lower MSE than the B-DKF under the same Byzantine attack, demonstrating its robustness. There is a performance gap between centralized and distributed Kalman filters, even without Byzantine agents,

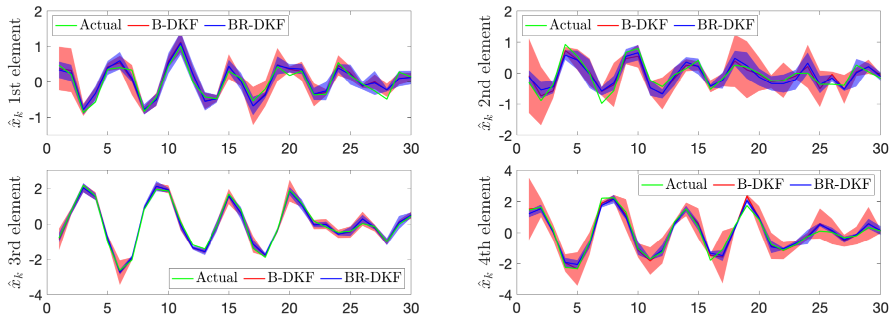


Figure 6.3: Estimation accuracy for different elements of the state in network of $N = 5$.

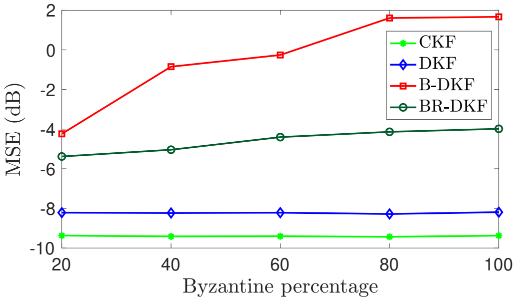


Figure 6.4: Steady-state MSE versus percentage of the Byzantine agents in the network with $N = 5$ agents.

due to the number of iterations of the subgradient solution. By increasing the number of l^*, the performance of the DKF will approach the CKF asymptotically. Figure 6.3 shows how the actual state of the network, with vector length $m = 4$, is closely estimated by various filtering methods. Tracking performance for different filtering settings is illustrated in shaded colors for all agents, and the average estimate among agents is shown as a solid line. We see that the BR-DKF outperforms the B-DKF by obtaining a lower estimation variance.

Figure 6.4 shows the MSE versus the percentage of Byzantine agents. We see that the BR-DKF is significantly less sensitive to the number of Byzantines than the B-DKF. Moreover, Figure 6.5 shows the MSE versus the trace of perturbation covariance of Byzantine agents. As shown, even without injecting any noise by the Byzantine agents, the MSE in the BR-DKF does not reach the DKF method because the sign$(\cdot)$ terms in the update equations limit the actual value of the state estimates. Upon starting the Byzantine attack, the obtained MSE under the B-DKF increases dramatically as more noise is injected, but the obtained MSE under

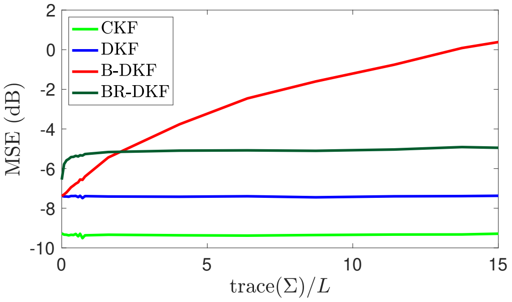


Figure 6.5: Steady-state MSE versus trace of the Byzantine agent attack covariance in the network with $N = 5$ agents.

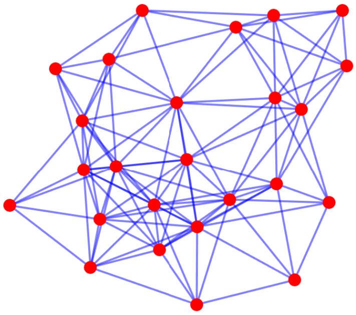


Figure 6.6: Network topology with $N = 25$ agents.

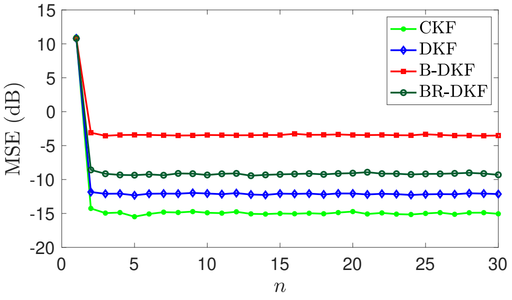


Figure 6.7: MSE versus filtering time index n in a network $N = 25$ agents.

the BR-DKF does not change. This is due to the restriction that the $\mathrm{sign}(\cdot)$ term provides, and as stated in *Remark 6.1*, the number of Byzantine agents is the only factor impacting the steady-state MSE in the BR-DKF.

In the second scenario, we consider a network of $N = 25$ agents as in Figure 6.6,

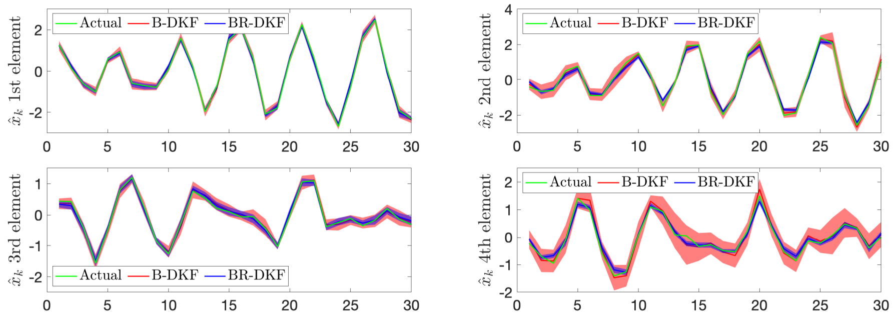


Figure 6.8: Estimation accuracy for different elements of the state in network of $N = 25$.

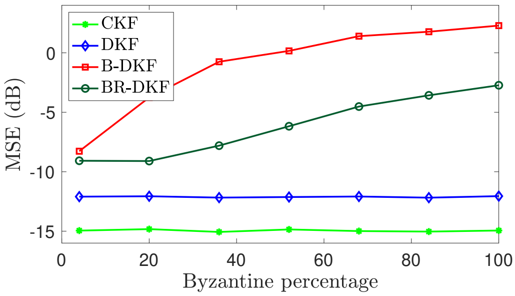


Figure 6.9: Steady-state MSE versus percentage of the Byzantine agents in the network with $N = 25$ agents.

including $B = 5$ Byzantine agents, which are chosen as network agents with the highest node degree. A similar tuning is made to the step size parameters to ensure the smallest difference in the MSE for DKF and BR-DKF algorithms in the absence of attack. In Figure 6.7, the MSE in (6.14) is plotted against the time index n. We see that the BR-DKF performs better than the B-DKF due to its lower MSE, which indicates its increased robustness to Byzantine attacks.

Similar to the previous scenario, the estimation accuracy for different state elements, with vector length $m = 4$, is shown in Figure 6.8. The estimated values of agents are plotted in shaded colors, and the average of the estimated values in solid colors. It can be seen that the BR-DKF reduces the variance of the state estimates and can robustly track the actual state of the network with higher accuracy than the B-DKF algorithm.

Figure 6.9 illustrates the MSE versus the percentage of Byzantine agents for different algorithms. A similar trend is observed, showing that the greater the percentage of Byzantine agents, the higher the MSE, while the BR-DKF sensitivity to the

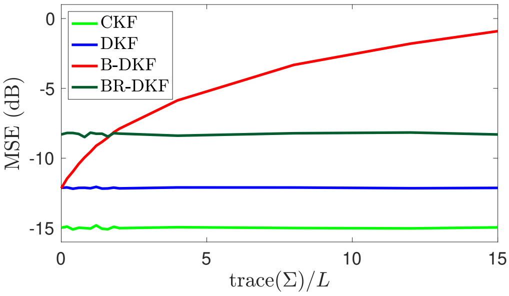


Figure 6.10: Steady-state MSE versus trace of the Byzantine agent attack covariance in the network with $N = 25$ agents.

Byzantine percentage is significantly less than the B-DKF. Moreover, Figure 6.10 illustrates the MSE versus the trace of the perturbation covariance of Byzantine agents. It shows that under the BR-DKF, as the trace of attack covariance is low, $\text{sign}(\cdot)$ terms in the state estimate update constrain the actual values and degrade the MSE compared to the DKF. However, when Byzantines inject more noise, the performance of the BR-DKF is not degraded, while the MSE of the B-DKF increases significantly.

6.4 Summary

This chapter modeled a DKF algorithm as a solution to a TV-norm penalized distributed optimization problem. It proposed a suboptimal solution for the resulting optimization problem and demonstrated that the optimization-based DKF solution is more robust to Byzantine attacks. Furthermore, it showed that the proposed suboptimal solution converges to a neighborhood of the optimal centralized solution within a bounded radius despite the presence of Byzantine agents. Lastly, it provided numerical simulations to corroborate the theoretical findings. Next, we will conclude the thesis and discuss its potential future extensions.

Chapter 7

Conclusions and Future Work

Local interaction between agents in distributed learning and estimation settings exposes private information to potential adversaries. In this thesis, we investigated privacy loopholes in such algorithms and proposed strategies to ensure agent privacy and provide resilience to adversaries. The main contribution of this thesis is organized into two subdivisions, namely threat analysis, and threat management. Threat analysis involves analyzing how different adversaries and attack strategies impact a network. It also involves identifying critical agents or links in the network through the evaluation of the optimal attack design from the perspective of an adversary. On the other hand, threat management focuses on developing strategies to limit the impact of potential adversaries on the network. The main goal of these strategies is to provide agent privacy and enhance attack robustness without significantly affecting performance. The desired strategy improves the privacy-performance tradeoffs without imposing additional burdens on agents.

As a prelude to discussing the proposed algorithms in this thesis and their potential future directions, it is essential to investigate the impact of improved threat analysis and management algorithms in the field. The primary objective of threat analysis is to identify potential adversaries and their capability to infer or falsify private data within a network. As described in Chapter 3, investigating the dynamics of the network from an adversarial perspective also enables us to gain a better understanding of the preferences and logic of the adversary when attacking a network. Thus, we can devise more effective strategies to reduce the impact of the adversary on the performance of the network. A more specific approach can also be used to identify and prevent information leaks in the network by investigating the available information of the adversary. In the DKF scenario, for instance, by knowing which parameters are in the interest of the adversaries and how specifically they falsify information traffic in the network, we can modify the local dynamics of the agents to minimize the impact of the adversaries, e.g., sharing only a fraction of information at a time.

On the other hand, threat management consists of all the measures taken to minimize the impact of adversaries in the network. In this thesis, we focus mainly on algorithms that work under the assumption that unidentified adversaries are present. This assumption makes the developed strategies effective in a variety of scenarios, both with and without information about adversaries. Even though the algorithms in Chapters 4, 5, and 6 are only evaluated against particular adversaries, they are capable of performing effectively regardless of the adversary type. The proposed attack-resilient algorithm in Chapter 6, for instance, can also deal with non-Gaussian falsification perturbations efficiently, while non-linear attacks might cause problems and require further investigation.

7.1 Summary and Future Directions

In the scope of threat analysis, Chapter 3 examined a distributed estimation scenario from the perspective of an adversary. By jointly optimizing attack sequences and the set of Byzantine agents, it characterized the optimal attack strategy to maximize the steady-state MSE. Later in the chapter, we examined the case where agents gained robustness to Byzantine attacks by sharing a fraction of their local information at each given time. Additionally, we designed an optimal attack strategy in which Byzantines cooperated on designing their attack covariances and the sequence of the information fractions they share. The next step in this research will involve specifying the protection mechanisms that agents use to reduce the impact of optimal attacks. For example, by assuming agents use a KL-based detector, the stealthiness constraints of attack design optimization problems are changed, and the problem becomes more similar to practical scenarios. A further extension to this research is to model the dynamic between Byzantine and regular agents as a game in which the regular and Byzantine agents constantly adjust their actions to minimize the impact of the other group on the network.

In the rest of the thesis, we mainly focused on threat management strategies. Chapter 4 investigated privacy breaches in the DKF settings and provided privacy by restricting and perturbing messages. Agent privacy was determined by the accuracy of the adversary in estimating the private data, where the higher the estimation error, the greater the privacy. We characterized the impact of privacy constraints on filtering performance and provided privacy bounds when EE and HBC adversaries were present. In Chapter 5, in addition to enhancing privacy, we attempted to improve the communication efficiency of agents. We proposed distributed learning algorithms that reduce inter-agent communication by partially sharing the information and provide privacy by using perturbation and state decomposition. Furthermore, later in Chapter 5, these algorithms were examined for privacy and performance in the presence of the HBC adversary.

In Chapters 4 and 5, perturbation and state decomposition were used to alter the average consensus steps while still achieving the exact average consensus among agents asymptotically. In practice, the number of consensus iterations is limited, and this results in performance degradation to the ACF. Thus, the future of this research will focus on a perturbation noise structure and state decomposition technique that does not compromise the performance of the system, even with limited consensus iterations. So far in this thesis, we have considered only the state estimates as private information, and the error covariances have been shared unprotected. Thus, analyzing the filtering performance and agent privacy bounds of the PP-DKF algorithm with perturbations to the error covariances is also an interesting future research direction. Additionally, this research can be further developed by answering the question of whether it is possible to design a perturbation noise structure that can be used in diffusion-based DKFs without sacrificing accuracy. Diffusion-based DKFs operate without an internal consensus loop; therefore, the same perturbation noise structure cannot be applied.

As another approach to threat management, in Chapter 6, we proposed a distributed Kalman filtering algorithm that provides robustness to Byzantine attacks without requiring agents to perform additional computations. In this method, the DKF algorithm was modeled as an optimization problem, and Byzantine attacks were restricted in variance by penalizing the objective function with a TV-norm term. We examined the impact of coordinated Byzantine attacks on the filtering performance and showed that the performance of the proposed algorithm is degraded by only the number of Byzantine agents. According to Figure 6.5 and 6.10, the BR-DKF performs poorly when the trace of the perturbation covariance is low. Thus, in the future, this research will focus on finding an adaptive solution for state estimation updates that reduces the impact of the $\text{sign}(\cdot)$ terms under low injected noise conditions. Furthermore, comparing different penalty terms for solving the optimization problem related to the state estimate and their response to Byzantine attacks is also an interesting direction to pursue. Additionally, the subgradient-based solution here is suboptimal; therefore, examining various methods to solve the optimization problem, such as the alternating direction method of multipliers (ADMM), can potentially improve the accuracy of the proposed algorithm.

Bibliography

[1] R. Olfati-Saber, “Distributed Kalman filtering and sensor fusion in sensor networks,” in *Netw. Embedded Sens. Control*, vol. 331, (Heidelberg, Germany), pp. 157–167, Springer, 2006.

[2] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *Proc. 46th IEEE Conf. Decis. and Control*, pp. 5492–5498, 2007.

[3] R. Olfati-Saber, “Distributed Kalman filter with embedded consensus filters,” in *Proc. 44th IEEE Conf. Decis. and Control*, pp. 8179–8184, 2005.

[4] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Approximate distributed Kalman filtering in sensor networks with quantifiable performance,” in *Proc. 4th IEEE Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, pp. 133–139, 2005.

[5] U. A. Khan and J. M. Moura, “Distributing the Kalman filter for large-scale systems,” *IEEE Trans. Signal Process.*, vol. 56, pp. 4919–4935, Oct. 2008.

[6] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Trans. Autom. Control*, vol. 55, pp. 2069–2084, Sept. 2010.

[7] R. Olfati, “Kalman-consensus filter: Optimality, stability, and performance,” in *Proc. 48th IEEE Conf. Decis. and Control*, pp. 7036–7042, 2009.

[8] S. Das and J. M. Moura, “Distributed Kalman filtering with dynamic observations consensus,” *IEEE Trans. Signal Process.*, vol. 63, pp. 4458–4473, Sept. 2015.

[9] S. Das and J. M. Moura, “Consensus+ innovations distributed Kalman filter with optimized gains,” *IEEE Trans. Signal Process.*, vol. 65, pp. 467–481, Jan. 2017.

[10] J. Qin, J. Wang, L. Shi, and Y. Kang, “Randomized consensus-based distributed Kalman filtering over wireless sensor networks,” *IEEE Trans. Autom. Control*, vol. 66, pp. 3794–3801, Aug. 2021.

[11] J. Le Ny and G. J. Pappas, “Differentially private filtering,” *IEEE Trans. Autom. Control*, vol. 59, pp. 341–354, Feb. 2014.

[12] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, “Distributed privacy-preserving data aggregation against dishonest nodes in network systems,” *IEEE Internet Things J.*, vol. 6, pp. 1462–1470, Apr. 2019.

[13] D. Kapetanovic, G. Zheng, and F. Rusek, “Physical layer security for massive MIMO: An overview on passive eavesdropping and active attacks,” *IEEE Commun. Mag.*, vol. 53, pp. 21–27, June 2015.

[14] R. K. Chang, “Defending against flooding-based distributed denial-of-service attacks: A tutorial,” *IEEE Commun. Mag.*, vol. 40, pp. 42–51, Oct. 2002.

[15] A. Vempaty, L. Tong, and P. K. Varshney, “Distributed inference with byzantine data: State-of-the-art review on data falsification attacks,” *IEEE Signal Process. Mag.*, vol. 30, pp. 65–75, Sept. 2013.

[16] F. Li and Y. Tang, “False data injection attack for cyber-physical systems with resource constraint,” *IEEE Trans. Cybern.*, vol. 50, pp. 729–738, Feb. 2020.

[17] L. Hu, Z. Wang, Q.-L. Han, and X. Liu, “State estimation under false data injection attacks: Security analysis and system protection,” *Elsevier Automatica*, vol. 87, pp. 176–183, Jan. 2018.

[18] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, “Optimal linear cyber-attack on remote state estimation,” *IEEE Control Netw. Syst.*, vol. 4, pp. 4–13, Mar. 2017.

[19] M. N. Kurt, Y. Yılmaz, and X. Wang, “Distributed quickest detection of cyber-attacks in smart grid,” *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 2015–2030, Aug. 2018.

[20] M. N. Kurt, Y. Yilmaz, and X. Wang, "Real-time detection of hybrid and stealthy cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 498–513, Feb. 2019.

[21] M. Aktukmak, Y. Yilmaz, and I. Uysal, "Sequential attack detection in recommender systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3285–3298, Apr. 2021.

[22] C.-Z. Bai, V. Gupta, and F. Pasqualetti, "On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds," *IEEE Trans. Autom. Control*, vol. 62, pp. 6641–6648, Dec. 2017.

[23] Y. Chen, S. Kar, and J. M. Moura, "Optimal attack strategies subject to detection constraints against cyber-physical systems," *IEEE Control Netw. Syst.*, vol. 5, pp. 1157–1168, Sept. 2017.

[24] X.-X. Ren, G. Yang, and X.-G. Zhang, "Statistical-based optimal-stealthy attack under stochastic communication protocol: An application to networked pmsm systems," *IEEE Trans. Ind. Electron.*, 2022.

[25] B. Kailkhura, S. Brahma, and P. K. Varshney, "Data falsification attacks on consensus-based detection systems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, pp. 145–158, March 2016.

[26] X. He, X. Ren, H. Sandberg, and K. H. Johansson, "How to secure distributed filters under sensor attacks," *IEEE Trans. Autom. Control*, vol. 67, no. 6, pp. 2843–2856, 2022.

[27] Y. Chen, S. Kar, and J. M. Moura, "Resilient distributed estimation: Sensor attacks," *IEEE Trans. Autom. Control*, vol. 64, pp. 3772–3779, Sept. 2018.

[28] Y. Chen, S. Kar, and J. M. Moura, "Resilient distributed parameter estimation with heterogeneous data," *IEEE Trans. Signal Process.*, vol. 67, pp. 4918–4933, Oct. 2019.

[29] J. G. Lee, J. Kim, and H. Shim, "Fully distributed resilient state estimation based on distributed median solver," *IEEE Trans. Autom. Control*, vol. 65, pp. 3935–3942, Sept. 2020.

[30] Y. Shi and Y. Wang, "Online secure state estimation of multiagent systems using average consensus," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, pp. 3174–3186, May 2022.

[31] M. Fauser and P. Zhang, “Resilient homomorphic encryption scheme for cyber-physical systems,” in *Proc. 60th IEEE Conf. Decis. and Control (CDC)*, pp. 5634–5639, 2021.

[32] Y. Ni, J. Wu, L. Li, and L. Shi, “Multi-party dynamic state estimation that preserves data and model privacy,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2288–2299, Jan. 2021.

[33] J. Zhou, W. Ding, and W. Yang, “A secure encoding mechanism against deception attacks on multi-sensor remote state estimation,” *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1959–1969, 2022.

[34] H. Lin, Z. T. Kalbarczyk, and R. K. Iyer, “Raincoat: Randomization of network communication in power grid cyber infrastructure to mislead attackers,” *IEEE Trans. Smart Grid*, vol. 10, pp. 4893–4906, Sept. 2019.

[35] A. Mitra and S. Sundaram, “Byzantine-resilient distributed observers for lti systems,” *Elsevier Automatica*, vol. 108, p. 108487, Oct. 2019.

[36] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos, “DETOX: A redundancy-based framework for faster and more robust gradient aggregation,” in *Proc. NIPS*, vol. 32, p. 10320–10330, 2019.

[37] P. Krishnamurthy and F. Khorrami, “Resilient redundancy-based control of cyber–physical systems through adaptive randomized switching,” *Systems & Control Letters*, vol. 158, p. 105066, Dec. 2021.

[38] A. Mitra, F. Ghawash, S. Sundaram, and W. Abbas, “On the impacts of redundancy, diversity, and trust in resilient distributed state estimation,” *IEEE Trans. Control Netw. Syst.*, vol. 8, pp. 713–724, June 2021.

[39] E. Nozari, P. Tallapragada, and J. Cortés, “Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design,” *Automatica*, vol. 81, pp. 221–231, Jul. 2017.

[40] M. Ruan, H. Gao, and Y. Wang, “Secure and privacy-preserving consensus,” *IEEE Trans. Autom. Control*, vol. 64, pp. 4035–4049, Oct. 2019.

[41] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, “Privacy-preserving average consensus: privacy analysis and algorithm design,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, pp. 127–138, Mar. 2019.

[42] Q. Li, R. Heusdens, and M. G. Christensen, “Privacy-preserving distributed optimization via subspace perturbation: A general framework,” *IEEE Trans. Signal Process.*, vol. 68, pp. 5983–5996, Oct. 2020.

[43] Y. Mo and B. Sinopoli, "Secure estimation in the presence of integrity attacks," *IEEE Trans. Autom. Control*, vol. 60, pp. 1145–1151, Apr. 2015.

[44] X. Ren, Y. Mo, J. Chen, and K. H. Johansson, "Secure state estimation with byzantine sensors: A probabilistic approach," *IEEE Trans. Autom. Control*, vol. 65, pp. 3742–3757, Sep. 2020.

[45] A.-Y. Lu and G.-H. Yang, "Distributed secure state estimation in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 66, pp. 2875–2882, Jun. 2021.

[46] X. Liu, Y. Mo, and E. Garone, "Local decomposition of Kalman filters and its application for secure state estimation," *IEEE Trans. Autom. Control*, vol. 66, pp. 5037–5044, Oct. 2020.

[47] J. He, L. Cai, and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 4069–4082, Jul. 2020.

[48] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, pp. 753–765, Feb. 2017.

[49] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Trans. Inf. Theory*, vol. 64, pp. 5677–5690, Aug. 2018.

[50] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, pp. 4711–4716, Nov. 2019.

[51] D. P. Mandic, S. Kanna, and A. G. Constantinides, "On the intrinsic relationship between the least mean square and Kalman filters [lecture notes]," *IEEE Signal Process. Mag.*, vol. 32, pp. 117–122, Nov. 2015.

[52] X. Ren, J. Wu, S. Dey, and L. Shi, "Attack allocation on remote state estimation in multi-systems: Structural results and asymptotic solution," *Automatica*, vol. 87, pp. 184 – 194, 2018.

[53] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Control Netw. Syst.*, vol. 4, pp. 4–13, Mar. 2017.

[54] C.-Z. Bai, V. Gupta, and F. Pasqualetti, "On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds," *IEEE Trans. Autom. Control*, vol. 62, pp. 6641–6648, Dec. 2017.

[55] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Worst-case stealthy innovation-based linear attack on remote state estimation," *Automatica*, vol. 89, pp. 117 – 124, Mar. 2018.

[56] R. Arablouei, S. Werner, Y.-F. Huang, and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, pp. 472–484, Jan. 2014.

[57] R. Arablouei, K. Doğançay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, pp. 3510–3522, Jul. 2014.

[58] J. Wang, R. Zhu, and S. Liu, "A differentially private unscented Kalman filter for streaming data in IoT," *IEEE Access*, vol. 6, pp. 6487–6495, Mar. 2018.

[59] K. H. Degue and J. Le Ny, "On differentially private Kalman filtering," in *Proc. 5th IEEE Global Conf. Signal and Inf. Process.*, pp. 487–491, 2017.

[60] J. Le Ny, "Differentially private Kalman filtering," in *Differential Privacy for Dynamic Data*, pp. 55–75, Springer, 2020.

[61] M. Fauser and P. Zhang, "Resilience of cyber-physical systems to covert attacks by exploiting an improved encryption scheme," in *Proc. 59th IEEE Conf. Decis. and Control*, pp. 5489–5494, 2020.

[62] J. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem," *IEEE Trans. Autom. Control*, vol. 24, pp. 266–269, April 1979.

[63] B. Rao, H. F. Durrant-Whyte, and J. Sheen, "A fully decentralized multisensor system for tracking and surveillance," *The International Journal of Robotics Research*, vol. 12, pp. 20–44, Feb. 1993.

[64] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, Jan. 2007.

[65] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th IEEE Int. Symp. Inf. Process. Sensor Netw.*, pp. 63–70, 2005.

[66] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conf. Decis. and Control*, pp. 2996–3000, 2005.

[67] S. P. Talebi and S. Werner, “Distributed Kalman filtering and control through embedded average consensus information fusion,” *IEEE Trans. Autom. Control*, vol. 64, pp. 4396–4403, Oct. 2019.

[68] L. Gao, S. Deng, and W. Ren, “Differentially private consensus with an event-triggered mechanism,” *IEEE Control Netw. Syst.*, vol. 6, pp. 60–71, March 2019.

[69] Z. Huang, S. Mitra, and G. Dullerud, “Differentially private iterative synchronous consensus,” in *Proc. ACM workshop on Privacy in the electronic society*, pp. 81–90, 2012.

[70] Y. Wang, J. Lam, and H. Lin, “Consensus of linear multivariable discrete-time multiagent systems: Differential privacy perspective,” *IEEE Trans. Cybern.*, Jan. 2022.

[71] R. Mendes and J. P. Vilela, “Privacy-preserving data mining: methods, metrics, and applications,” *IEEE Access*, vol. 5, pp. 10562–10582, June 2017.

[72] A. F. Westin, “Privacy and freedom,” *Washington and Lee Law Review*, vol. 25, no. 1, p. 166, 1968.

[73] C. Dwork, “Differential privacy: A survey of results,” in *Proc. Springer Int. conf. theory applications of models of computation*, pp. 1–19, 2008.

[74] I. Wagner and D. Eckhoff, “Technical privacy metrics: a systematic survey,” *ACM Comput. Surveys*, vol. 51, pp. 1–38, Jun. 2018.

[75] M. Lopuhaä-Zwakenberg, B. Škorić, and N. Li, “Information-theoretic metrics for local differential privacy protocols,” *arXiv preprint arXiv:1910.07826*, 2019.

[76] P. Braca, R. Lazzeretti, S. Marano, and V. Matta, “Learning with privacy in consensus + obfuscation,” *IEEE Signal Process. Lett.*, vol. 23, pp. 1174–1178, Sept. 2016.

[77] Y. Chen, S. Kar, and J. M. Moura, “Optimal attack strategies subject to detection constraints against cyber-physical systems,” *IEEE Control Netw. Syst.*, vol. 5, pp. 1157–1168, Sept. 2018.

[78] P. Srikantha, J. Liu, and J. Samarabandu, “A novel distributed and stealthy attack on active distribution networks and a mitigation strategy,” *IEEE Trans. Ind. Informat.*, vol. 16, pp. 823–831, Feb. 2020.

[79] M. Choraria, A. Chattopadhyay, U. Mitra, and E. G. Ström, "Design of false data injection attack on distributed process estimation," *IEEE Trans. Inf. Forensics Security*, vol. 17, Jan. 2022.

[80] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "Recursive filtering of distributed cyber-physical systems with attack detection," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, pp. 6466–6476, Oct. 2021.

[81] L. An and G.-H. Yang, "Byzantine-resilient distributed state estimation: A min-switching approach," *Elsevier Automatica*, vol. 129, p. 109664, July 2021.

[82] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, pp. 451–462, Feb. 2009.

[83] W. Yang, C. Yang, H. Shi, L. Shi, and G. Chen, "Stochastic link activation for distributed filtering under sensor power constraint," *Automatica*, vol. 75, pp. 109 – 118, Jan. 2017.

[84] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202 – 210, April 2017.

[85] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, pp. 57 – 77, Jan. 2016.

[86] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*, vol. 112. Springer Publishing Company, Incorporated, 2014.

[87] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Worst-case stealthy innovation-based linear attack on remote state estimation," *Elsevier Automatica*, vol. 89, pp. 117–124, Mar. 2018.

[88] M. Ruan, M. Ahmad, and Y. Wang, "Secure and privacy-preserving average consensus," in *Proc. Workshop Cyber-phys. Syst. Secur. Privacy*, pp. 123–129, 2017.

[89] X. Wang, J. He, P. Cheng, and J. Chen, "Privacy preserving average consensus with different privacy guarantee," in *Proc. Annu. Amer. Control Conf.*, pp. 5189–5194, 2018.

[90] C. Altafini, “A dynamical approach to privacy preserving average consensus,” in *Proc. 58th IEEE Conf. Decis. and Control*, pp. 4501–4506, 2019.

[91] Z. Huang, S. Mitra, and N. Vaidya, “Differentially private distributed optimization,” in *Proc. 16th Int. Conf. Distrib. Comput. and Netw.*, pp. 1–10, 2015.

[92] Q. Li, M. Coutino, G. Leus, and M. G. Christensen, “Privacy-preserving distributed graph filtering,” in *Proc. 28th IEEE Eur. Signal Process. Conf.*, pp. 2155–2159, 2021.

[93] Y. Song, C. X. Wang, and W. P. Tay, “Privacy-aware Kalman filtering,” in *Proc. 43rd IEEE Int. Conf. Acoust., Speech and Signal Process.*, pp. 4434–4438, 2018.

[94] J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, S.-L. Kim, and M. Debbah, “Communication-efficient and distributed learning over wireless networks: Principles and applications,” *Proc. of the IEEE*, vol. 109, pp. 796–819, May 2021.

[95] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, “Communication-efficient edge ai: Algorithms and systems,” *IEEE Commun. Surveys & Tutorials*, vol. 22, no. 4, pp. 2167–2191, 2020.

[96] Y. Yang, Z. Zhang, and Q. Yang, “Communication-efficient federated learning with binary neural networks,” *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 3836–3850, Dec. 2021.

[97] M. Asad, A. Moustafa, and T. Ito, “Fedopt: Towards communication efficiency and privacy preservation in federated learning,” *MDPI Applied Sciences*, vol. 10, p. 2864, April 2020.

[98] X. Ren, C.-M. Yu, W. Yu, X. Yang, J. Zhao, and S. Yang, “Dpcrowd: privacy-preserving and communication-efficient decentralized statistical estimation for real-time crowdsourced data,” *IEEE Internet Things J.*, vol. 8, pp. 2775–2791, Feb. 2020.

[99] W. Du, A. Li, P. Zhou, Z. Xu, X. Wang, H. Jiang, and D. Wu, “Approximate to be great: Communication efficient and privacy-preserving large-scale distributed deep learning in internet of things,” *IEEE Internet Things J.*, vol. 7, pp. 11678–11692, Dec. 2020.

[100] T. Li and L. Song, “Privacy-preserving communication-efficient federated multi-armed bandits,” *IEEE Journal on Selected Areas in Communications*, vol. 40, pp. 773–787, Mar. 2022.

[101] R. Lu, “A new communication-efficient privacy-preserving range query scheme in fog-enhanced iot,” *IEEE Internet Things J.*, vol. 6, pp. 2497–2505, April 2018.

[102] H. Mahdikhani, R. Lu, Y. Zheng, J. Shao, and A. A. Ghorbani, “Achieving o ($\log^3$n) communication-efficient privacy-preserving range query in fog-based iot,” *IEEE Internet Things J.*, vol. 7, pp. 5220–5232, June 2020.

[103] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Distributed LMS for consensus-based in-network adaptive processing,” *IEEE Trans. Signal Process.*, vol. 57, pp. 2365–2382, Feb. 2009.

[104] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Elsevier Systems & Control Letters*, vol. 53, pp. 65–78, Sept. 2004.

[105] K. Ryu and J. Back, “Distributed Kalman-filtering: Distributed optimization viewpoint,” in *Proc. 58th IEEE Conf. Decis. and Control*, pp. 2640–2645, 2019.

[106] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, “Robust distributed consensus using total variation,” *IEEE Trans. Autom. Control*, vol. 61, pp. 1550–1564, June 2016.

[107] J. Peng, W. Li, and Q. Ling, “Byzantine-robust decentralized stochastic optimization over static and time-varying networks,” *Elsevier Signal Process.*, vol. 183, p. 108020, June 2021.

[108] J. Peng, W. Li, and Q. Ling, “Variance reduction-boosted Byzantine robustness in decentralized stochastic optimization,” in *Proc. 47th IEEE Int. Conf. Acoust., Speech and Signal Process.*, pp. 4283–4287, 2022.

Appendix A

Publications in Chapter 3

P1: A. Moradi, N. K. D. Venkategowda and S. Werner, "Coordinated Data-Falsification Attacks in Consensus-based Distributed Kalman Filtering," in Proceedings 8th *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2019, pp. 495-499.

P2: A. Moradi, V. C. Gogineni, N. K. D. Venkategowda and S. Werner, "Distributed Filtering Design with Enhanced Resilience to Coordinated Byzantine Attacks," submitted to *IEEE Transactions on Signal Processing*, pp. 1-10, 2022.

P1 is available at https://doi.org/10.1109/CAMSAP45676.2019.9022448
P2 is awaiting publication and is not included

Coordinated Data-Falsification Attacks in Consensus-based Distributed Kalman Filtering

Ashkan Moradi, Naveen K. D. Venkategowda, Stefan Werner
Department of Electronic Systems, NTNU, Trondheim, Norway
Email: {ashkan.moradi, naveen.dv, stefan.werner}@ntnu.no

***Abstract*—This paper considers consensus-based distributed Kalman filtering subject to data-falsification attack, where Byzantine agents share manipulated data with their neighboring agents. The attack is assumed to be coordinated among the Byzantine agents and follows a linear model. The goal of the Byzantine agents is to maximize the network-wide estimation error while evading false-data detectors at honest agents. To that end, we propose a joint selection of Byzantine agents and covariance matrices of attack sequences to maximize the network-wide estimation error subject to constraints on stealthiness and the number of Byzantine agents. The attack strategy is then obtained by employing block-coordinate descent method via Boolean relaxation and backward stepwise based subset selection method. Numerical results show the efficiency of the proposed attack strategy in comparison with other naive and uncoordinated attacks.**

I. Introduction

The adoption of internet of things (IoT) is rapidly growing with applications in security, environmental monitoring, and smart infrastructure [1]. IoT employs distributed signal processing algorithms in which an individual agent exchanges information with its neighboring agents for inference tasks such as event detection, tracking, and parameter estimation. Limited computational and energy resources at the IoT devices and the distributed nature of IoT render them vulnerable to cybersecurity threats and malicious attacks from adversaries [2]. Thus, attack and defense mechanisms for secure distributed inference in IoT has garnered significant attention recently.

In data-falsification attacks, Byzantine agents inject malicious data or share manipulated information to decrease the system performance [3]. In such scenarios, the main challenges for distributed algorithms are trustworthiness of local information and resilient inference during attacks. To mitigate data-falsification attacks in distributed detection, adaptive design of local fusion rules to detect Byzantine agents was proposed in [4] and audit-bit based architecture where sensors transmit their decision via local groups in addition to direct communication with fusion center was presented in [5]. The authors in [6] proposed an attack detection procedure by employing reliable innovation data from the neighboring sensors in the distributed estimation process. In [7] weighted combination of local innovations and the information shared by neighbors is proposed for robust parameter estimation in presence of attacks. Joint attack detection and secure estimation methods have been proposed in [8] and [9]. The authors in [10] consider secure estimation for a networked cyber-physical system (CPS) under simultaneous false data injection and jamming attacks and propose a two-step attack detection mechanism and a measurement output model refinement to overcome the attacks.

On the other hand, knowledge of the optimal attack strategy and its impacts on the performance of IoT plays an important role in secure inference. It helps to understand the system behavior in presence of attacks, to identify critical links and agents, and to determine the regime in which the IoT no longer satisfies the operational goals. In this context, the trade-off between the detection performance with no attackers and the worst-case detection performance with an attacker was studied in [11] for hypothesis testing. For remote state estimation setting, optimal jamming policies for attacking the communication channels between sensors and fusion center to maximize the estimation error was proposed in [12] and optimal linear deception attack, which can successfully bypass a χ^2 false data detector, was presented in [13]. In [14] the mean square error (MSE) performance of single sensor Kalman filter with data-falsification attacks was analyzed considering the Kullback-Leibler (KL) divergence as a measure of attack stealthiness. Similarly, in [15] it was shown that with KL divergence as the stealth metric, the worst-case linear attack strategy that maximizes the estimation error covariance is a zero-mean Gaussian distributed attack sequence. In [16], the authors propose algorithms to design attack sequence to move the state of a CPS to a target state while satisfying the probability of detection constraints. These works [12]–[16] are limited to single sensor scenarios or centralized state estimation problems. Further, the performance and behavior of distributed state estimation with Byzantine agents are not addressed in the existing literature.

In this paper, we investigate the performance of consensus-based distributed Kalman filtering in presence of Byzantine agents. Assuming a linear attack model, we propose joint selection of Byzantine agents and their attack sequences that maximize the network-wide estimation error subject to constraints on stealthiness and the number of Byzantine agents. This results in an NP-hard optimization problem. Hence, we obtain suboptimal solutions by solving a sequence of semidefinite program (SDP) through the block-coordinate descent method and Boolean relaxation of the NP-hard optimization problem. To benchmark the proposed method, we present a backward stepwise subset selection based algorithm to determine the best set of Byzantine agents that maximizes the error.

Notations: Transpose and trace are denoted by $(\cdot)^T$ and $\mathsf{tr}(\cdot)$, the identity matrix of size n is represented by $\mathbf{I}_n$, the ones vector of length L is denoted by $\mathbf{1}_L$, whereas $\otimes$ denotes Kronecker product. Positive semidefinite matrix is represented by $\mathbf{A} \succeq 0$ and $\sup$ denotes the supremum. Matrices $\mathsf{diag}(\mathbf{a})$

This work was supported, in part, by the Research Council of Norway.

and $\mathsf{diag}(\{\mathbf{A}_i\}_{i=1}^L)$ denote diagonal and block-diagonal matrices whose respective diagonals are the elements of vector $\mathbf{a}$ and matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_L$.

II. System Model and Problem Formulation

Consider a connected multi-agent network of $L \in \mathbb{N}$ agents that collectively aim to estimate the state vector sequence $\{\mathbf{x}(k),\, k = 1, 2 \dots\}$ from local observations $\{\mathbf{y}_i(k),\, k = 1, 2 \dots, i = 1, 2 \dots, L\}$ at the agents. The network is modeled as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all agents of the network with $|\mathcal{V}| = L$, and $\mathcal{E}$ is the edge set that represents the communication links between the agents. The neighbor set $\mathcal{N}_i$ comprises all the agents that are connected to i within one hop and excludes the agent itself. The network adjacency matrix is denoted by $\mathbf{E}$ and the graph Laplacian is defined as $\mathbf{L} = \mathsf{diag}(\{|\mathcal{N}_i|\}_{i=1}^L) - \mathbf{E}$.

A. Distributed Filtering

The state vector and observation sequences at the ith agent are characterized by the state-space model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{w}(k) \\ \mathbf{y}_i(k) &= \mathbf{H}_i\mathbf{x}(k) + \mathbf{v}_i(k), \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times m}$ is the state-transition matrix, $\mathbf{H}_i \in \mathbb{R}^{n\times m}$ is the observation matrix at agent i, whereas $\mathbf{w}(k)$ and $\mathbf{v}_i(k)$ are mutually independent zero-mean Gaussian processes with covariance matrices $\mathbf{Q} \in \mathbb{R}^{m\times m}$ and $\mathbf{R}_i \in \mathbb{R}^{n\times n}$, respectively.

The agents employ the consensus-based distributed Kalman filter to estimate $\mathbf{x}(k)$ in a collaborative manner [17]. The state estimate at agent i is given by

$$\begin{aligned} \hat{\mathbf{x}}_i(k+1) = \mathbf{A}\hat{\mathbf{x}}_i(k) &+ \mathbf{K}_i(k)\big(\mathbf{y}_i(k) - \mathbf{H}_i\hat{\mathbf{x}}_i(k)\big) \\ &- \varepsilon\mathbf{A}\textstyle\sum_{j\in\mathcal{N}_i}\big(\hat{\mathbf{x}}_i(k) - \bar{\mathbf{x}}_j(k)\big), \end{aligned} \tag{2}$$

where $\mathbf{K}_i(k) \in \mathbb{R}^{m\times n}$ is the Kalman gain at agent i, ε is the consensus gain chosen as $0 \le \varepsilon \le 1/\max_i |\mathcal{N}_i|$, and $\{\bar{\mathbf{x}}_j(k)\}_{j\in\mathcal{N}_i}$ are estimates shared by the agents in the neighborhood set $\mathcal{N}_i$.

The optimal Kalman gain $\mathbf{K}_i(k)$ in (2) is found by minimizing the trace of the estimation error covariance $\mathbf{P}_i(k) \triangleq \mathbb{E}\{\mathbf{e}_i(k)\mathbf{e}_i^T(k)\}$, where the estimation error $\mathbf{e}_i(k)$ at agent i evolves as

$$\begin{aligned} \mathbf{e}_i(k+1) \triangleq \hat{\mathbf{x}}_i(k) - \mathbf{x}(k) = (\mathbf{A} - \mathbf{K}_i(k)\mathbf{H}_i)\mathbf{e}_i(k) - \mathbf{w}(k) \\ + \mathbf{K}_i(k)\mathbf{v}_i(k) - \varepsilon\mathbf{A}\textstyle\sum_{j\in\mathcal{N}_i}\big(\mathbf{e}_i(k) - \mathbf{e}_j(k)\big). \end{aligned} \tag{3}$$

After some calculations, the estimation error covariance can be expressed as

$$\begin{aligned} \mathbf{P}_i(k+1) = {} & \mathbf{F}_i(k)\mathbf{P}_i(k)\mathbf{F}_i^T(k) + \mathbf{Q} + \mathbf{K}_i(k)\mathbf{R}_i\mathbf{K}_i^T(k) \\ & - \varepsilon\mathbf{F}_i(k)\textstyle\sum_{s\in\mathcal{N}_i}\big(\mathbf{P}_i(k) - \mathbf{P}_{is}(k)\big)\mathbf{A}^T \\ & - \varepsilon\mathbf{A}\textstyle\sum_{r\in\mathcal{N}_i}\big(\mathbf{P}_i(k) - \mathbf{P}_{ri}(k)\big)\mathbf{F}_i^T(k) \\ & + \varepsilon^2\sum_{r\in\mathcal{N}_i}\sum_{s\in\mathcal{N}_i}\mathbf{A}\Big(\mathbf{P}_i(k) - \mathbf{P}_{is}(k) - \mathbf{P}_{ri}(k) + \mathbf{P}_{rs}(k)\Big)\mathbf{A}^T, \end{aligned} \tag{4}$$

where $\mathbf{P}_{ij}(k) \triangleq \mathbb{E}\{\mathbf{e}_i(k)\mathbf{e}_j^T(k)\}$ and $\mathbf{F}_i(k) = \mathbf{A} - \mathbf{K}_i(k)\mathbf{H}_i$. Thus, the optimal Kalman gain, which is found by differentiating the trace of (4) with respect to $\mathbf{K}_i(k)$, is given by

$$\mathbf{K}_i^*(k) = \mathbf{A}\Big(\mathbf{P}_i(k) - \varepsilon\sum_{j\in\mathcal{N}_i}\big(\mathbf{P}_i(k) - \mathbf{P}_{ji}(k)\big)\Big)\mathbf{H}_i^T\mathbf{M}_i^{-1}(k), \tag{5}$$

where $\mathbf{M}_i(k) = \mathbf{H}_i\mathbf{P}_i(k)\mathbf{H}_i^T + \mathbf{R}_i$.

B. Attack Model

In the following it is assumed that a subset $\mathcal{B} \subseteq \mathcal{V}$ with $|\mathcal{B}| \le L$ are Byzantine agents. In contrast to the “honest agents”, Byzantines share a falsified version of their state estimate with their neighbors to deteriorate the network-wide estimation performance [3]. Byzantine agent $j \in \mathcal{B}$ shares a modified state estimate $\hat{\mathbf{x}}_j(k) + \boldsymbol{\delta}_j(k)$ instead of $\hat{\mathbf{x}}_j(k)$ for $k \ge k_0$, where k_0 is the time instant when attack is initiated. Consequently, for $k \ge k_0$, the local estimates used for consensus building in (2) can be expressed as

$$\bar{\mathbf{x}}_j(k) = \begin{cases} \hat{\mathbf{x}}_j(k) + \boldsymbol{\delta}_j(k) & j \in \mathcal{B} \\ \hat{\mathbf{x}}_j(k) & j \notin \mathcal{B}, \end{cases} \tag{6}$$

where $\boldsymbol{\delta}_j(k) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_j)$ denotes the *data-falsification sequence*. Assuming a coordinated attack by the Byzantine agents, the augmented attack sequence across the network is given by

$$\boldsymbol{\delta}(k) \triangleq [\boldsymbol{\delta}_1^T(k), \boldsymbol{\delta}_2^T(k), \dots \boldsymbol{\delta}_L^T(k)]^T \tag{7}$$

and its covariance matrix is denoted by $\boldsymbol{\Sigma} = \mathbb{E}\{\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\}$. It assumed that the Byzantine agents have knowledge of the network and observation matrices. To maximize the attack stealthiness, $\boldsymbol{\delta}(k)$ is chosen as a zero-mean Gaussian sequence with covariance $\boldsymbol{\Sigma}$ [14]–[16]. The probability of attack-detection is proportional to the covariance of the attack sequence [14]–[16]. Therefore, $\boldsymbol{\Sigma}$ is limited to $\mathsf{tr}(\boldsymbol{\Sigma}) \le \eta$, where η captures the stealthiness of the attack.

C. Problem Statement

The main objective of the Byzantine attack is to maximize the network-wide mean squared error (NMSE) defined as

$$\mathsf{NMSE} = \limsup_{K\to\infty}\frac{1}{K}\textstyle\sum_{k=1}^K\sum_{i=1}^L \mathsf{tr}\big(\mathbf{P}_i(k)\big) \tag{8}$$

while still maintaining a desired level of stealthiness. Due to limited resources only a subset of agents can be Byzantines, which is denoted by $\mathcal{B}$. We need to decide the subset of agents that participate in the attack and determine the covariance matrices $\boldsymbol{\Sigma}_j$, $j \in \mathcal{B}$, of the corresponding falsification sequences. To that end, we introduce the Boolean variable $z_j = 1$ if $j \in \mathcal{B}$ and zero otherwise, and define the selection vector $\mathbf{z} = [z_1, z_2, \dots, z_L]^T$ [18]. The optimal attack strategy can be expressed as an optimization problem given by

$$\begin{aligned} \max_{\boldsymbol{\Sigma},\,\mathbf{z}}.\quad & \mathsf{NMSE} \\ \text{s. t.}\quad & \textstyle\sum_{j\in\mathcal{B}}\mathsf{tr}(\boldsymbol{\Sigma}_j) \le \eta, \\ & \boldsymbol{\Sigma} \succeq 0, \\ & \mathbf{z} \in \{0, 1\}^L, \quad \mathbf{1}^T\mathbf{z} = B, \end{aligned} \tag{9}$$

where the first constraint is related to the stealthiness, i.e., the ability to evade detection and the last constraint limits the number of Byzantine agents to $|\mathcal{B}| = B$. The parameter η is employed to restrict the total power of the falsification sequences and satisfy the detection-avoidance target.

In the next section, we compute the network-wide mean squared error as a function of the attack sequence covariance matrices and propose different methods for joint design of the attack sequence and subset of Byzantines with the aim of maximizing the error.

III. Joint Selection of Byzantine Agents and Design of Attack Sequences

To solve the problem in (9), we first derive the expression for the objective function to capture the NMSE. To that end, define the network-wide estimation error in presence of Byzantine attack after $k \geq k_0$ as

$$\bar{\mathbf{e}}(k) \triangleq [\bar{\mathbf{e}}_1^T(k), \bar{\mathbf{e}}_2^T(k), \ldots, \bar{\mathbf{e}}_L^T(k)]^T, \tag{10}$$

where the error at the ith agent is given by

$$\begin{aligned}\bar{\mathbf{e}}_i(k+1) = (\mathbf{A} - \mathbf{K}_i(k)\mathbf{H}_i)\bar{\mathbf{e}}_i(k) - \mathbf{w}(k) + \mathbf{K}_i(k)\mathbf{v}_i(k)\\ - \varepsilon\mathbf{A}\textstyle\sum_{j\in\mathcal{N}_i}\big(\bar{\mathbf{e}}_i(k) - \bar{\mathbf{e}}_j(k) - \boldsymbol{\delta}_j(k)\big).\end{aligned} \tag{11}$$

Defining $\boldsymbol{\Gamma} = \mathbf{E}\ \mathsf{diag}(\mathbf{z}) \otimes \mathbf{A}$, the evolution of the network estimation error can be expressed as

$$\bar{\mathbf{e}}(k+1) = \bar{\mathbf{A}}(k)\bar{\mathbf{e}}(k) + \bar{\mathbf{b}}(k) + \varepsilon\boldsymbol{\Gamma}\boldsymbol{\delta}(k), \tag{12}$$

where $\bar{\mathbf{A}}(k) = (\mathbf{I}_L - \varepsilon\mathbf{L})\otimes\mathbf{A} - \mathsf{diag}(\{\hat{\mathbf{K}}_i(k)\mathbf{H}_i\}_{i=1}^L)$, $\hat{\mathbf{K}}_i(k)$ is the Kalman gain assuming the statistics of the attack sequence is known, and

$$\bar{\mathbf{b}}(k) = \mathsf{diag}(\{\hat{\mathbf{K}}_i(k)\mathbf{v}_i(k)\}_{i=1}^L) - \mathbf{1}_L \otimes \mathbf{w}(k).$$

From (12), the covariance matrix of the error $\hat{\mathbf{P}}(k+1) \triangleq \mathbb{E}\{\mathbf{e}(k+1)\mathbf{e}^T(k+1)\}$ is given by

$$\hat{\mathbf{P}}(k+1) = \bar{\mathbf{A}}(k)\hat{\mathbf{P}}(k)\bar{\mathbf{A}}^T(k) + \bar{\mathbf{Q}}(k) + \varepsilon^2\boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T, \tag{13}$$

where $\bar{\mathbf{Q}}(k) = \mathsf{diag}(\{\hat{\mathbf{K}}_i(k)\mathbf{R}_i\hat{\mathbf{K}}_i^T(k)\}_{i=1}^L + \mathbf{1}_L\mathbf{1}_L^T \otimes \mathbf{Q}$. The optimal Kalman gain that minimizes $\mathsf{tr}(\hat{\mathbf{P}}(k))$ in (13) can obtained as

$$\hat{\mathbf{K}}_i(k) = \mathbf{A}\Big(\hat{\mathbf{P}}_i(k) - \varepsilon\sum_{j\in\mathcal{N}_i}\big(\hat{\mathbf{P}}_i(k) - \hat{\mathbf{P}}_{ji}(k)\big)\Big)\mathbf{H}_i^T\hat{\mathbf{M}}_i^{-1}(k), \tag{14}$$

where $\hat{\mathbf{M}}_i(k) = \mathbf{H}_i\hat{\mathbf{P}}_i(k)\mathbf{H}_i^T + \mathbf{R}_i$. In contrast to (4) and (5), (13) and (14) capture the error dynamics in presence of a Byzantine attack.

Assuming that the network is connected, $(\mathbf{A}, \bar{\mathbf{Q}}^{1/2})$ is controllable, and $(\mathbf{A}, \mathbf{H}_i)$ is observable, it can be shown that $\lim_{k\to\infty}\hat{\mathbf{P}}(k) = \hat{\mathbf{P}}$ i.e., $\hat{\mathbf{P}}(k)$ converges to a bounded value. In other words, there exists a matrix $\hat{\mathbf{K}}_i(k)$ such that $\hat{\mathbf{P}}(k)$ is bounded and converges to a unique positive definite matrix for all k and any initial non-negative symmetric matrix. Since obtaining a closed form expression for the covariance matrix of the actual error in (11) induced by the attack is intractable, we employ $\mathsf{tr}(\hat{\mathbf{P}})$ as a proxy to the objective function. Here $\mathsf{tr}(\hat{\mathbf{P}})$ is a lower bound for the actual NMSE.

The solution to the Riccati equation in (13) can be obtained from an SDP [19]. Motivated by this fact and substituting $\mathsf{NMSE} = \mathsf{tr}(\hat{\mathbf{P}})$ in (9), we express the joint Byzantine agent selection and attack design optimization problem as

$$\begin{aligned}\mathcal{P}:\quad & \max_{\mathbf{X},\boldsymbol{\Sigma},\mathbf{z}} && \mathsf{tr}(\mathbf{X})\\ & \text{s. t.} && \mathbf{X} \succeq \bar{\mathbf{A}}\mathbf{X}\bar{\mathbf{A}}^T + \bar{\mathbf{Q}} + \varepsilon^2\boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T,\\ &&& \boldsymbol{\Gamma} = \mathbf{E}\ \mathsf{diag}(\mathbf{z}) \otimes \mathbf{A},\\ &&& \mathbf{X} \succeq 0\\ &&& \textstyle\sum_{j\in\mathcal{B}}\mathsf{tr}(\boldsymbol{\Sigma}_j) \leq \eta, \quad \boldsymbol{\Sigma} \succeq 0,\\ &&& \mathbf{1}^T\mathbf{z} \leq B, \quad z_i \in \{0,1\}, \quad i = 1, \ldots L.\end{aligned} \tag{15}$$

The above problem is NP-hard [20], and difficult to solve due to the non-convex quadratic terms in the first constraint. In the subsequent sections we propose different methods to find a suboptimal solution to the above problem.

A. Block-Coordinate Descent (BCD) based Approach

The problem in (15) is non-convex due to the Boolean variables. To circumvent this, we relax the Boolean constraint $z_i \in \{0,1\}$ to a linear inequality constraint $0 \leq z_i \leq 1$. We see that for a given $\mathbf{z}$ or $\boldsymbol{\Sigma}$ the problem (15) is an SDP, as its first constraint is convex. Therefore, we employ the block-coordinate descent (BCD) method where $(\mathbf{X}, \boldsymbol{\Sigma})$ and $(\mathbf{X}, \mathbf{z})$ are alternately optimized with the other variable fixed. Applying the trace operator on both sides of the convergence constraint leads to a linear approximation with respect to $\mathbf{z}$ and $\boldsymbol{\Sigma}$. The proposed approach starts with an arbitrary $\mathbf{z}_0$ as initial condition and its first step is given by

$$\begin{aligned}\mathcal{P}_1:\quad & \max_{\mathbf{X},\boldsymbol{\Sigma}} && \mathsf{tr}(\mathbf{X})\\ & \text{s. t.} && \mathsf{tr}(\mathbf{X}) \succeq \mathsf{tr}(\bar{\mathbf{A}}\mathbf{X}\bar{\mathbf{A}}^T + \bar{\mathbf{Q}}) + \varepsilon^2\mathsf{tr}(\boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T),\\ &&& \mathbf{X} \succeq 0,\\ &&& \textstyle\sum_{j\in\mathcal{B}}\mathsf{tr}(\boldsymbol{\Sigma}_j) \leq \eta, \quad \boldsymbol{\Sigma} \succeq 0.\end{aligned} \tag{16}$$

The second step of the BCD approach is to determine the Byzantine agents by solving

$$\begin{aligned}\mathcal{P}_2:\quad & \max_{\mathbf{X},\mathbf{z}} && \mathsf{tr}(\mathbf{X})\\ & \text{s. t.} && \mathsf{tr}(\mathbf{X}) \succeq \mathsf{tr}(\bar{\mathbf{A}}\mathbf{X}\bar{\mathbf{A}}^T + \bar{\mathbf{Q}}) + \varepsilon^2\mathsf{tr}(\boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T),\\ &&& \boldsymbol{\Gamma} = \mathbf{E}\ \mathsf{diag}(\mathbf{z}) \otimes \mathbf{A},\\ &&& \mathbf{X} \succeq 0,\\ &&& \mathbf{1}^T\mathbf{z} \leq B, \quad 0 \leq z_i \leq 1, \quad i = 1, \ldots L.\end{aligned} \tag{17}$$

The subproblems (16) and (17) are convex and (16) has an unique solution for a given $\mathbf{z}$. Hence from [21, Theorem 1], we conclude that the proposed algorithm converges to a stationary point. The steps in (16) and (17) reduce the problem in (15) to that of solving a sequence of SDPs, which can be efficiently solved by interior-point methods.

The optimal $\mathbf{z}^* \in [0,1]^L$ is not Boolean due to the relaxation in (17). Hence, we recover a feasible solution $\mathbf{z}'$ of (15) by sorting the elements of $\mathbf{z}^*$ in descending order and set $z_i' = 1$ for the agents corresponding to the $|\mathcal{B}| = B$ largest elements.

Algorithm 1 Backward Stepwise Selection based Attack

```
Initialize: $\mathcal{B}_L = \mathcal{V}$,
for $j = L$ downto $B+1$ do
    Determine $l_j^* = \arg\max_{l \in \mathcal{B}_j} U(\mathcal{B}_j \backslash \{l\})$.
    Update $\mathcal{B}_{j-1} = \mathcal{B}_j \backslash \{l_j^*\}$.
end for
Set attack strategy $z_i = 1$ if $i \in \mathcal{B}_B$ else $z_i = 0$.
Find optimal attack sequence covariance matrix from (16).
```

B. Backward Stepwise Selection based Attack Strategy

For a given attack selection vector $\mathbf{z}$, the problem in (15) is an SDP. Hence, instead of relaxing the Boolean constraints, we employ an improved greedy search based method to determine the set of Byzantine agents and then find the corresponding optimal covariance matrices from (16). To select the Byzantine agents, we adopt the backward stepwise selection algorithm [22]. In this method, the algorithm begins by considering all agents as Byzantine i.e., $\mathcal{B} = \mathcal{V}$, and then iteratively removes the agent that contributes least to the overall objective. The algorithm stops when only B most effective agents are remaining. At iteration index j, let $\mathcal{B}_j$ denote the set of Byzantine agents with $|\mathcal{B}_j| = j$ and the corresponding performance of the network is defined as

$$U(\mathcal{B}_j) = \sum_{i \in \mathcal{V} \backslash \mathcal{B}_j} \mathsf{tr}(\hat{\mathbf{P}}_i) + \sum_{i \in \mathcal{B}_j} \mathsf{tr}(\hat{\mathbf{P}}_i), \tag{18}$$

which is computed from (16). The agent l_j^* that contributes lowest to the overall objective $U(\mathcal{B}_j)$ is removed from $\mathcal{B}_j$ at iteration j by determining l_j^* from

$$l_j^* = \arg\max_{l \in \mathcal{B}_j} U(\mathcal{B}_j \backslash \{l\}).$$

The algorithm is terminated when $\mathcal{B}_j$ consists of B agents and the attack sequence covariance matrix is determined from (16) with $z_i = 1$ if $i \in \mathcal{B}_B$ else $z_i = 0$. The proposed backward stepwise selection based attack strategy is summarized in Algorithm 1.

IV. Simulation Results

We consider a randomly generated undirected connected network with $L = 25$ sensor agents, maximum degree of $\Delta = 11$ and consensus gain $\varepsilon = 0.08$. The discrete time system and agent parameters are considered to be $\mathbf{A} = [0.6,\ 0.005;\ 0.25,\ 0.6]$, $\mathbf{Q} = \mathbf{I}_2$, $\mathbf{R}_i = \mathbf{I}_2$ and $\mathbf{H}_i = \mu_i \mathbf{I}_2$, $i = 1, 2, \cdots, L$ with $\mu_i \sim \mathcal{U}(0, 1)$. We set $N = 10$ iterations for the BCD method. The Byzantine agents start falsifying data at time index $k_0 = 20$ and the stealthiness parameter is set to $\eta' = \eta/|\mathcal{B}| = 15$ per Byzantine agent.

The proposed attack strategies are compared with two naive strategies, namely, random selection attack and uniform perturbation attack. The former strategy randomly selects the Byzantine agents, while the associate covariance matrices are obtained from (16). The latter strategy, choose the attack sequence covariance matrices as $\Sigma_j = \frac{P}{m}\mathbf{I}_m$ for all $j \in \mathcal{B}$ and the set of Byzantines are determined from (17). Fig. 1, illustrates the steady-state NMSE for the considered strategies, with $|\mathcal{B}| = 5$. It shows that the proposed methods significantly outperform the naive random and uniform attack strategies.The BCD based approach is computationally less intensive and performs close to the greedy search based method. It can be inferred that the covariance design influences the overall performance more in comparison with Byzantine agent selection.

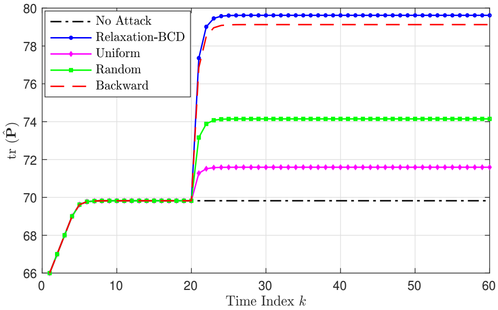


Fig. 1. NMSE for different attack strategies in a network with $L = 25$ agents, $B = |\mathcal{B}| = 5$ Byzantine agents, and stealthiness parameter $\eta' = \eta/B = 15$.

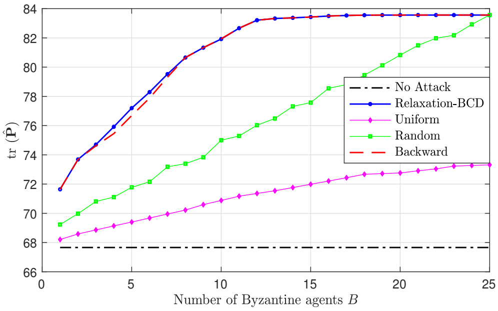


Fig. 2. NMSE versus number of Byzantine agents for a network with $L = 25$ agents and stealthiness parameter $\eta' = \eta/B = 15$.

Fig. 2 shows the NMSE versus the number of Byzantine agents for fixed stealthiness parameter $\eta' = \eta/|\mathcal{B}| = 15$ per Byzantine agent. We observe that the joint attack strategy performs close to the backward stepwise selection based method. When compared with random and uniform attack strategies, the proposed methods cause larger degradation in the NMSE for a fixed number of Byzantine agents.

V. Conclusion

This paper considered a distributed Kalman filter in presence of a coordinated data-falsification attack with Byzantine agents. It has been shown that the optimal set of Byzantine agents and covariance matrices of the falsification data that maximize the network-wide estimation error can be obtained by solving a sequence of semidefinite programs. Further, a greedy strategy for the Byzantine agent selection problem has been presented as an alternative to the Boolean relaxation based block-coordinate descent method. Simulation results demonstrate the efficacy of the proposed attack strategies in comparison with the naive approaches.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[2] Y. Chen, S. Kar, and J. M. F. Moura, "The internet of things: Secure distributed inference," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 64–75, Sep. 2018.

[3] A. Vempaty, L. Tong, and P. K. Varshney, "Distributed inference with Byzantine data: State-of-the-art review on data falsification attacks," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 65–75, Sep. 2013.

[4] B. Kailkhura, S. Brahma, and P. K. Varshney, "Data falsification attacks on consensus-based detection systems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 145–158, Mar. 2017.

[5] W. Hashlamoun, S. Brahma, and P. K. Varshney, "Audit bit based distributed bayesian detection in the presence of Byzantines," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 643–655, Dec. 2018.

[6] W. Yang, Y. Zhang, G. Chen, C. Yang, and L. Shi, "Distributed filtering under false data injection attacks," *Automatica*, vol. 102, pp. 34 – 44, 2019.

[7] Y. Chen, S. Kar, and J. M. F. Moura, "Resilient distributed estimation: Sensor attacks," *IEEE Trans. Autom. Control*, pp. 1–1, 2019.

[8] N. Forti, G. Battistelli, L. Chisci, S. Li, B. Wang, and B. Sinopoli, "Distributed joint attack detection and secure state estimation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 96–110, Mar. 2018.

[9] Y. Chen, S. Kar, and J. M. F. Moura, "Resilient distributed estimation through adversary detection," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2455–2469, May 2018.

[10] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 48–59, Mar. 2018.

[11] X. Ren, J. Yan, and Y. Mo, "Binary hypothesis testing with Byzantine sensors: Fundamental tradeoff between security and efficiency," *IEEE Trans. Signal Process.*, vol. 66, no. 6, pp. 1454–1468, Mar. 2018.

[12] X. Ren, J. Wu, S. Dey, and L. Shi, "Attack allocation on remote state estimation in multi-systems: Structural results and asymptotic solution," *Automatica*, vol. 87, pp. 184 – 194, 2018.

[13] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Control Netw. Syst.*, vol. 4, no. 1, pp. 4–13, Mar. 2017.

[14] C. Bai, V. Gupta, and F. Pasqualetti, "On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6641–6648, Dec. 2017.

[15] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Worst-case stealthy innovation-based linear attack on remote state estimation," *Automatica*, vol. 89, pp. 117 – 124, 2018.

[16] Y. Chen, S. Kar, and J. M. F. Moura, "Optimal attack strategies subject to detection constraints against cyber-physical systems," *IEEE Control Netw. Syst.*, vol. 5, no. 3, pp. 1157–1168, Sep. 2018.

[17] R. Olfati-Saber, "Kalman-consensus filter : Optimality, stability, and performance," in *Proc. of the 48th IEEE Conference on Decision and Control (CDC)*, Dec. 2009, pp. 7036–7042.

[18] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[19] W. Yang, C. Yang, H. Shi, L. Shi, and G. Chen, "Stochastic link activation for distributed filtering under sensor power constraint," *Automatica*, vol. 75, pp. 109 – 118, 2017.

[20] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202 – 210, 2017.

[21] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.

[22] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014, vol. 112.

Appendix B

Publications in Chapter 4

P3: A. Moradi, N. K. D. Venkategowda, S. P. Talebi and S. Werner, “Privacy-Preserving Distributed Kalman Filtering,” in *IEEE Transactions on Signal Processing*, vol. 70, pp. 3074-3089, June 2022.

P4: A. Moradi, N. K. D. Venkategowda, S. Pouria Talebi and S. Werner, “Distributed Kalman Filtering with Privacy against Honest-but-Curious Adversaries,” in Proceedings 55th *Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 790-794.

P5: A. Moradi, N. K. D. Venkategowda, S. Pouria Talebi and S. Werner, “Securing the Distributed Kalman Filter Against Curious Agents,” in Proceedings 24th *IEEE International Conference on Information Fusion*, 2021, pp. 1-7.

P3 is available at https://doi.org/10.1109/TSP.2022.3182590
P4 is available at https://doi.org/10.1109/IEEECONF53345.2021.9723222
P5 is available at https://doi.org/10.23919/FUSION49465.2021.9627034

Privacy-Preserving Distributed Kalman Filtering

Ashkan Moradi, *Member, IEEE*, Naveen K. D. Venkategowda, *Member, IEEE*,
Sayed Pouria Talebi, *Member, IEEE*, and Stefan Werner, *Senior Member, IEEE*

***Abstract*—Distributed Kalman filtering techniques enable agents of a multiagent network to enhance their ability to track a system and learn from local cooperation with neighbors. Enabling this cooperation, however, requires agents to share information, which raises the question of privacy. This paper proposes a privacy-preserving distributed Kalman filter (PP-DKF) that protects local agent information by restricting and obfuscating the information exchanged. The derived PP-DKF embeds two state-of-the-art average consensus techniques that guarantee agent privacy. The resulting PP-DKF utilizes noise injection-based and decomposition-based privacy-preserving techniques to implement a robust distributed Kalman filtering solution against perturbation. We characterize the performance and convergence of the proposed PP-DKF and demonstrate its robustness against the injected noise variance. We also assess the privacy-preserving properties of the proposed algorithm for two types of adversaries, namely, an external eavesdropper and an honest-but-curious (HBC) agent, by providing bounds on the privacy leakage for both adversaries. Finally, several simulation examples illustrate that the proposed PP-DKF achieves better performance and higher privacy levels than the distributed Kalman filtering solutions employing contemporary privacy-preserving techniques.**

***Index Terms*—Sensor networks, privacy, information fusion, average consensus, distributed Kalman filtering, multiagent systems.**

I. Introduction

THE proliferation of affordable sensor equipment with built-in networking capabilities has kindled a great deal of interest in distributed learning and estimation techniques in multiagent systems [1]–[6]. Furthermore, these systems incorporate honest communication with neighbors to enable cooperation and achieve a common target. In this work, we mainly focus on the distributed Kalman filtering techniques due to their computational efficiency, high accuracy, and the ability to model an extensive array of real-world physical systems. This broad applicability has made distributed Kalman filtering techniques a prominent fixture of multiagent learning and estimation applications in the signal processing community [7]–[11].

The distributed Kalman filtering techniques became more applicable to large-scale systems [10] and became widely used with the emergence of consensus filtering [12] and [13]. Kalman consensus filtering has a significant impact on the dynamic state estimation and was originally proposed in [8] and has been analyzed for stability and performance in [14]. The literature also includes a variety of consensus-based distributed Kalman filtering techniques to improve the performance in distributed estimation scenarios [6], [15], [16]. In the meantime, a diffusion-based strategy is proposed for distributed filtering and smoothing to estimate the state of linear dynamic systems in [11]. Generally, distributed Kalman filtering techniques rely on agents running local Kalman filtering operations using consensus filters to fuse observation and state vector information [9], [14]. On the one hand, sharing information among agents of the network facilitates cooperation between the agents. On the other hand, sharing of observation and state vector estimates gives rise to concerns about privacy [17], [18]; hence, there is a demand for secure filtering solutions [19], [20] and data aggregation [21]. Moreover, distributed filtering techniques are vulnerable to eavesdroppers that can potentially obtain private information by tapping communication links. This vulnerability turns privacy-preservation into an urgent requirement in many applications [22]–[32]. Also, privacy and security concerns become more pronounced when considering that even a single-agent infiltration can threaten the entire network integrity [25], [33].

The literature contains various methods that address the privacy issues in distributed processing problems, such as consensus [25]–[32], [34], optimization [22], [23], filtering [24], and state estimation [35]–[44]. A secure estimator is presented as a minimax optimization problem in the presence of a resource-limited attacker in [35], while the study in [36] detects the attacker by using χ^2 detectors to investigate the impact of intermittent data integrity attacks on Kalman filter-based estimators. By locating the misbehaving agents, [37] proposed a secure distributed state estimator based on a Gaussian mixture model detection mechanism, while [38] proposed a secure estimator that differentiates the malicious from the faulty agents. As opposed to detection-based secure state estimation, the work in [39] and [40] is designed to perform robustly in the presence of Byzantine agents without specifically detecting malicious agents. Additionally, to generate secure estimates, we can convert the problem of secure estimation into a distributed optimization problem [41]. A secure estimation scheme based on Kalman filters is proposed in [42], which fuses the local estimates securely using a quadratic programming approach. In [43], the authors propose a secure multi-party dynamic state estimation method based on Paillier encryption, while [44] investigates how to maximize privacy of stochastic dynamical systems with an information-theoretic privacy approach based on mutual information. Although these frameworks provide privacy, they are computationally demanding, and finding a secure and computationally efficient distributed state estimation remains a challenge.

When it comes to privacy concerns in distributed consensus areas, differential privacy is one of the main approaches [26]–

This work was supported in part by the Research Council of Norway.

A. Moradi, S. P. Talebi, and S. Werner are with the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim NO-7491 Norway, E-mail: {ashkan.moradi,pouria,stefan.werner}@ntnu.no.

N. K. D. Venkategowda is with the Department of Science and Technology, Linköping University, Sweden, E-mail: naveen.venkategowda@liu.se.

[28]. The differential privacy technique perturbs local message exchanges to protect individual information from being inferred by other agents or an external eavesdropper [26]–[28]. However, this privacy comes at a performance penalty. Among more recent consensus approaches, noise-injection-based methods [45], [46] have gained wide acceptance due to their improved privacy-accuracy trade-off. At the same time, decomposition-based techniques mainly focus on the amount of information exchanged between neighbors. For instance, in [47], [48], the initial state at each agent is decomposed into two substates, one for inter-node interactions and another that remains invisible to other agents.

Regarding privacy concerns in Kalman filtering settings, the work in [49] designs a differentially private Kalman filter in both input and output perturbation cases. Furthermore, differentially private Kalman filtering solutions that minimize the achieved mean squared error (MSE) under the differential privacy constraints are proposed in [19], [20], [50]. These works address the problem of releasing filtered signals that respect the privacy of individual data by employing differential privacy constraints over the filtering operations. In contrast, we apply privacy constraints to protect the value of agent-sensitive information from being estimated by adversaries. The proposed privacy-aware Kalman filter in [51] linearly transforms the sensor measurements before releasing them to the fusion center to maximize the estimation error for the private state and minimize that for the public state. Although considerable research has been devoted to privacy-preserving Kalman filtering solutions, no attention has been paid to a privacy-preserving framework for distributed Kalman filtering strategies.

In this paper, we assume that the local state estimates of individual agents are sensitive and must be kept private from adversaries. To that end, we propose a privacy-preserving distributed Kalman filter (PP-DKF) based on embedded average consensus that guarantees privacy via decomposition of local states and perturbation of the messages exchanged with neighboring agents. In the proposed approach, the local state at the agent is decomposed into private and public substates, where only public substates are shared with neighbors to reduce the amount of information exchanged. Furthermore, these shared messages are perturbed with a zero-mean Gaussian noise to further limit the information leakage. We show that the proposed DKF converges to unbiased steady-state estimates regardless of the initializing values or privacy-preserving perturbations. In addition, we provide rigorous mathematical analysis for the convergence behavior and the achievable MSE performance.

Next, we characterize the privacy performance of the proposed PP-DKF under two different adversaries, namely external eavesdroppers and honest-but-curious (HBC) agents. Defining the MSE of the estimate of the private information at the adversary as the privacy measure, we provide bounds on the privacy leakage for both adversaries. More importantly, we also derive the conditions under which perfect privacy can be achieved, i.e., conditions where there is no privacy leakage. Further, we show that the proposed PP-DKF achieves a better privacy-accuracy trade-off than state-of-the-art solutions, implying that PP-DKF achieves a higher state estimation accuracy for a given privacy level.

The rest of the paper is organized as follows. Section II provides preliminaries on distributed Kalman filtering and its vulnerability to internal and external adversaries. Section III presents the derivation of the proposed PP-DKF that protects private information through state decomposition and noise perturbation. In Section IV, the performance of the proposed PP-DKF is investigated in detail. In particular, we study the convergence of the PP-DKF, in the mean and mean-squared senses, for a finite number of consensus iterations and provide closed-form solutions incorporating state decomposition and noise perturbation effects. In Section V, we study the privacy guarantees provided by the PP-DKF when the network is subjected to external eavesdroppers and HBC agents. Section VI presents simulation results that corroborate our theoretical findings. Finally, conclusions are given in Section VII.

Mathematical Notations: Scalars, vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters, while $\mathbf{I}_l$, $\mathbf{0}_l$, and $\mathbf{1}_l$ represent an $l \times l$ identity matrix, an $l \times l$ zero matrix, and a column vector with l elements where all entries are one, respectively. The transpose and statistical expectation operators are denoted by $(\cdot)^{\mathrm{T}}$ and $\mathbb{E}\{\cdot\}$, while $\otimes$ denotes the matrix Kronecker product. The trace operator is denoted as $\mathsf{tr}(\cdot)$, whereas the $\mathsf{Blockdiag}(\{\mathbf{A}_i\}_{i=1}^{N})$ represents a block diagonal matrix containing $\mathbf{A}_i$s on the main diagonal. In order to distinguish between Kalman filtering operations and consensus filter iterations, consensus iterations are denoted in parenthesis and Kalman filtering time instants are denoted using subscripts, e.g., $\mathbf{x}_{i,n}(k)$ denotes the state at agent i and time instant n, after k consensus iterations. A white Gaussian sequence $\mathbf{x}(k)$ with covariance $\mathbf{\Sigma}$ is represented as $\mathbf{x}(k) \sim \mathscr{N}(\mathbf{0}, \mathbf{\Sigma})$, $\dagger$ denotes the Moore–Penrose pseudoinverse operator.

II. Background and Problem Formulation

This section revisits the classical distributed Kalman filtering problem of tracking a dynamic system state through observations from a network of sensors/agents. The network is modeled as a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with node set $\mathcal{N}$, representing agents, and edge set $\mathcal{E}$, representing bidirectional communication links. The neighborhood of node i, denoted by $\mathcal{N}_i$, is the set of nodes that agent i receives information from, which does not include agent i itself. The cardinality of the set $\mathcal{N}_i$ is denoted by N_i, while N is the number of agents in the network.

The state-space model, characterizing the state vector evolution and observation, is given by

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{v}_n \tag{1}$$

$$\mathbf{y}_{i,n} = \mathbf{H}_i\mathbf{x}_n + \mathbf{w}_{i,n} \tag{2}$$

where for time instant n and agent i, $\mathbf{A} \in \mathbb{R}^{m \times m}$ denotes the state transition matrix, $\mathbf{H}_i \in \mathbb{R}^{q \times m}$ denotes the observation matrix, $\mathbf{y}_{i,n} \in \mathbb{R}^q$ is the local observation, and $\mathbf{w}_{i,n} \in \mathbb{R}^q$ and $\mathbf{v}_n \in \mathbb{R}^m$, are observation and process noises, respectively. The process noise and observation noise are zero-mean

Algorithm 1 Distributed Kalman Filter

```
Initialization: For each agent $i \in \mathcal{N}$
  $\hat{\mathbf{x}}_{i,0|0} = \mathbb{E}\{\mathbf{x}_0\}$
  $\mathbf{M}_{i,0|0} = \mathbb{E}\left\{(\mathbf{x}_0 - \mathbb{E}\{\mathbf{x}_0\})(\mathbf{x}_0 - \mathbb{E}\{\mathbf{x}_0\})^{\mathrm{T}}\right\}$
Model update:
  $\hat{\mathbf{x}}_{i,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$
  $\mathbf{M}_{i,n|n-1} = \mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{C}_{\mathbf{v}_n}$
Measurement update:
  $\boldsymbol{\Gamma}_{i,n} = \mathbf{M}^{-1}_{i,n|n-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}\mathbf{H}_i$
  $\mathbf{M}^{-1}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \boldsymbol{\Gamma}_{j,n}\}$
  $\mathbf{G}_{i,n} = N\mathbf{M}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}$
  $\boldsymbol{r}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)$
  $\hat{\mathbf{x}}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \boldsymbol{r}_{j,n}\}$
```

Gaussian noise processes with a joint covariance matrix given by

$$\mathbb{E}\left\{\begin{bmatrix} \mathbf{v}_n \\ \mathbf{w}_{i,n} \end{bmatrix}\begin{bmatrix} \mathbf{v}_l^{\mathrm{T}} & \mathbf{w}_{j,l}^{\mathrm{T}} \end{bmatrix}\right\} = \begin{bmatrix} \mathbf{C}_{\mathbf{v}_n} & \mathbf{0}_{m\times q} \\ \mathbf{0}_{q\times m} & \mathbf{C}_{\mathbf{w}_{i,n}}\delta_{i,j} \end{bmatrix}\delta_{n,l}$$

with $\delta_{n,l}$ denoting the Kronecker delta function. The operations of the distributed Kalman filtering solution is summarized in Algorithm 1.

As can be seen from Algorithm 1, each agent first updates its local state estimate, where $\hat{\mathbf{x}}_{i,n|n-1}$ and $\hat{\mathbf{x}}_{i,n|n}$ are the respective *a priori* and *a posteriori* estimates of the state vector. Thereafter, the *a priori* covariance information at agent i and time instant n, denoted by $\mathbf{M}_{i,n|n-1} \in \mathbb{R}^{m\times m}$, is updated as

$$\boldsymbol{\Gamma}_{i,n} = \mathbf{M}^{-1}_{i,n|n-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}\mathbf{H}_i. \tag{3}$$

As shown in [3], the *a posteriori* centralized covariance information is the network average of the updates in (3). Hence, a distributed update of $\mathbf{M}^{-1}_{i,n|n}$ is obtained via an average consensus filter (ACF), wherein the agents refine their updates through local averaging within their neighborhoods. Finally, the *a posteriori* covariance $\mathbf{M}^{-1}_{i,n|n}$ is used to determine the local intermediate state estimate

$$\boldsymbol{r}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right) \tag{4}$$

which is, similar to $\boldsymbol{\Gamma}_{i,n}$, passed through an ACF to get the *a posteriori* state estimate $\hat{\mathbf{x}}_{i,n|n}$.

In particular, a generic iterative average consensus filter (ACF) is given by

$$\mathbf{S}_{i,n}(k) = q_{ii}\mathbf{S}_{i,n}(k-1) + \sum_{j\in\mathcal{N}_i} q_{ij}\mathbf{S}_{j,n}(k-1) \tag{5}$$

where consensus weights $\{q_{ij} : \forall i,j \in \mathcal{N}\}$ are positive real-valued weights so that the consensus weight matrix $\mathbf{Q}$ where $q_{ij} = [\mathbf{Q}]_{ij}$ is a doubly stochastic matrix. In Algorithm 1, we represent the general ACF with the following schematic [3]:

$$\mathbf{S}_{i,n}(k) \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i \cup i : \mathbf{S}_{j,n}(0)\} \tag{6}$$

where $\mathbf{S}_{j,n}(0)$, $j \in \mathcal{N}_i \cup i$ are the initial inputs to the ACF at node i, and $\mathbf{S}_{i,n}(k)$ is the output at node i after k iterations.

The shared intermediate state vector estimates $\boldsymbol{r}_{i,n} \in \mathbb{R}^m$ contain node-sensitive information that can be exploited by adversaries [49], [50]. We, therefore, need to modify the distributed Kalman filter (DKF) to protect node-sensitive information from possible privacy breaches. In what follows, we consider two types of adversaries, namely:

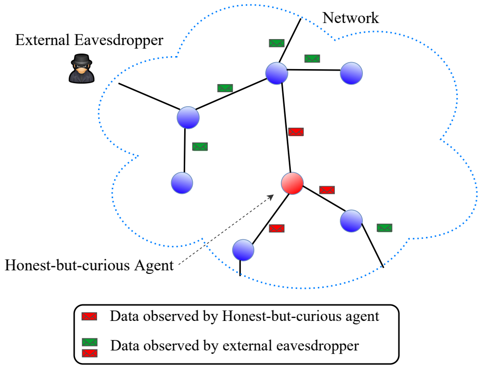

Network
External Eavesdropper
Honest-but-curious Agent
🟥 Data observed by Honest-but-curious agent
🟩🟥 Data observed by external eavesdropper

Fig. 1. Illustration of information accessible to external eavesdroppers and HBC agents.

- An *external eavesdropper*, who is external to the network, is trying to learn private information by accessing all the information exchanged between agents.
- An *HBC agent*, a legitimate node of the network, is contributing to the overall estimation task but, at the same time, passively attempts to infer private information from the messages shared by its immediate neighbors.

The two types of adversaries above can access different types and amounts of information; Fig 1 illustrates the different information types accessible to the adversaries and more details on their observation models is provided in Section V. In addition to the adversaries, the network includes regular agents that contribute to the overall estimation task without colluding with adversaries. Next, we propose a DKF that modifies the state messages exchanged by neighbors to induce privacy.

III. Privacy-Preserving Distributed Kalman Filter

In this section, we propose a PP-DKF based on the framework in [3]. In the distributed Kalman filtering setting, information leakage happens when agents share private information amongst each other. Without loss of generality, we will consider the local states, $\boldsymbol{r}_{i,n}$, private. We aim to protect the private information from being estimated by an adversary inside the network or an external eavesdropper. For this purpose, we decompose the agent states into public and private substates, where only noisy versions of the public substates are shared between neighbors.

The proposed PP-DKF tracks the dynamic system state by

$$\begin{aligned}\hat{\mathbf{x}}_{i,n|n-1} &= \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1} \\ \mathbf{M}_{i,n|n-1} &= \mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{C}_{\mathbf{v}_n}\end{aligned} \tag{7}$$

where, for agent i, $\hat{\mathbf{x}}_{i,n|n-1}$ and $\hat{\mathbf{x}}_{i,n|n}$ are the respective *a priori* and *a posteriori* state vector estimates. The intermediate

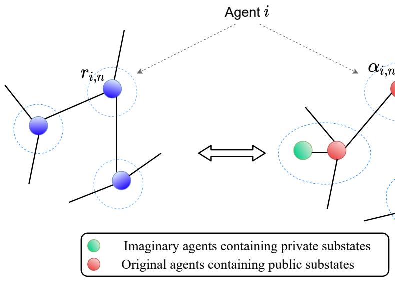

Agent i
$r_{i,n}$
$\alpha_{i,n}$
$\beta_{i,n}$
🟩 Imaginary agents containing private substates
🟥 Original agents containing public substates

Fig. 2. State decomposition representation of $\boldsymbol{r}_{i,n}$ to public substate $\boldsymbol{\alpha}_{i,n}$ and private substate $\boldsymbol{\beta}_{i,n}$.

information of agent i, at time instant n, denoted by $\mathbf{\Gamma}_{i,n}$, is updated as in (3), and shared with neighbors to reach average consensus. We assume that the condition for convergence of the covariance matrices $\{\mathbf{M}_{i,n|n} : \forall i \in \mathcal{N}, n = 1, 2, \ldots\}$ to unique stabilizing solutions, as given in [3], are satisfied. Therefore, we have $\lim_{n\to\infty} \mathbf{M}_{i,n|n} = \mathbf{M}_i$ for each $i \in \mathcal{N}$. Then, the average-consensus covariance matrix is employed to compute the intermediate state vector estimate of agent i as in (4), with the local gain matrix

$$\mathbf{G}_{i,n} = N\mathbf{M}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}.$$

The local state estimate is improved through local collaboration. As mentioned above, the local state, $\boldsymbol{r}_{i,n}$, is decomposed into a public substate $\boldsymbol{\alpha}_{i,n} \in \mathbb{R}^m$ and a private substate $\boldsymbol{\beta}_{i,n} \in \mathbb{R}^m$. Only a perturbed version of the public substate is shared among neighbors in the ensuing consensus process.

In particular, the proposed PP-DKF chooses the initial values $\boldsymbol{\alpha}_{i,n}(0)$ and $\boldsymbol{\beta}_{i,n}(0)$ randomly from the set of all real numbers in a manner that they satisfy the following relation [47]:

$$\frac{1}{2}(\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0)) = \boldsymbol{r}_{i,n} \tag{8}$$

where $\boldsymbol{r}_{i,n}$ is the ith agent initial information to start the privacy-preserving average consensus mechanism. The substate $\boldsymbol{\alpha}_{i,n}$ is the only value that is shared with neighbors, while substate $\boldsymbol{\beta}_{i,n}$ evolves internally and will not be observed by neighbors, as represented in Fig. 2. Although $\boldsymbol{\beta}_{i,n}$ remains invisible to neighbors, it directly affects the evolution of $\boldsymbol{\alpha}_{i,n}$.

In order to improve privacy preservation, we also inject noise into the messages shared by neighbors; see, e.g., [45]. To that end, each agent i shares a perturbed version of its public substate $\tilde{\boldsymbol{\alpha}}_{i,n}(k) = \boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\omega}_i(k)$, with noise sequence $\boldsymbol{\omega}_i(k) \in \mathbb{R}^m$, at each consensus iteration k. In particular, at consensus iteration k, each agent, i, perturbs its public substate with the following random noise vector

$$\boldsymbol{\omega}_i(k) = \begin{cases} \boldsymbol{\nu}_i(0) & k = 0 \\ \phi^k \boldsymbol{\nu}_i(k) - \phi^{k-1}\boldsymbol{\nu}_i(k-1) & \text{o.w.} \end{cases} \tag{9}$$

where $\phi \in (0, 1)$ is a common constant for all agents and $\boldsymbol{\nu}_i(k) \in \mathbb{R}^m \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_m)$ is an independent and identically distributed white Gaussian sequence for each k and $i \in \mathcal{N}$. At each consensus iteration k, agent i updates its local substates using the received neighbor messages as follows:

$$\begin{cases} \boldsymbol{\alpha}_{i,n}(k+1) = \boldsymbol{\alpha}_{i,n}(k) + \varepsilon \mathbf{U}_i(k)\left(\boldsymbol{\beta}_{i,n}(k) - \boldsymbol{\alpha}_{i,n}(k)\right) \\ \qquad + \varepsilon \sum\limits_{j \in \mathcal{N}_i} w_{ij}(k)\left(\tilde{\boldsymbol{\alpha}}_{j,n}(k) - \boldsymbol{\alpha}_{i,n}(k)\right) \\ \boldsymbol{\beta}_{i,n}(k+1) = \boldsymbol{\beta}_{i,n}(k) + \varepsilon \mathbf{U}_i(k)\left(\boldsymbol{\alpha}_{i,n}(k) - \boldsymbol{\beta}_{i,n}(k)\right) \end{cases} \tag{10}$$

where ε is the consensus step size, residing in $(0, \frac{1}{\Delta+1}]$ with $\Delta \triangleq \max_{i\in\mathcal{N}} N_i$. In (10), $w_{ij}(k) = w_{ji}(k)$ denotes the interaction weight of agents i and j, while $\mathbf{U}_i(k) \triangleq \mathsf{diag}(\mathbf{u}_i(k)) \in \mathbb{R}^{m\times m}$ is a diagonal matrix defined by the coupling weight vector $\mathbf{u}_i(k) \in \mathbb{R}^m$ of agent i. In particular, for $k = 0$, $w_{ij}(0) = w_{ji}(0)$ can be arbitrarily chosen from the set of all real numbers, while, for $k > 0$, we require that there exists a scalar $0 < \eta < 1$ such that all $w_{ij}(k) = w_{ji}(k)$, $j \in \mathcal{N}_i$ must reside in the range $[\eta, 1)$. This assumption ensures that each agent gives sufficient weight to the information received from its neighbors, including the private substates of the extended graph in Fig. 2. As a result, the information from each agent continuously affects the information of other agents over time. Similarly, for $\mathbf{u}_i(k)$, the elements of $\mathbf{u}_i(0)$ are independently chosen from the set of all real numbers, while, for $k > 0$, they are limited to $[\eta, 1)$. In the subsequent convergence analysis, we assume that the interaction and coupling weights are arbitrarily chosen at $k = 0$ and remain fixed for $k > 0$, while satisfying the weighting mechanism in [47]. For notational convenience, the interaction weights of the entire network is collected into matrix $\mathbf{W}(k) \triangleq [w_{ij}(k)] \in \mathbb{R}^{N\times N}$.

Finally, after repeating the steps in (10) for sufficient number of iterations, say K iterations, the local state estimate, $\hat{\mathbf{x}}_{i,n|n}$, is taken as

$$\hat{\mathbf{x}}_{i,n|n} = \boldsymbol{\alpha}_{i,n}(K) \;\; \forall i \in \mathcal{N}.$$

The operations of the proposed PP-DKF at each agent are summarized in Algorithm 2.

The privacy-preserving average consensus mechanism in (10), asymptotically converges to the exact average state estimate among agents. In particular, considering the convergence of the decomposition-based consensus operations in Appendix A, it can be shown that under the symmetric weight assumption for the interaction weight, the sum of all substates, defined as

$$\boldsymbol{\zeta}(k) = \sum_{i=1}^{N}(\boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\beta}_{i,n}(k)),$$

is preserved across the consensus iterations k, i.e., the sum of all substates are always time-invariant. This can be verified by simplifying $\boldsymbol{\zeta}(k)$ as

$$\boldsymbol{\zeta}(k) = \boldsymbol{\zeta}(0) + \varepsilon \sum_{i=1}^{N} d_i \left(\sum_{l=1}^{k-1} \boldsymbol{\omega}_i(l) \right) \tag{11}$$

with $d_i = \sum_{j\in\mathcal{N}_i} w_{ij}$ and showing that $\boldsymbol{\zeta}(k)$ converges to $\boldsymbol{\zeta}(0)$ in the mean square sense, i.e.,

$$\boldsymbol{\zeta}(k) \xrightarrow{\text{m.s.}} \boldsymbol{\zeta}(0) \Leftrightarrow \lim_{k\to\infty} \mathbb{E}\{\|\boldsymbol{\zeta}(k) - \boldsymbol{\zeta}(0)\|^2\} = 0\cdot$$

This is due to the connected network properties and assumptions of symmetric weights for $k \geq 0$, [47], [52], and decaying covariance of the noise sequences. Consequently, the substates will converge to the average of $\frac{1}{2N}\sum_{i=1}^{N}(\boldsymbol{\alpha}_{i,n}(k)+\boldsymbol{\beta}_{i,n}(k))$, which equals $\frac{1}{2N}\sum_{i=1}^{N}(\boldsymbol{\alpha}_{i,n}(0)+\boldsymbol{\beta}_{i,n}(0))$, and due to the initial condition $\boldsymbol{\alpha}_{i,n}(0)+\boldsymbol{\beta}_{i,n}(0)=2\boldsymbol{r}_{i,n}$, we have

$$\lim_{k\to\infty}\boldsymbol{\alpha}_{i,n}(k)=\lim_{k\to\infty}\boldsymbol{\beta}_{i,n}(k)=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{r}_{i,n}$$

that completes the convergence of substates to the desired average consensus value for each agent $i \in \mathcal{N}$.

Despite the above asymptotic performance guarantees, in practice, the number of consensus iterations is always finite; hence, questions arise concerning its consequences in filtering performance, convergence behavior, and resulting privacy. Therefore, it is imperative to examine the effect of injected noise and state decomposition on the proposed distributed Kalman filtering accuracy with a finite number of consensus iterations and the resulting privacy protection capabilities against internal and external adversaries. These topics are treated in detail in the following two sections.

Remark 1. Public and private substates $\boldsymbol{\alpha}_{i,n}(k)$ and $\boldsymbol{\beta}_{i,n}(k)$ are chosen randomly at $k=0$ such that $\boldsymbol{\alpha}_{i,n}(0)+\boldsymbol{\beta}_{i,n}(0)=\mathbf{r}_{i,n}$ and updated according to (10) for $k \geq 1$. Therefore, the intermediate state estimate $\mathbf{r}_{i,n}$ cannot be obtained by concatenating the public and private substates at each consensus iteration k.

IV. Kalman Filtering Performance Evaluation

In order to provide an intuitive analysis and a proper insight into the effects of incorporating the privacy-preserving mechanism, we commence our analysis with simplifying assumptions and subsequently generalize the results. Without loss of generality, it is assumed that agents initialize the privacy-preserving steps with equal substates, so that $\boldsymbol{\alpha}_{i,n}(0)=\boldsymbol{\beta}_{i,n}(0)$ for all $i \in \mathcal{N}$, and the noise added to the shared substate leaks into the private substate as well. This presents a worst-case scenario and upper-bounds the achievable MSE performance. Proceeding on the basis of Fig. 2, a network of $2N$ agents is considered so that each private substate corresponds to an agent only attached to its peer in the original network. In this case, to analyze the mean and mean-square performances of Algorithm 2, we consider the intermediate estimation error of agents in the decomposed network (see Fig. 2) as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n-\boldsymbol{\alpha}_{i,n}(0) \qquad i=1,\cdots,N\\ \boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n-\boldsymbol{\beta}_{i-N,n}(0) \quad i=N+1,\cdots,2N\end{aligned} \tag{12}$$

From the made assumption on the substates, we have $\boldsymbol{\alpha}_{i,n}(0)=\boldsymbol{\beta}_{i,n}(0)=\boldsymbol{r}_{i,n}$. Now, by substituting the intermediate state $\boldsymbol{r}_{i,n}$, from line 8 in Algorithm 2, and the local

Algorithm 2 Privacy-Preserving Distributed Kalman Filter

```
Initialization: For each agent $i \in \mathcal{N}$
 1: $\hat{\mathbf{x}}_{i,0|0} = \mathbb{E}\{\mathbf{x}_0\}$
 2: $\mathbf{M}_{i,0|0} = \mathbb{E}\left\{(\mathbf{x}_0-\mathbb{E}\{\mathbf{x}_0\})(\mathbf{x}_0-\mathbb{E}\{\mathbf{x}_0\})^{\mathrm{T}}\right\}$
Model update:
 3: $\hat{\mathbf{x}}_{i,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$
 4: $\mathbf{M}_{i,n|n-1} = \mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{C}_{\mathbf{v}_n}$
Measurement update:
 5: $\boldsymbol{\Gamma}_{i,n} = \mathbf{M}^{-1}_{i,n|n-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}\mathbf{H}_i$
 6: $\mathbf{M}^{-1}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \boldsymbol{\Gamma}_{j,n}\}$
 7: $\mathbf{G}_{i,n} = N\mathbf{M}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}$
 8: $\boldsymbol{r}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)$
Privacy-Preserving Mechanism:
 9: Select $\boldsymbol{\alpha}_{i,n}(0)$, and set $\boldsymbol{\beta}_{i,n}(0) = 2\boldsymbol{r}_{i,n} - \boldsymbol{\alpha}_{i,n}(0)$
10: Select weights $w_{ij}(k), \mathbf{u}_i(k), j \in \mathcal{N}_i$ and $k = 0,1,\cdots,K$
11: Share weights $w_{ij}(k)$, $j \in \mathcal{N}_i$ and $k = 0,1,\cdots,K$
12: Generate $\{\boldsymbol{\omega}_i(k), k = 0,1,\cdots,K\}$ based on (9)
13: Share $\tilde{\boldsymbol{\alpha}}_{i,n}(0) = \boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\omega}_i(0)$
14: for $k = 1$ to $K$ do
15:     Receive $\tilde{\boldsymbol{\alpha}}_{j,n}(k-1)$, $\forall j \in \mathcal{N}_i$
16:     Update $\boldsymbol{\alpha}_{i,n}(k)$ and $\boldsymbol{\beta}_{i,n}(k)$, as given in (10)
17:     Share $\tilde{\boldsymbol{\alpha}}_{i,n}(k) = \boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\omega}_i(k)$,
18: end for
19: $\hat{\mathbf{x}}_{i,n|n} = \boldsymbol{\alpha}_{i,n}(K)$
```

observation (2) into (12), the intermediate estimation error of each agent $i \in \{1,2,\cdots,2N\}$ is formulated as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n-\boldsymbol{r}_{i,n}\\ =& \mathbf{x}_n-\hat{\mathbf{x}}_{i,n|n-1}-N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\left(\mathbf{y}_{i,n}-\mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)\\ =& \mathbf{x}_n-\hat{\mathbf{x}}_{i,n|n-1}-N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i\left(\mathbf{x}_n-\hat{\mathbf{x}}_{i,n|n-1}\right)\\ & -N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{w}_{i,n}.\end{aligned} \tag{13}$$

Here, we assume that the imaginary agents $\{N+1,\cdots,2N\}$ employ the same observation parameters, $\mathbf{y}_{i,n}$, $\mathbf{H}_i$, and $\mathbf{C}_{\mathbf{w}_i}$, as their original peers. Substituting (1) into (13) and using the relation $\hat{\mathbf{x}}_{i,n|n-1}=\mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$ from (7), we have:

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \left(\mathbf{I}_m-N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i\right)\mathbf{A}\boldsymbol{\epsilon}_{i,n-1|n-1}\\ &+\left(\mathbf{I}_m-N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i\right)\mathbf{v}_n-N\ \mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{w}_{i,n}.\end{aligned} \tag{14}$$

where $\boldsymbol{\epsilon}_{i,n-1|n-1}=\mathbf{x}_{n-1}-\hat{\mathbf{x}}_{i,n-1|n-1}$. Considering the stacked vectors organizing all error terms as

$$\boldsymbol{\mathcal{E}}_n \triangleq [\boldsymbol{\epsilon}^{\mathrm{T}}_{1,n},\cdots,\boldsymbol{\epsilon}^{\mathrm{T}}_{2N,n}]^{\mathrm{T}} \in \mathbb{R}^{2Nm} \tag{15}$$

$$\boldsymbol{\mathcal{E}}_{n-1|n-1} \triangleq [\boldsymbol{\epsilon}^{\mathrm{T}}_{1,n-1|n-1},\cdots,\boldsymbol{\epsilon}^{\mathrm{T}}_{2N,n-1|n-1}]^{\mathrm{T}} \in \mathbb{R}^{2Nm} \tag{16}$$

and the state estimation error of the state-decomposed network after k consensus iterations, at each agent i, as $\boldsymbol{\epsilon}_{i,n|n,k}$, the stacked vector organizing all error terms of $\boldsymbol{\epsilon}_{i,n|n,k}$ after the privacy-preserving average consensus operations in (10), is denoted as

$$\boldsymbol{\mathcal{E}}_{n|n,k} = [\boldsymbol{\epsilon}^{\mathrm{T}}_{1,n|n,k},\cdots,\boldsymbol{\epsilon}^{\mathrm{T}}_{2N,n|n,k}]^{\mathrm{T}} \in \mathbb{R}^{2Nm}.$$

Due to notational convenience, we are no longer including the index k in error parameters, and the stacked vector estimation error can be computed as

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} &= \mathbf{G}^k\boldsymbol{\mathcal{E}}_n + \phi^{k-1}\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-1) \\ &\quad + \sum_{s=2}^{k}\phi^{k-s}\left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-s)\end{aligned} \tag{17}$$

where $\boldsymbol{\nu}(k) = [\boldsymbol{\nu}_1^{\mathrm{T}}(k), \cdots, \boldsymbol{\nu}_N^{\mathrm{T}}(k)]^{\mathrm{T}}$, $\boldsymbol{\mathcal{B}} = \varepsilon[\mathbf{W}, \mathbf{W}]^{\mathrm{T}} \otimes \mathbf{I}_m \in \mathbb{R}^{2Nm\times Nm}$, and $\mathbf{G} \in \mathbb{R}^{2Nm\times 2Nm}$ is a doubly stochastic matrix given by

$$\mathbf{G} = \begin{bmatrix}\mathbf{M} & \varepsilon\mathbf{U} \\ \varepsilon\mathbf{U} & \mathbf{I}_{Nm} - \varepsilon\mathbf{U}\end{bmatrix} \tag{18}$$

with $\mathbf{M} \triangleq (\mathbf{I}_N - \varepsilon(\mathbf{D} - \mathbf{W})) \otimes \mathbf{I}_m - \varepsilon\mathbf{U}$, $\mathbf{U} = \text{Blockdiag}(\{\mathbf{U}_i\}_{i=1}^N)$, and $\mathbf{D} \triangleq \text{diag}(\{\sum_{j\in\mathcal{N}_i} w_{ij}\}_{i=1}^N)$. To simplify the state vector estimation error analysis, we assume that the interaction and coupling weight matrices are time-invariant. Substituting the network-wide intermediate state vector estimation error $\boldsymbol{\mathcal{E}}_n$ from (14) into (17) results

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} =& \boldsymbol{\mathcal{P}}_k\boldsymbol{\mathcal{E}}_{n-1|n-1} + \boldsymbol{\mathcal{Q}}_k\boldsymbol{\Upsilon}_n - \boldsymbol{\Omega}_{n,k} + \phi^{k-1}\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-1) \\ &+ \sum_{s=2}^{k}\phi^{k-s}\left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-s)\end{aligned} \tag{19}$$

where $\boldsymbol{\Upsilon}_n = [\mathbf{v}_n^{\mathrm{T}}, \cdots, \mathbf{v}_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2Nm}$ and

$$\begin{aligned}\boldsymbol{\mathcal{P}}_k &= \mathbf{G}^k\text{Blockdiag}(\{\mathbf{P}_i\mathbf{A}\}_{i=1}^{2N}) \\ \boldsymbol{\mathcal{Q}}_k &= \mathbf{G}^k\text{Blockdiag}(\{\mathbf{P}_i\}_{i=1}^{2N}) \\ \boldsymbol{\Omega}_{n,k} &= \mathbf{G}^k\text{Blockdiag}(\{\mathbf{Q}_i\}_{i=1}^{2N})[\mathbf{w}_{1,n}^{\mathrm{T}}, \cdots, \mathbf{w}_{2N,n}^{\mathrm{T}}]^{\mathrm{T}}\end{aligned}$$

with $\mathbf{P}_i = \mathbf{I}_m - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{H}_i$ and $\mathbf{Q}_i = \mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}$. Assuming the mutual independence of the noise sequences $\mathbf{w}_{i,n}$, $\mathbf{v}_n$, and $\boldsymbol{\nu}_i(k)$ for all $n = 1, 2, \cdots$, $i \in \mathcal{N}$, and $k \in [1, K]$, the recursive expression of the state vector estimation error in (19), is used to formulate the second-order statistics of all agents, denoted by $\boldsymbol{\Sigma}_{n,k} = \mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\boldsymbol{\mathcal{E}}_{n|n}^{\mathrm{T}}\} \in \mathbb{R}^{2Nm\times 2Nm}$, as

$$\boldsymbol{\Sigma}_{n,k} = \boldsymbol{\mathcal{P}}_k\boldsymbol{\Sigma}_{n-1,k}\boldsymbol{\mathcal{P}}_k^{\mathrm{T}} + \boldsymbol{\mathcal{Q}}_k\mathbf{C}_{\boldsymbol{\Upsilon}}\boldsymbol{\mathcal{Q}}_k^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\Omega}_k} + \boldsymbol{\mathcal{T}}_k \tag{20}$$

where $\mathbf{C}_{\boldsymbol{\Upsilon}} = \mathbb{E}\{\boldsymbol{\Upsilon}_n\boldsymbol{\Upsilon}_n^{\mathrm{T}}\}, \mathbf{C}_{\boldsymbol{\Omega}_k} = \mathbb{E}\{\boldsymbol{\Omega}_{n,k}\boldsymbol{\Omega}_{n,k}^{\mathrm{T}}\} \in \mathbb{R}^{2Nm\times 2Nm}$, and given k consensus iterations

$$\boldsymbol{\mathcal{T}}_k = \sum_{s=2}^{k}\phi^{2(k-s)}\bar{\boldsymbol{\mathcal{T}}}_s + \phi^{2(k-1)}\boldsymbol{\mathcal{B}}\mathbf{C}_{\boldsymbol{\nu}}\boldsymbol{\mathcal{B}}^{\mathrm{T}} \tag{21}$$

with $\mathbf{C}_{\boldsymbol{\nu}} = \mathbb{E}\{\boldsymbol{\nu}(s)\boldsymbol{\nu}^{\mathrm{T}}(s)\} \in \mathbb{R}^{Nm\times Nm}$ at each consensus iteration s and $\bar{\boldsymbol{\mathcal{T}}}_s = \left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)\boldsymbol{\mathcal{B}}\mathbf{C}_{\boldsymbol{\nu}}\boldsymbol{\mathcal{B}}^{\mathrm{T}}\left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)^{\mathrm{T}}$.

Due to the doubly stochastic matrix $\mathbf{G}$ and similar to [3], $\mathbf{P}_i$ and $\mathbf{A}$ are stable, $\boldsymbol{\mathcal{P}}_k$ is stable; therefore, $\boldsymbol{\Sigma}_{n,k} \to \boldsymbol{\Sigma}_k$ as $n \to \infty$, where $\boldsymbol{\Sigma}_k$ is the solution of the discrete time Lyapunov equation in (20) that represents the MSE convergence of the filtering performance. The effect of injected noise, considering a privacy-preserving average consensus with k consensus iterations, is manifested in $\boldsymbol{\mathcal{T}}_k$. It degrades the steady-state MSE of Algorithm 2 compared to the non-private approach and introduces a performance-privacy trade-off. On the other hand, taking the statistical expectation of (19) yields

$$\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\} = \boldsymbol{\mathcal{P}}_k\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n-1|n-1}\} = \boldsymbol{\mathcal{P}}_k^n\mathbb{E}\{\boldsymbol{\mathcal{E}}_{0|0}\}.$$

Once again, since $\boldsymbol{\mathcal{P}}_k$ is stable, we have $\lim_{n\to\infty}\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\} = \mathbf{0}$ that indicates the steady-state estimates are unbiased regardless of their initializing values or privacy-preserving perturbations. The effect of injected noise, considering a privacy-preserving average consensus with k consensus iterations, is manifested in $\boldsymbol{\mathcal{T}}_k$, which degrades the steady-state MSE of Algorithm 2 compared to the non-private approach, introducing a performance-privacy trade-off.

For the case where agents start the privacy-preserving steps with different initial substates, one can claim that the imaginary agents that hold the private substates, demonstrated in Fig. 2, are perturbed by noise sequence with vanishing covariance. In the privacy-preserving mechanism, the private substates affect the updating equations without being perturbed; this will reduce the effect of term $\boldsymbol{\mathcal{T}}_k$ in the corresponding Lyapunov equation, resulting in improved MSE performance without affecting the convergence. This trade-off is shown using numerical simulation examples in Section VI. Next, we evaluate the privacy guarantees of the PP-DKF for the cases of internal and external adversaries.

V. Privacy Analysis

This section provides a comprehensive privacy analysis of the PP-DKF for two different adversaries: an external eavesdropper and an honest-but-curious (HBC) agent. The state estimate $\mathbf{r}_{j,n}$ is considered private since it corresponds to the local *a posteriori* estimate and includes more node-specific information than the global *a posteriori* state estimate $\hat{\mathbf{x}}_{j,n|n}$. As an output of the ACF, the *a posteriori* state estimate $\hat{\mathbf{x}}_{j,n|n}$ has the same value among agents, therefore it contains less local information about the agents. Similar to [45], [53], we assume that the adversary employs an estimator to infer the states of the agents $\boldsymbol{r}_{j,n}$, $j = 1, 2, \ldots, N$ at time n and consider the MSE of the estimator as the privacy metric. The MSE metric is used here to measure how accurately the adversary can estimate the exact value of the initial local *a posteriori* state estimates given a specific attack model and information available to the adversary. Let $\hat{\boldsymbol{r}}_{j,n}(k)$ denote the estimate of the state of agent j at the adversary at time n after k consensus iterations and the corresponding privacy loss $\mathcal{E}_{j,n}(k)$ is the MSE given by

$$\mathcal{E}_{j,n}(k) \triangleq \text{tr}\left(\mathbb{E}\{(\boldsymbol{r}_{j,n} - \hat{\boldsymbol{r}}_{j,n}(k))(\boldsymbol{r}_{j,n} - \hat{\boldsymbol{r}}_{j,n}(k))^{\mathrm{T}}\}\right). \tag{22}$$

A. External eavesdropper

We assume that the external eavesdropper knows the network topology and can access all information exchanged by the agents with their neighbors. As can be seen from Algorithm 2, the messages exchanged after k consensus iterations form the following information set at the eavesdropper

$$\mathcal{I}_E(k) = \{\tilde{\boldsymbol{\alpha}}_{j,n}(l), w_{ij}(l), \forall i, j \in \mathcal{N}, l = 0, 1, \ldots, k\} \tag{23}$$

where $\tilde{\boldsymbol{\alpha}}_{j,n}(l)$ is the perturbed state and $w_{ij}(l)$ is the interaction weights exchanged with the neighbors. The eavesdropper estimates the states of the agents $\hat{\boldsymbol{r}}_{j,n}(k)$ $\forall j \in \mathcal{N}$ by constructing an observer at each consensus iteration using the information set (23). Under this adversarial model, the proposed filtering Algorithm 2 is privacy-preserving.

Theorem 1. *If the external eavesdropper can only access messages shared by the agents, Algorithm 2 is privacy-preserving and the privacy leakage for agent j is given by*

$$\mathcal{E}_j = \lim_{n\to\infty}\lim_{k\to\infty}\mathcal{E}_{j,n}(k) = \textit{tr}\left(\left(\mathbf{e}_j^T\otimes\mathbf{I}_m\right)\tilde{\boldsymbol{\mathcal{L}}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{\mathcal{L}}}^T\left(\mathbf{e}_j\otimes\mathbf{I}_m\right)\right) \tag{24}$$

where $\mathbf{e}_j \in \mathbb{R}^N$ is a vector with 1 in the jth entry and zeros elsewhere, $\tilde{\boldsymbol{\Sigma}}$ is the stabilizing solution for (20)*, $\tilde{\boldsymbol{\mathcal{L}}} = \frac{1}{2}\boldsymbol{\mathcal{L}} - \varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\boldsymbol{\Lambda}$, $\boldsymbol{\Lambda} = \boldsymbol{\Theta}\,diag(\frac{1}{1-\lambda_1}, \frac{1}{1-\lambda_2}, 1, \cdots, 1)\boldsymbol{\Theta}^T$, $\lambda_1 < \cdots < \lambda_{2Nm-m} < 1$ are eigenvalues of $\mathbf{G}$ and $\boldsymbol{\Theta}$ is the matrix of eigenvectors corresponding $\{\lambda_i\}_{i=1}^{2Nm}$, and $\boldsymbol{\mathcal{L}} = [-\mathbf{I}_{Nm}, \mathbf{I}_{Nm}],$.*

Proof: The proof is given in Appendix B. ■

In Algorithm 2, we see that agents communicate with their neighbors to choose the weights $w_{ij}(l)$ so that $w_{ij}(l) = w_{ji}(l), \forall i, j \in \mathcal{N}, \forall l$ and hence the adversary can acquire $w_{ij}(l)$. However, if the external eavesdropper does not know the interaction weights $w_{ij}(0)$, $\forall i, j \in \mathcal{N}$, then the state of the network agents remains private with no information leakage and we can guarantee a stronger privacy. We can see that in Algorithm 2, the nodes perturb the substates transmitted to their neighbors in addition to independently selecting coupling weights for different elements of the substates $\boldsymbol{\alpha}_{i,j}(l)$ and $\boldsymbol{\beta}_{i,j}(l)$. From [47, Theorem 3], we can show that any variation in the initial state of the jth agent remains hidden from the external eavesdropper, and hence, no privacy leakage.

B. Honest-but-curious agent

Without loss of generality, let us assume that agent N is the HBC agent as defined in Section II. Agent N uses its own local information $\{\boldsymbol{\alpha}_{N,n}(l), \boldsymbol{\beta}_{N,n}(l), \boldsymbol{\omega}_N(l), \mathbf{u}_N(l)\}_{l=0}^k$ and the information received from its neighbors $\mathcal{N}_N$ to estimate the sensitive information of other agents. From Algorithm 2, we can see that the information available at the HBC agent N at the kth consensus iteration is given by

$$\begin{aligned}\mathcal{I}_N(k) = \{&\boldsymbol{\alpha}_{N,n}(l), \boldsymbol{\beta}_{N,n}(l), \boldsymbol{\omega}_N(l), \mathbf{u}_N(l), \\ &w_{Nj}(l), \tilde{\boldsymbol{\alpha}}_{j,n}(l) : \forall j \in \mathcal{N}_N, l = 0, 1, \ldots, k\}.\end{aligned} \tag{25}$$

The proposed filtering algorithm offers privacy even against HBC agent.

Theorem 2. *If an HBC agent has access only to messages shared by its neighbors and every agent has at least one regular agent in its neighborhood, then an HBC agent cannot infer private information of any other agent in the network.*

Proof: We show that an arbitrary change in the information of agent j, change from $\boldsymbol{r}_{j,n}$ to $\bar{\boldsymbol{r}}_{j,n}$, remains indistinguishable from the HBC agent if agent j has at least one neighboring regular agent l. Compared to Theorem 2 in [47], the shared substates are multivariate and perturbed by noise. However, due to the diminishing perturbation noise and independent coupling weights of the different elements the procedure in the proof of Theorem 2 in [47] is applicable. Consequently, the change from $\boldsymbol{r}_{j,n}$ to $\bar{\boldsymbol{r}}_{j,n}$ remains indistinguishable for the HBC agent, which completes the proof. ■

In Theorem 2, we assumed that the HBC agent has access only to information related to its neighboring agents. We can observe that agent privacy depends on the availability of the interaction and coupling weights at the adversary. Therefore, next, we consider the scenario where the HBC agent has access to the entire weight matrix $\mathbf{W}$ and an estimate of the coupling weight matrix $\hat{\mathbf{U}}$ in addition to information in (25). This information set at the adversary can be represented as

$$\tilde{\mathcal{I}}_N(k) = \mathcal{I}_N(k) \cup \{\mathbf{W}(l), \hat{\mathbf{U}}(l), l = 0, 1, \ldots, k\} \tag{26}$$

where $\hat{\mathbf{U}}$ denotes the estimate of the coupling weight matrix $\mathbf{U}$ at the adversary.

Under these assumptions, the HBC agent estimates the initial substate of the network agents, i.e., $\mathbf{z}_n(0) \triangleq [\boldsymbol{\alpha}_n^{\mathrm{T}}(0), \boldsymbol{\beta}_n^{\mathrm{T}}(0)]^{\mathrm{T}}$. To this end, we require defining an observation vector that includes the shared information of the neighbors and the information of the HBC agent itself at each time instant k, denoted as $\{\tilde{\boldsymbol{\alpha}}_{j,n}(t), \forall j \in \mathcal{N}_N, \alpha_{N,n}(t), \beta_{N,n}(t)\}$, that can be expressed as

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{z}_n(k) + \mathbf{C}_\alpha\boldsymbol{\omega}(k), \tag{27}$$

at each consensus iteration k with $\mathbf{z}_n(k) = [\boldsymbol{\alpha}_n^{\mathrm{T}}(k), \boldsymbol{\beta}_n^{\mathrm{T}}(k)]^{\mathrm{T}}$. In order to capture the relevant set of information, we define $\mathbf{C} = [\mathbf{C}_\alpha, \mathbf{C}_\beta]$ with $\mathbf{C}_\beta = [\mathbf{0}, \mathbf{e}_N]^{\mathrm{T}} \otimes \mathbf{I}_m \in \mathbb{R}^{(N_N+1)m\times Nm}$ that captures the private substates of the HBC agent itself and

$$\mathbf{C}_\alpha = \left[\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, \cdots, \mathbf{e}_{j_{N_N}}, \mathbf{e}_N\right]^{\mathrm{T}} \otimes \mathbf{I}_m \in \mathbb{R}^{(N_N+1)m\times Nm},$$

that captures the public substate of neighbors and the HBC agent itself. The vector $\mathbf{e}_j \in \mathbb{R}^N$ is a vector with 1 in the jth entry and zeros elsewhere, $\mathcal{N}_N = \{j_1, j_2, \cdots, j_{N_N}\}$ is the adjacency set of the HBC agent and N_N denotes the number of its neighbors. As a result, the HBC agent infers the information of all agents as $\boldsymbol{r}_n = \frac{1}{2}(\boldsymbol{\alpha}_n(0) + \boldsymbol{\beta}_n(0))$. Substituting the network-wide substate update equations in (10), i.e.,

$$\begin{aligned}\boldsymbol{\alpha}_n(k+1) =& \mathbf{M}\boldsymbol{\alpha}_n(k) + \varepsilon\mathbf{U}\boldsymbol{\beta}_n(k) + \varepsilon(\mathbf{W}\otimes\mathbf{I}_m)\boldsymbol{\omega}(k)\\ \boldsymbol{\beta}_n(k+1) =& \varepsilon\mathbf{U}\boldsymbol{\alpha}_n(k) + (\mathbf{I}_{Nm} - \varepsilon\mathbf{U})\,\boldsymbol{\beta}_n(k)\end{aligned}$$

into (27) gives

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{G}^k\mathbf{z}_n(0) + \mathbf{C}_\alpha\left(\sum_{t=0}^{k-1}\boldsymbol{\mathcal{C}}_{k-1-t}\mathbf{B}\boldsymbol{\omega}(t) + \boldsymbol{\omega}(k)\right) \tag{28}$$

where $\boldsymbol{\mathcal{C}}_k = \begin{bmatrix}\mathbf{I}_{Nm} & \mathbf{0}_{Nm}\end{bmatrix}\mathbf{G}^k\begin{bmatrix}\mathbf{I}_{Nm} & \mathbf{0}_{Nm}\end{bmatrix}^{\mathrm{T}}$ and $\mathbf{B} = \varepsilon(\mathbf{W}\otimes\mathbf{I}_m)$. Further, $\mathbf{G}$ can be written as $\mathbf{G} = \boldsymbol{\Theta}\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Theta}^{\mathrm{T}}$, where $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_{2Nm}] \in \mathbb{R}^{2Nm\times 2Nm}$ and $\tilde{\boldsymbol{\Lambda}} = \mathsf{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{2Nm})$ consists of eigenvalues of matrix $\mathbf{G}$, with $\lambda_1 < \lambda_2 < \cdots < \lambda_{2Nm-m+1} = \cdots = \lambda_{2Nm} = 1$. Subsequently, we have $\mathbf{G}^l = \boldsymbol{\Theta}\bar{\boldsymbol{\Lambda}}^l\boldsymbol{\Theta}^{\mathrm{T}} + \frac{1}{2N}(\mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}}\otimes\mathbf{I}_m)$ and

$$\boldsymbol{\mathcal{C}}_k \;=\; \boldsymbol{\Theta}_{1:Nm}\bar{\boldsymbol{\Lambda}}^k\boldsymbol{\Theta}_{1:Nm}^{\mathrm{T}} + \frac{1}{2N}(\mathbf{1}_N\mathbf{1}_N^{\mathrm{T}}\otimes\mathbf{I}_m) \tag{29}$$

where $\bar{\boldsymbol{\Lambda}} = \mathsf{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{(2Nm-m)}, 0, \cdots, 0)$ and $\boldsymbol{\Theta}_{1:Nm}$ denotes a matrix that contains the first Nm rows of matrix $\boldsymbol{\Theta}$.

Since $\boldsymbol{\nu}(k)$ is a zero-mean i.i.d. sequence, the accumulated observation of the HBC agent set-up at consensus iteration k is simplified as

$$\begin{aligned}\sum_{t=0}^{k}\mathbf{y}_n(t) =& \mathbf{C}(\mathbf{I}_{2Nm}-\mathbf{G})^{k+1}(\mathbf{I}_{2Nm}-\mathbf{G})^{-1}\mathbf{z}_n(0)\\ &+\mathbf{C}_\alpha\left(\sum_{t=0}^{k-1}\phi^t\boldsymbol{\mathcal{C}}_{k-1-t}\mathbf{B}\boldsymbol{\nu}(t)+\phi^k\boldsymbol{\nu}(k)\right).\end{aligned}$$

Stacking all the available accumulated observations at each consensus iteration k in a vector gives

$$\begin{bmatrix}\sum_{t=0}^{0}\mathbf{y}_n(t)/\phi^0\\ \sum_{t=0}^{1}\mathbf{y}_n(t)/\phi^1\\ \vdots\\ \sum_{t=0}^{k}\mathbf{y}_n(t)/\phi^k\end{bmatrix}=\mathbf{H}(k)\mathbf{z}_n(0)+\mathbf{F}(k)\begin{bmatrix}\boldsymbol{\nu}(0)\\ \boldsymbol{\nu}(1)\\ \vdots\\ \boldsymbol{\nu}(k)\end{bmatrix} \tag{30}$$

where

$$\mathbf{H}(k)=\begin{bmatrix}\mathbf{C}\\ \phi^{-1}\mathbf{C}(\mathbf{I}_{2Nm}+\mathbf{G})\\ \vdots\\ \phi^{-k}\mathbf{C}(\mathbf{I}_{2Nm}+\sum_{t=1}^{k}\mathbf{G}^t)\end{bmatrix}, \tag{31}$$

and

$$\mathbf{F}(k)=\begin{bmatrix}\hat{\mathbf{F}}_0 & \mathbf{0} & \cdots & \mathbf{0}\\ \hat{\mathbf{F}}_1 & \hat{\mathbf{F}}_0 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \hat{\mathbf{F}}_{k-1} & \hat{\mathbf{F}}_{k-2} & \cdots & \hat{\mathbf{F}}_0\end{bmatrix} \tag{32}$$

with $\hat{\mathbf{F}}_0=\mathbf{C}_\alpha$ and $\hat{\mathbf{F}}_k=\frac{\varepsilon}{\phi^{k+1}}\mathbf{C}_\alpha\boldsymbol{\mathcal{C}}_k(\mathbf{W}\otimes\mathbf{I}_m)$ for $k\geq 1$ which, by substituting $\boldsymbol{\mathcal{C}}_k$ in (29), simplifies as

$$\begin{aligned}\hat{\mathbf{F}}_k =& \frac{\varepsilon}{\phi^{k+1}}\mathbf{C}_\alpha\left(\boldsymbol{\Theta}_{1:Nm}\bar{\boldsymbol{\Lambda}}^k\boldsymbol{\Theta}_{1:Nm}^{\mathrm{T}}\right)(\mathbf{W}\otimes\mathbf{I}_m)\\ &+\frac{\varepsilon}{2N\phi^{k+1}}\left(\mathbf{1}_{(N_N+1)}[d_1,d_2,\ldots,d_N]\right)\otimes\mathbf{I}_m\end{aligned} \tag{33}$$

where $d_i=\sum_{j\in\mathcal{N}_i}w_{ij}$. Assuming the estimate of the coupling weight matrix $\mathbf{U}$ at the adversary as $\hat{\mathbf{U}}=\mathbf{U}+\boldsymbol{\Delta}_{\mathbf{U}}$, where $\boldsymbol{\Delta}_{\mathbf{U}}$ denotes the uncertainty in adversary's estimate, we quantify the privacy guarantee in the following results.

Theorem 3. *If an HBC agent has access to the information $\{\mathbf{W}(l)\}_{l=0}^{k}$, the messages shared by its neighbors, and an estimate of the coupling weight matrix $\hat{\mathbf{U}}$, then the error covariance at the HBC agent corresponding to estimate the initial substates $[\boldsymbol{\alpha}_n^T(0),\boldsymbol{\beta}_n^T(0)]^T$ is given by*

$$\tilde{\mathbf{P}}_n(k)=\bar{\mathbf{P}}_n(k)+\mathbb{E}_{\mathbf{U}}\{\varepsilon^2\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\tilde{\boldsymbol{\Pi}}_n\boldsymbol{\Delta}_{\mathbf{H}}^T(k)(\mathbf{H}^\dagger(k))^T\} \tag{34}$$

where $\tilde{\boldsymbol{\Pi}}_n=\mathbf{1}_{2N}\mathbf{1}_{2N}^T\otimes\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^T\}$ with $\mathbf{x}_n$ as the state vector,

$$\begin{aligned}\bar{\mathbf{P}}_n(k)=&\mathbb{E}_{\mathbf{U}}\{\varepsilon^2\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\tilde{\boldsymbol{\Sigma}}_n\boldsymbol{\Delta}_{\mathbf{H}}^T(k)(\mathbf{H}^\dagger(k))^T\\ &+\sigma^2(\mathbf{I}-\varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))\mathbf{H}^\dagger(k)\mathbf{F}(k)\mathbf{F}^T(k)(\mathbf{H}^\dagger(k))^T\\ &(\mathbf{I}-\varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))^T\}\end{aligned} \tag{35}$$

where $\tilde{\boldsymbol{\Sigma}}_n$ is the covariance matrix for (20), *$\mathbf{H}(k)$ and $\mathbf{F}(k)$ are defined in* (31) *and* (32), *respectively, and*

$$\boldsymbol{\Delta}_{\mathbf{H}}(k)=\begin{bmatrix}\mathbf{0}\\ \phi^{-1}\mathbf{C}\boldsymbol{\Delta}_{\mathbf{G}_1}\\ \vdots\\ \phi^{-k}\mathbf{C}\sum_{t=1}^{k}\boldsymbol{\Delta}_{\mathbf{G}_t}\end{bmatrix}$$

with $\boldsymbol{\Delta}_{\mathbf{G}_k}=\sum_{t=1}^{k}\frac{k!\varepsilon^{t-1}}{(k-t)!t!}\mathbf{G}^{k-t}\boldsymbol{\Delta}_{\mathbf{G}_1}^t$, $\boldsymbol{\Delta}_{\mathbf{G}_1}=-\boldsymbol{\mathcal{L}}^T\boldsymbol{\Delta}_{\mathbf{U}}\boldsymbol{\mathcal{L}}$, and $\boldsymbol{\mathcal{L}}=[-\mathbf{I}_{Nm},\mathbf{I}_{Nm}]$.

Proof: The proof is given in Appendix C. ■

From Theorem 3, we can show that the first term in (34) converges to the fixed matrix $\bar{\mathbf{P}}_{\mathrm{LB}}(k)=\lim_{n\to\infty}\bar{\mathbf{P}}_{\mathrm{n}}(k)$ as $\lim_{n\to\infty}\tilde{\boldsymbol{\Sigma}}_n=\tilde{\boldsymbol{\Sigma}}$ and the second term diverges as $\lim_{n\to\infty}\mathrm{tr}\left(\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\}\right)=\infty$. Therefore, a lower bound of the privacy leakage at agent j after k consensus iterations is given by

$$\bar{\mathcal{E}}_j(k)=\mathrm{tr}\left((\mathbf{e}_j^{\mathrm{T}}\otimes\mathbf{I}_m)\mathbf{P}(k)(\mathbf{e}_j\otimes\mathbf{I}_m)\right) \tag{36}$$

where $\mathbf{e}_j\in\mathbb{R}^N$ is a vector with 1 in the jth entry and zeros elsewhere and

$$\mathbf{P}(k)=\frac{1}{4}\begin{bmatrix}\mathbf{I}_{mN} & \mathbf{I}_{mN}\end{bmatrix}\bar{\mathbf{P}}_{\mathrm{LB}}(k)\begin{bmatrix}\mathbf{I}_{mN} & \mathbf{I}_{mN}\end{bmatrix}^{\mathrm{T}}. \tag{37}$$

For the worst-case scenario, when the HBC agent knows the exact coupling weights of the entire network, we can establish the privacy leakage as follows.

Theorem 4. *If an HBC agent knows the exact coupling weights $\mathbf{U}$, i.e., $\boldsymbol{\Delta}_{\mathbf{U}}=\mathbf{0}$, then the error covariance $\tilde{\mathbf{P}}_n(k)$ in* (34) *is*

$$\tilde{\mathbf{P}}(k)=\sigma^2\left(\mathbf{H}^T(k)\left(\mathbf{F}(k)\mathbf{F}^T(k)\right)^{-1}\mathbf{H}(k)\right)^{-1},\quad\forall n. \tag{38}$$

Proof: The proof is given in Appendix D. ■

Remark 2. The privacy guarantee of agents under the special case of $\boldsymbol{\alpha}_{i,n}(0)=\boldsymbol{\beta}_{i,n}(0)=\boldsymbol{r}_{i,n}$ can only be provided by the noise injection technique, and the decomposition technique does not provide privacy. Fortunately, this special case is not of great interest, and the algorithm can be configured to avoid this specific scenario of initial decomposition.

VI. Simulation Results

To illustrate the performance of the proposed PP-DKF algorithm, we consider the undirected connected network with $N=25$ agents shown in Fig. 3. The proposed PP-DKF is used to collaboratively track the speed and position of a target moving in two dimensions where the state vector $\mathbf{x}_n=[X_n,Y_n,\dot{X}_n,\dot{Y}_n]^{\mathrm{T}}$ consists of the positions $\{X_n,Y_n\}$ and velocities $\{\dot{X}_n,\dot{Y}_n\}$ in the horizontal and vertical directions, respectively. The state evolution of such a dynamic system is given by

$$\mathbf{x}_n=\begin{bmatrix}1 & 0 & \Delta T & 0\\ 0 & 1 & 0 & \Delta T\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\mathbf{x}_{n-1}+\begin{bmatrix}\frac{1}{2}(\Delta T)^2 & 0\\ 0 & \frac{1}{2}(\Delta T)^2\\ \Delta T & 0\\ 0 & \Delta T\end{bmatrix}\hat{\mathbf{v}}_n$$

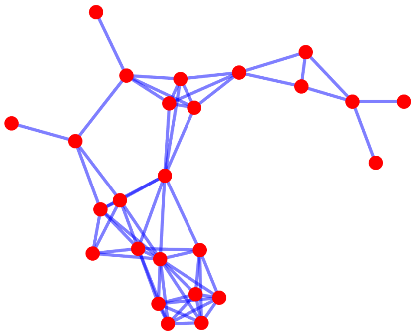


Fig. 3. Network topology with $N = 25$ agents.

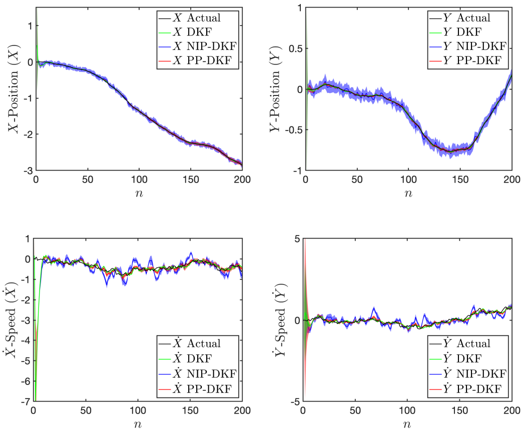


Fig. 4. Tracking performance of distributed Kalman filtering settings for each $N = 25$ agents (shaded color) and their average as a solid line with $K = 30$ consensus iterations and noise variance $\sigma^2 = 4$.

where $\hat{\mathbf{v}}_n = [\ddot{X}_n, \ddot{Y}_n]^{\mathrm{T}}$ denotes the unknown acceleration in horizontal and vertical directions and $\Delta T = 0.04$ is the sampling interval. The acceleration is modeled as zero-mean Gaussian process with covariance matrix of $\mathbb{E}\{\hat{\mathbf{v}}_n \hat{\mathbf{v}}_n^{\mathrm{T}}\} = 1.44\,\mathbf{I}_2$ while the observation parameters as considered as

$$\mathbf{H}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{C}_{\mathbf{w}_{i,n}} = \begin{bmatrix} 0.0416 & 0.008 \\ 0.008 & 0.04 \end{bmatrix}$$

for each agents $i \in \mathcal{N}$. For comparison purposes, we introduce a DKF that employs the conventional noise-injection based average consensus technique proposed in [45], with the injected noise following (9). This algorithm is hereafter referred to as the noise-injection based privacy-preserving DKF (NIP-DKF). The consensus and noise parameters are selected as $\varepsilon = 1/4$ and $\phi = 0.9$, respectively. We considered the interaction weights given in [47], which is $\mathbf{W} = 0.75\mathbf{E}$ where $\mathbf{E}$ denotes the adjacency matrix of the network shown in Fig. 3. The elements of the coupling weight $\mathbf{u}_i$ are chosen independently with distribution $\mathcal{U}(\eta, 1)$ where $\eta = 0.4$.

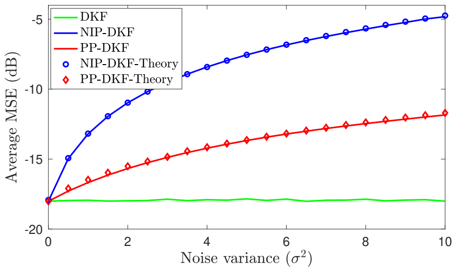


Fig. 5. Average MSE of the filtering process versus noise variance σ^2 for both theory and simulation with $K = 30$ consensus iterations.

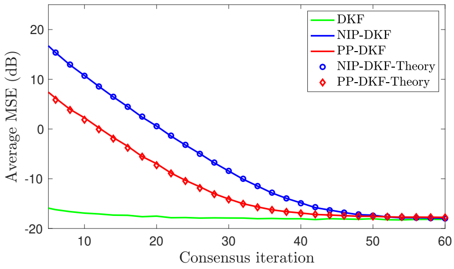


Fig. 6. The overall filtering average MSE versus the number of consensus iteration with noise variance $\sigma^2 = 4$.

A. Kalman filtering performance

Fig. 4 shows the tracking capabilities of the conventional DKF [3], the NIP-DKF, and the proposed PP-DKF, respectively. We see that the PP-DKF performs as well as the conventional DKF, which demonstrates the robustness of the PP-DKF to noise injection and state decomposition. Fig. 5 shows the average MSE of the Kalman filtering process versus the perturbation noise variance σ^2. We see that the perturbation noise degrades the performance of both approaches, PP-DKF and NIP-DKF, compared to the conventional DKF [3]. In other words, increasing the variance of the perturbation noise increases the MSE. The slower growth rate of the PP-DKF compared to the NIP-DKF implies its improved robustness to the injected noise. To compute the filtering state vector estimation error for the NIP-DKF, we follow a similar approach to the PP-DKF (cf. (17)); the detailed derivation is provided in Appendix E. Fig. 5 also shows that the theoretical predictions for NIP-DKF (75) and PP-DKF (20) match the simulation results perfectly.

Fig. 6 shows the average MSE of the PP-DKF and the NIP-DKF versus the number of consensus iteration. We see that increasing the number of consensus iterations reduces the resulting average MSE. For a sufficiently large number of iterations, the filtering performance of the PP-DKF and the NIP-DKF converges to the conventional DKF [3]. Also, it can be seen that the theoretical predictions for a finite number of consensus iterations match the simulation results.

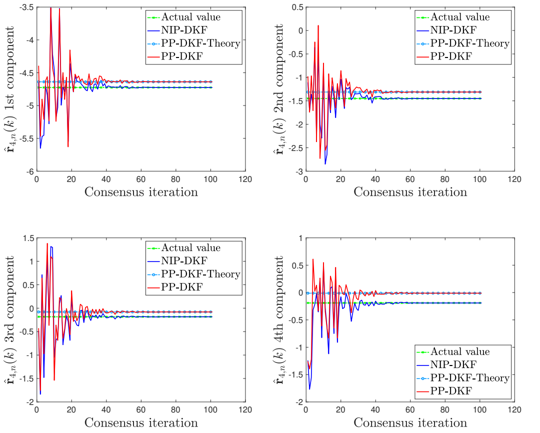


Fig. 7. The observer of the external eavesdropper to estimate all components of the initial state $\boldsymbol{r}_{4,n}(0)$, i.e., $\hat{\boldsymbol{r}}_{4,n}(k)$, given the noise variance $\sigma^2 = 4$.

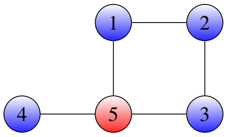


Fig. 8. Network topology with $N = 5$ agents.

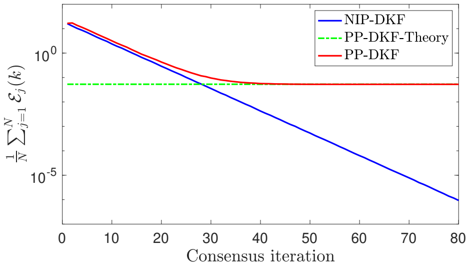


Fig. 9. Average privacy $\frac{1}{N}\sum_{j=1}^{N}\mathcal{E}_j(k)$ versus the number of consensus iterations in the presence of the external eavesdropper.

B. External eavesdropper: privacy analysis

To investigate the privacy performance of the proposed PP-DKF algorithm, we need to focus more on the network and the effect of adversaries on each individual agents. We therefore consider a smaller undirected connected network with $N = 5$ agents shown in Fig. 8. When the NIP-DKF is employed, the external eavesdropper can construct the following observer (cf. (46))

$$\hat{\boldsymbol{r}}_n(k+1) = \hat{\boldsymbol{r}}_n(k) + \tilde{\boldsymbol{r}}_n(k+1) - (\mathbf{Q} \otimes \mathbf{I}_m)\,\tilde{\boldsymbol{r}}_n(k) \tag{39}$$

where $\hat{\boldsymbol{r}}_n(k)$ is the estimate $\boldsymbol{r}_n$ at the eavesdropper at time n after k consensus iterations, $\mathbf{Q} \in \mathbb{R}^{N\times N}$ is a doubly stochastic consensus weight matrix, and $\tilde{\boldsymbol{r}}_n(k) = \boldsymbol{r}_n(k) + \boldsymbol{\omega}(k)$. After some algebraic manipulation the observer in (39) is simplified as

$$\hat{\boldsymbol{r}}_n(k+1) = \boldsymbol{r}_n(0) + \phi^{k+1}\boldsymbol{\nu}(k+1) \tag{40}$$

Since $\phi < 1$, the observer converges to the exact values of the initial states, i.e., $\lim_{k\to\infty}\hat{\boldsymbol{r}}_n(k) = \boldsymbol{r}_n(0)$. Fig. 7 shows the state estimate of the eavesdropper versus the number of consensus iterations. As mentioned above, whenever the NIP-DKF is employed, the eavesdropper can estimate the initial state with great accuracy. In contrast, the PP-DKF prevents the initial state of the agents from being correctly estimated, as predicted by Theorem 1. Fig. 7 shows that the estimate at the eavesdropper in (60) is biased and does not converge to the exact initial state of the agents. It also represents that the predicted estimation bias at the eavesdropper under the PP-DKF matches the simulation perfectly.

Fig 9 shows the average MSE at the external eavesdropper, i.e., $\frac{1}{N}\sum_{j=1}^{N}\mathcal{E}_j(k)$ with $\mathcal{E}_j(k)$ in (22), versus the number of consensus iterations. In general, the larger this MSE becomes, the better the privacy of agent j. Under the NIP-DKF, the average MSE of the external eavesdropper decreases monotonically with the number of consensus iterations. In other words, the MSE at the eavesdropper tends to zero, meaning that the external eavesdropper can determine the initial *a posteriori* state of the agents exactly. In contrast, when considering the proposed PP-DKF, the achievable MSE at the adversary is bounded as in (24) and, therefore, cannot be improved by extending the number of consensus iterations. Fig 9 also shows that the predicted bound of the privacy leakage in Theorem 1 matches the simulation.

C. HBC agent: privacy analysis

Here, we investigate the case when an HBC agent attempts to estimate the initial state of the network agents. We consider the 5th agent to be an HBC agent (see Fig. 8). The HBC agent has no access to the coupling weights of other agents, while as a legitimate agent of the network knows the parameter η. Based on the assumption about the coupling weights distribution, the HBC agent uses an average value $\bar{\mathbf{U}}$, with uncertainty $\boldsymbol{\Delta}_{\mathbf{U}} = \mathbf{U} - \bar{\mathbf{U}}$, to estimate the initial states of the other agents.

Fig 10 shows the lower bound of the agent privacy in (36) after $K = 30$ consensus iterations versus the injected noise variance σ^2. We see that employing the NIP-DKF, the privacy of agent 4 is breached due to the lack of neighbors other than the HBC agent. Consequently, the HBC agent can estimate the initial state of the 4th agent with negligible error. In contrast, the proposed PP-DKF significantly improves the privacy for all agents (agents obtain a substantial level of privacy even with a low amount of injected noise).

The trade-off between Kalman filtering accuracy and the average privacy $\sum_{j=1}^{4}\bar{\mathcal{E}}_j(k)/4$, after $K = 30$ consensus iterations, is shown in Fig. 11. It illustrates the privacy-MSE trade-off for different values of the injected noise variance σ^2. For both PP-DKF and NIP-DKF, we see that a larger privacy guarantee brings a reduction in filtering accuracy, which is reflected in a higher MSE. We see that the Kalman

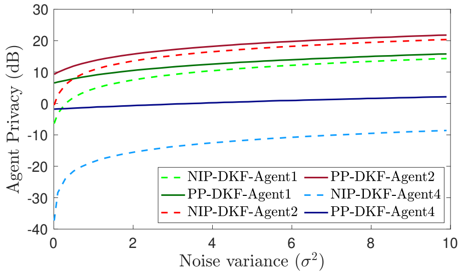


Fig. 10. Agent privacy versus noise variance (σ^2), given $K = 30$ consensus iterations. Due to the symmetric topology, agents 1 and 3 achieve same privacy level and only the result of the 1st agent is shown in the figure.

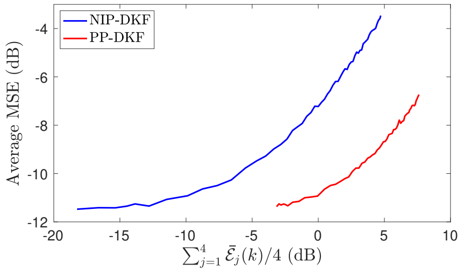


Fig. 11. The trade-off between Kalman filtering accuracy and average privacy $\sum_{j=1}^{4} \bar{\mathcal{E}}_j(k)/4$ for different values of the injected noise variance σ^2.

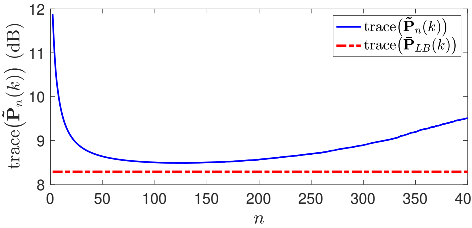


Fig. 12. The mean squared estimation error at the HBC agent after $K = 30$ consensus iterations versus filtering time instant n.

filter accuracy and the average privacy can be controlled with injected noise variance. A fixed privacy guarantee is ensured with the PP-DKF, which has a lower filtering MSE than the NIP-DKF. This is because the NIP-DKF perturbs the entire intermediate state vector estimate before sharing it, whereas the PP-DKF perturbs only its public substate and keeps the private substate noise-free.

Fig. 12 shows the average of the diagonal elements of $\tilde{\mathbf{P}}_n(k)$ in (34) after $K = 30$ consensus iterations versus filtering time instant n. It illustrates the impact of the diverging term $\mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}} \otimes \mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\}$ in $\tilde{\mathbf{P}}_n(k)$, as stated in Theorem 3, and also demonstrates the accuracy of the proposed lower bound of the error covariance matrix, i.e., $\bar{\mathbf{P}}_{LB}(k)$, at the HBC.

VII. Conclusions

This paper introduced a privacy-preserving distributed Kalman filter (PP-DKF) using state-decomposition and noise injection to protect sensitive data of the network agents. The convergence of the PP-DKF was analyzed in the mean and mean-square senses, and we provided closed-form expressions that capture the privacy-related state-decomposition and noise perturbation effects. Further, the agent-privacy provided by the PP-DKF was studied in two adversarial settings, namely, when the network is subjected to external eavesdroppers and honest-but-curious agents. In particular, we established conditions for zero privacy leakage and provided lower bounds on achieved privacy for various practical scenarios. Furthermore, it was shown that the proposed PP-DKF enhances the privacy level of all agents and reduces the sensitivity of the Kalman filtering operations to the injected noise. In addition, the PP-DKF achieved lower MSE than distributed Kalman filters employing other recently proposed privacy-preserving techniques. Lastly, several simulations were presented to corroborate the theoretical results.

Appendix A
Convergence of the decomposition method

To prove that the noise-free version of the update equations (10) converge to the exact average of the initial information, let us assume

$$\begin{aligned}\boldsymbol{\alpha}_n(k) =& [\boldsymbol{\alpha}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\alpha}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{Nm}\\ \boldsymbol{\beta}_n(k) =& [\boldsymbol{\beta}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\beta}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{Nm}.\end{aligned} \tag{41}$$

then network-wide update equations of agents in (10), without perturbation, can be expressed as

$$\begin{aligned}\boldsymbol{\alpha}_n(k+1) =& \mathbf{M}\boldsymbol{\alpha}_n(k) + \varepsilon\mathbf{U}\boldsymbol{\beta}_n(k)\\ \boldsymbol{\beta}_n(k+1) =& \varepsilon\mathbf{U}\boldsymbol{\alpha}_n(k) + (\mathbf{I}_{Nm} - \varepsilon\mathbf{U})\,\boldsymbol{\beta}_n(k)\end{aligned} \tag{42}$$

where $\mathbf{M} = (\mathbf{I}_N - \varepsilon(\mathbf{D} - \mathbf{W})) \otimes \mathbf{I}_m - \varepsilon\mathbf{U}$ with $\mathbf{U} = \mathsf{Blockdiag}(\{\mathbf{U}_i\}_{i=1}^N)$ and $\mathbf{D} = \mathsf{diag}(\{\sum_{j\in\mathcal{N}_i} w_{ij}\}_{i=1}^N)$. Alternatively, (42) can be represented as

$$\underbrace{\begin{bmatrix}\boldsymbol{\alpha}_n(k+1)\\ \boldsymbol{\beta}_n(k+1)\end{bmatrix}}_{\mathbf{z}(k+1)} = \underbrace{\begin{bmatrix}\mathbf{M} & \varepsilon\mathbf{U}\\ \varepsilon\mathbf{U} & \mathbf{I}_{Nm} - \varepsilon\mathbf{U}\end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix}\boldsymbol{\alpha}_n(k)\\ \boldsymbol{\beta}_n(k)\end{bmatrix}}_{\mathbf{z}(k)} \tag{43}$$

where $\mathbf{G} \in \mathbb{R}^{2Nm \times 2Nm}$ is a doubly stochastic matrix. We can derive $\mathbf{z}(k)$'s recursive equation based on its initial value as

$$\mathbf{z}(k+1) = \mathbf{G}^{k+1}\mathbf{z}(0)\cdot \tag{44}$$

Since $\mathbf{G}$ is doubly stochastic, all elements of both $\boldsymbol{\alpha}_n(k+1)$ and $\boldsymbol{\beta}_n(k+1)$ converge to the average of the initial value $\mathbf{z}(0) = [\boldsymbol{\alpha}_n^{\mathrm{T}}(0), \boldsymbol{\beta}_n^{\mathrm{T}}(0)]^{\mathrm{T}}$, i.e., $\sum_{i=1}^N \frac{1}{2N}(\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0))$, asymptotically. Further, since we have the initial condition $\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0) = 2\boldsymbol{r}_{i,n}$, we conclude that

$$\begin{aligned}\lim_{k\to\infty} \boldsymbol{\alpha}_{i,n}(k) = \lim_{k\to\infty} \boldsymbol{\beta}_{i,n}(k) &= \sum_{i=1}^N \frac{1}{2N}(\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0))\\ &= \sum_{i=1}^N \frac{1}{2N}(2\boldsymbol{r}_{i,n}) = \frac{1}{N}\sum_{i=1}^N \boldsymbol{r}_{i,n}\end{aligned}$$

that is the desired average consensus value and completes the proof.

APPENDIX B
PROOF OF THEOREM 1

With the information set $\mathcal{I}_E(k)$ in (23) and the update model in (10), the eavesdropper can construct the following observation model pertaining to agent j

$$\begin{aligned}\hat{\boldsymbol{r}}_{j,n}(k+1) &= \hat{\boldsymbol{r}}_{j,n}(k) + \tilde{\boldsymbol{\alpha}}_{j,n}(k+1) \\ &\quad - \Big(\tilde{\boldsymbol{\alpha}}_{j,n}(k) + \varepsilon \sum_{l\in\mathcal{N}_j} w_{jl}\left(\tilde{\boldsymbol{\alpha}}_{l,n}(k) - \tilde{\boldsymbol{\alpha}}_{j,n}(k)\right)\Big)\end{aligned} \tag{45}$$

with initial value $\hat{\boldsymbol{r}}_{j,n}(0) = \tilde{\boldsymbol{\alpha}}_{j,n}(0)$. After collecting the states and corresponding eavesdropper estimates in the network-wide vectors

$$\begin{aligned}\boldsymbol{r}_n(0) &\triangleq [\boldsymbol{r}_{1,n}^{\mathrm{T}}(0), \cdots, \boldsymbol{r}_{N,n}^{\mathrm{T}}(0)]^{\mathrm{T}} \in \mathbb{R}^{Nm} \\ \hat{\boldsymbol{r}}_n(k) &\triangleq [\hat{\boldsymbol{r}}_{1,n}^{\mathrm{T}}(k), \cdots, \hat{\boldsymbol{r}}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{Nm},\end{aligned}$$

we can, using (45), express the network-wide eavesdropper-estimate as

$$\begin{aligned}\hat{\boldsymbol{r}}_n(k+1) =& \hat{\boldsymbol{r}}_n(k) + \tilde{\boldsymbol{\alpha}}_n(k+1) \\ &- \left((\mathbf{I}_N - \varepsilon(\mathbf{D}-\mathbf{W}))\otimes \mathbf{I}_m\right)\tilde{\boldsymbol{\alpha}}_n(k)\end{aligned} \tag{46}$$

where $\tilde{\boldsymbol{\alpha}}_n(k) = \boldsymbol{\alpha}_n(k) + \boldsymbol{\omega}(k)$ and

$$\begin{aligned}\boldsymbol{\omega}(k) \triangleq& [\boldsymbol{\omega}_1^{\mathrm{T}}(k), \cdots, \boldsymbol{\omega}_N^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{Nm} \\ \boldsymbol{\alpha}_n(k) \triangleq& [\boldsymbol{\alpha}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\alpha}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}} \in \mathbb{R}^{Nm}.\end{aligned}$$

Employing $\tilde{\boldsymbol{\alpha}}_n(k+1) = \boldsymbol{\alpha}_n(k+1) + \boldsymbol{\omega}(k+1)$ and $\tilde{\boldsymbol{\alpha}}_n(k) = \boldsymbol{\alpha}_n(k) + \boldsymbol{\omega}(k)$, the network-wide eavesdropper-estimate in (46) can be further simplified as

$$\begin{aligned}\hat{\boldsymbol{r}}_n(k+1) &= \hat{\boldsymbol{r}}_n(k) + \boldsymbol{\alpha}_n(k+1) + \boldsymbol{\omega}(k+1) \\ &- \left((\mathbf{I}_N - \varepsilon(\mathbf{D}-\mathbf{W}))\otimes \mathbf{I}_m\right)(\boldsymbol{\alpha}_n(k) + \boldsymbol{\omega}(k))\cdot\end{aligned} \tag{47}$$

Considering the network-wide substate update equations in (10), i.e.,

$$\begin{aligned}\boldsymbol{\alpha}_n(k+1) =& \mathbf{M}\boldsymbol{\alpha}_n(k) + \varepsilon\mathbf{U}\boldsymbol{\beta}_n(k) + \varepsilon(\mathbf{W}\otimes\mathbf{I}_m)\boldsymbol{\omega}(k) && (48)\\ \boldsymbol{\beta}_n(k+1) =& \varepsilon\mathbf{U}\boldsymbol{\alpha}_n(k) + (\mathbf{I}_{Nm} - \varepsilon\mathbf{U})\boldsymbol{\beta}_n(k) && (49)\end{aligned}$$

where $\mathbf{M} = (\mathbf{I}_N - \varepsilon(\mathbf{D}-\mathbf{W}))\otimes\mathbf{I}_m - \varepsilon\mathbf{U}$, we obtain from (48) that

$$\begin{aligned}\boldsymbol{\alpha}_n(k+1) &- \left((\mathbf{I}_N - \varepsilon(\mathbf{D}-\mathbf{W}))\otimes \mathbf{I}_m\right)\boldsymbol{\alpha}_n(k) \\ &= \varepsilon\mathbf{U}\left(\boldsymbol{\beta}_n(k) - \boldsymbol{\alpha}_n(k)\right) + \varepsilon(\mathbf{W}\otimes\mathbf{I}_m)\boldsymbol{\omega}(k).\end{aligned} \tag{50}$$

By substituting (50) into (47), we obtain

$$\begin{aligned}\hat{\boldsymbol{r}}_n(k+1) =& \hat{\boldsymbol{r}}_n(k) + \varepsilon\mathbf{U}\left(\boldsymbol{\beta}_n(k) - \boldsymbol{\alpha}_n(k)\right) \\ &- \left((\mathbf{I}_N - \varepsilon\mathbf{D})\otimes\mathbf{I}_m\right)\boldsymbol{\omega}(k) + \boldsymbol{\omega}(k+1)\end{aligned} \tag{51}$$

where $\boldsymbol{\beta}_n(k) = [\boldsymbol{\beta}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\beta}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}}$.

Using (51) and $\hat{\boldsymbol{r}}_n(0) = \boldsymbol{\alpha}_n(0) + \boldsymbol{\omega}(0)$, we can derive the recursive equation of $\hat{\boldsymbol{r}}_n(k)$ as

$$\begin{aligned}\hat{\boldsymbol{r}}_n(k+1) =& \boldsymbol{\alpha}_n(0) + \varepsilon\mathbf{U}\sum_{l=0}^{k}\left(\boldsymbol{\beta}_n(l) - \boldsymbol{\alpha}_n(l)\right) \\ &+ \varepsilon\left(\mathbf{D}\otimes\mathbf{I}_m\right)\sum_{l=0}^{k}\boldsymbol{\omega}(l) + \boldsymbol{\omega}(k+1).\end{aligned} \tag{52}$$

Employing the network-wide update equations in (48) and (49), we obtain

$$\mathbf{z}_n(l) = \begin{bmatrix}\boldsymbol{\alpha}_n(l) \\ \boldsymbol{\beta}_n(l)\end{bmatrix} = \mathbf{G}^l\mathbf{z}_n(0) + \sum_{s=0}^{l-1}\mathbf{G}^{l-1-s}\bar{\boldsymbol{\mathcal{B}}}\boldsymbol{\omega}(s) \tag{53}$$

with $\bar{\boldsymbol{\mathcal{B}}} = \varepsilon[\mathbf{W}, \mathbf{0}_N]^{\mathrm{T}}\otimes\mathbf{I}_m$, and as a result, we can compute $\boldsymbol{\beta}_n(l) - \boldsymbol{\alpha}_n(l)$ as

$$\boldsymbol{\mathcal{L}}\mathbf{z}(l) = \boldsymbol{\beta}_n(l) - \boldsymbol{\alpha}_n(l) = \boldsymbol{\mathcal{L}}\mathbf{G}^l\mathbf{z}_n(0) + \boldsymbol{\mathcal{L}}\sum_{s=0}^{l-1}\mathbf{G}^{l-1-s}\bar{\boldsymbol{\mathcal{B}}}\boldsymbol{\omega}(s) \tag{54}$$

with $\boldsymbol{\mathcal{L}} = [-\mathbf{I}_{Nm}, \mathbf{I}_{Nm}]$. Substituting (54) into (52) results in

$$\hat{\boldsymbol{r}}_n(k+1) = \boldsymbol{\alpha}_n(0) + \varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\left(\sum_{l=0}^{k}\mathbf{G}^l\right)\mathbf{z}_n(0) + \mathbf{n}(k+1) \tag{55}$$

where noise $\mathbf{n}(k+1)$ is given by

$$\begin{aligned}\mathbf{n}(k+1) =& \varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\sum_{l=1}^{k}\sum_{s=0}^{l-1}\mathbf{G}^{l-1-s}\bar{\boldsymbol{\mathcal{B}}}\boldsymbol{\omega}(s) \\ &+ \varepsilon\left(\mathbf{D}\otimes\mathbf{I}_m\right)\sum_{l=0}^{k}\boldsymbol{\omega}(l) + \boldsymbol{\omega}(k+1).\end{aligned} \tag{56}$$

Employing the network-wide definition of the perturbation sequences in (9) results

$$\begin{aligned}\mathbf{n}(k+1) =& \varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\sum_{s=0}^{k-1}\phi^s\mathbf{G}^{k-1-s}\bar{\boldsymbol{\mathcal{B}}}\boldsymbol{\nu}(s) \\ &+ \phi^k\left((\varepsilon\mathbf{D} - \mathbf{I}_N)\otimes\mathbf{I}_m\right)\boldsymbol{\nu}(k) + \phi^{k+1}\boldsymbol{\nu}(k+1).\end{aligned} \tag{57}$$

Since $\mathbf{G}$ is a symmetric and doubly stochastic matrix, by construction, we have

$$\mathbf{G}^k = \begin{bmatrix}\boldsymbol{\mathcal{C}}_k & \boldsymbol{\mathcal{X}}_k \\ \boldsymbol{\mathcal{X}}_k & \boldsymbol{\mathcal{S}}_k\end{bmatrix}.$$

Substituting $\mathbf{G}^k$ in (57), we obtain

$$\begin{aligned}\mathbf{n}(k+1) =& \varepsilon^2\mathbf{U}\sum_{s=0}^{k-1}\phi^s(\boldsymbol{\mathcal{X}}_{k-1-s} - \boldsymbol{\mathcal{C}}_{k-1-s})\left(\mathbf{W}\otimes\mathbf{I}_m\right)\boldsymbol{\nu}(s) \\ &+ \phi^k\left((\varepsilon\mathbf{D} - \mathbf{I}_N)\otimes\mathbf{I}_m\right)\boldsymbol{\nu}(k) + \phi^{k+1}\boldsymbol{\nu}(k+1).\end{aligned}$$

Due to the structure of $\mathbf{G}$ and $\phi \in (0,1)$, $\lim_{k\to\infty}\mathbf{n}(k+1) = \mathbf{0}$. Consequently, the estimate $\hat{\boldsymbol{r}}_n(k)$ converges to $\hat{\boldsymbol{r}}_n = \lim_{k\to\infty}\hat{\boldsymbol{r}}_n(k)$ where

$$\hat{\boldsymbol{r}}_n = \boldsymbol{\alpha}_n(0) + \lim_{k\to\infty}\left(\varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\left(\sum_{l=0}^{k}\mathbf{G}^l\right)\mathbf{z}_n(0)\right). \tag{58}$$

Further, $\mathbf{G}$ can be written as $\mathbf{G} = \boldsymbol{\Theta}\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Theta}^{\mathrm{T}}$, where $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_{2Nm}] \in \mathbb{R}^{2Nm\times 2Nm}$ and $\tilde{\boldsymbol{\Lambda}} = \mathsf{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{2Nm})$ consists of eigenvalues of matrix $\mathbf{G}$, with $\lambda_1 < \lambda_2 < \cdots < \lambda_{2Nm-m+1} = \cdots = \lambda_{2Nm} = 1$. Subsequently, we have

$$\mathbf{G}^l = \boldsymbol{\Theta}\bar{\boldsymbol{\Lambda}}^l\boldsymbol{\Theta}^{\mathrm{T}} + \frac{1}{2N}(\mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}}\otimes\mathbf{I}_m) \tag{59}$$

where $\bar{\boldsymbol{\Lambda}} = \mathsf{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{(2Nm-m)}, 0, \cdots, 0)$. Since the spectral radius of the $\bar{\boldsymbol{\Lambda}}$ is less than one, we have

$\lim_{k\to\infty}\sum_{l=0}^{k}\bar{\mathbf{\Lambda}}^{l} = (\mathbf{I}-\bar{\mathbf{\Lambda}})^{-1}$ and the asymptotic estimate $\hat{\boldsymbol{r}}_n$ in (58) simplifies to

$$\hat{\boldsymbol{r}}_n = \boldsymbol{\alpha}_n(0) + \varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\mathbf{\Lambda}\mathbf{z}_n(0) \tag{60}$$

where $\mathbf{\Lambda} = \mathbf{\Theta}(\mathbf{I}-\bar{\mathbf{\Lambda}})^{-1}\mathbf{\Theta}^{\mathrm{T}} \in \mathbb{R}^{2Nm\times 2Nm}$. The MSE at the eavesdropper corresponding to agent j can be computed as

$$\begin{aligned}\mathcal{E}_j &= \lim_{n\to\infty}\lim_{k\to\infty}\mathcal{E}_j(k)\\ &= \lim_{n\to\infty}\mathsf{tr}\left((\mathbf{e}_j^{\mathrm{T}}\otimes\mathbf{I}_m)\mathbb{E}\{(\boldsymbol{r}_n-\hat{\boldsymbol{r}}_n)(\boldsymbol{r}_n-\hat{\boldsymbol{r}}_n)^{\mathrm{T}}\}(\mathbf{e}_j\otimes\mathbf{I}_m)\right)\end{aligned}$$

Hence, from the state decomposition constraint in (8), the privacy leakage for agent j in (22) can be expressed as

$$\mathcal{E}_j = \lim_{n\to\infty}\mathsf{tr}\left((\mathbf{e}_j^{\mathrm{T}}\otimes\mathbf{I}_m)\,\tilde{\boldsymbol{\mathcal{L}}}\,\mathbb{E}\{\mathbf{z}_n(0)\mathbf{z}_n^{\mathrm{T}}(0)\}\,\tilde{\boldsymbol{\mathcal{L}}}^{\mathrm{T}}\,(\mathbf{e}_j\otimes\mathbf{I}_m)\right) \tag{61}$$

where $\tilde{\boldsymbol{\mathcal{L}}} = \frac{1}{2}\boldsymbol{\mathcal{L}} - \varepsilon\mathbf{U}\boldsymbol{\mathcal{L}}\mathbf{\Lambda}$. Since we are considering the asymptotic analysis, for notational convenience, we remove the index of k from the parameters. In order to remove the time-dependence, $\mathbb{E}\{\mathbf{z}_n(0)\mathbf{z}_n^{\mathrm{T}}(0)\}$ needs to be computed. By stacking all the vectors in (12), we obtain a network-wide intermediate estimation error as $\boldsymbol{\mathcal{E}}_n = \mathbf{1}_{2N}\otimes\mathbf{x}_n - \mathbf{z}_n(0)$. Since $\mathbf{x}_n$ and the intermediate estimation error $\boldsymbol{\mathcal{E}}_n$ are uncorrelated, we have

$$\mathbb{E}\{\mathbf{z}_n(0)\mathbf{z}_n^{\mathrm{T}}(0)\} = \tilde{\mathbf{\Sigma}}_n + \mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}}\otimes\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\} \tag{62}$$

where $\tilde{\mathbf{\Sigma}}_n = \mathbb{E}\{\boldsymbol{\mathcal{E}}_n\boldsymbol{\mathcal{E}}_n^{\mathrm{T}}\}$. From (1) and assuming that $\mathbf{x}_{-1}\sim\mathcal{N}(\mathbf{0},\mathbf{\Pi}_0)$, we can obtain

$$\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\} = \mathbf{A}^{n+1}\mathbf{\Pi}_0(\mathbf{A}^{n+1})^{\mathrm{T}} + \sum_{i=0}^{n}\mathbf{A}^{n-i}\mathbf{C}_{\mathbf{v}_i}(\mathbf{A}^{n-i})^{\mathrm{T}} \tag{63}$$

which is diverging. Since $\lim_{n\to\infty}\mathbf{\Sigma}_n = \mathbf{\Sigma}$, it follows that $\lim_{n\to\infty}\tilde{\mathbf{\Sigma}}_n = \tilde{\mathbf{\Sigma}}$. Thus, $\lim_{n\to\infty}\mathbb{E}\{\mathbf{z}_n(0)\mathbf{z}_n^{\mathrm{T}}(0)\}$ consists of a fixed term $\tilde{\mathbf{\Sigma}}$ and a diverging term as

$$\lim_{n\to\infty}\mathbb{E}\{\mathbf{z}_n(0)\mathbf{z}_n^{\mathrm{T}}(0)\} = \tilde{\mathbf{\Sigma}} + \mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}}\otimes\lim_{n\to\infty}\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\}. \tag{64}$$

From (59) and since $\mathbf{G}^l$ is a doubly stochastic matrix, it follows that for all l the sum of elements in each row (column) of the matrix $\mathbf{\Theta}\bar{\mathbf{\Lambda}}^{l}\mathbf{\Theta}^{\mathrm{T}}$ is zero. Subsequently, the sum of elements in every row (column) of the matrix $\mathbf{\Lambda} = \mathbf{\Theta}(\sum_{l=0}^{\infty}\bar{\mathbf{\Lambda}}^{l})\mathbf{\Theta}^{\mathrm{T}}$ is equal to one. Thus, the term of $\tilde{\boldsymbol{\mathcal{L}}}(\mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}}\otimes\lim_{n\to\infty}\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\})\tilde{\boldsymbol{\mathcal{L}}}^{\mathrm{T}}$ in (61) becomes zero due to the structure of $\tilde{\boldsymbol{\mathcal{L}}}$, and, privacy leakage for agent j is obtained as

$$\mathcal{E}_j = \mathsf{tr}\left((\mathbf{e}_j^{\mathrm{T}}\otimes\mathbf{I}_m)\,\tilde{\boldsymbol{\mathcal{L}}}\tilde{\mathbf{\Sigma}}\tilde{\boldsymbol{\mathcal{L}}}^{\mathrm{T}}\,(\mathbf{e}_j\otimes\mathbf{I}_m)\right)$$

which completes the proof.

APPENDIX C
PROOF OF THEOREM 3

To find a closed-form expression for the error covariance $\tilde{\mathbf{P}}_n(k)$ in (34), we estimate the initial substates $\mathbf{z}_n(0)$ using the observation model in (30). If the perfect observation matrix $\mathbf{H}(k)$ is available, the estimate of the initial substates $\mathbf{z}_n(0) = [\boldsymbol{\alpha}_n^{\mathrm{T}}(0), \boldsymbol{\beta}_n^{\mathrm{T}}(0)]^{\mathrm{T}}$ can be modeled as

$$\bar{\mathbf{z}}_n(0) = \mathbf{H}^{\dagger}(k)(\mathbf{H}(k)\mathbf{z}_n(0) + \mathbf{F}(k)\bar{\boldsymbol{\nu}}(k)) \tag{65}$$

where $\bar{\boldsymbol{\nu}}(k) = [\boldsymbol{\nu}^{\mathrm{T}}(0), \boldsymbol{\nu}^{\mathrm{T}}(1), \cdots, \boldsymbol{\nu}^{\mathrm{T}}(k)]^{\mathrm{T}}$. However, the observation matrix $\mathbf{H}(k)$ has to be estimated at the HBC agent due to the uncertainty of the coupling weight matrix $\mathbf{U}$ at the HBC agent.

Following the estimation procedure in [54], the HBC agent estimates the coupling weight matrix as $\hat{\mathbf{U}} = \mathbf{U} + \mathbf{\Delta}_{\mathbf{U}}$ where $\mathbf{\Delta}_{\mathbf{U}}$ shows its uncertainty to determine the coupling weight matrix $\mathbf{U}$. An estimate of matrix $\mathbf{G}$ is obtained using uncertainty modeling above as $\hat{\mathbf{G}} = \mathbf{G} + \varepsilon\mathbf{\Delta}_{\mathbf{G}_1}$ where $\mathbf{\Delta}_{\mathbf{G}_1} = -\boldsymbol{\mathcal{L}}^{\mathrm{T}}\mathbf{\Delta}_{\mathbf{U}}\boldsymbol{\mathcal{L}}$. Employing the binomial expansion, the uncertainty of $\hat{\mathbf{G}}^k$ is simplified as $\hat{\mathbf{G}}^k = \mathbf{G}^k + \varepsilon\mathbf{\Delta}_{\mathbf{G}_k}$ where

$$\mathbf{\Delta}_{\mathbf{G}_k} = \sum_{t=1}^{k}\frac{k!\varepsilon^{t-1}}{(k-t)!t!}\mathbf{G}^{k-t}\mathbf{\Delta}_{\mathbf{G}_1}^{t} \;\; \forall k\geq 2.$$

Thus, estimate of the observation matrix $\mathbf{H}(k)$ is is formulated as $\hat{\mathbf{H}}(k) = \mathbf{H}(k) + \varepsilon\mathbf{\Delta}_{\mathbf{H}}(k)$ where $\mathbf{\Delta}_{\mathbf{H}}(k)$ denotes the uncertainty of the observation matrix, independent of $\mathbf{H}(k)$, and is computed as

$$\mathbf{\Delta}_{\mathbf{H}}(k) = \begin{bmatrix}\mathbf{0}\\ \phi^{-1}\mathbf{C}\mathbf{\Delta}_{\mathbf{G}_1}\\ \vdots\\ \phi^{-k}\mathbf{C}\sum_{t=1}^{k}\mathbf{\Delta}_{\mathbf{G}_t}\end{bmatrix}.$$

Subsequently, the estimate of the initial substates in (65) is reformulated as

$$\hat{\mathbf{z}}_n(0) = \hat{\mathbf{H}}^{\dagger}(k)\mathbf{y}_n(k) \tag{66}$$

where $\hat{\mathbf{H}}^{\dagger}(k) = (\mathbf{H}(k) + \mathbf{\Delta}_{\mathbf{H}}(k))^{\dagger}$. The HBC agent is a legitimate agent of the network and knows the distribution of coupling weights. Given a negligible uncertainty in $\hat{\mathbf{H}}(k)$, the pseudo-inverse in (66) can be approximated by the first order Taylor expansion as

$$\hat{\mathbf{H}}^{\dagger}(k) \cong \mathbf{H}^{\dagger}(k)\left(\mathbf{I}_{(k+1)(N_N+1)m} - \varepsilon\mathbf{\Delta}_{\mathbf{H}}(k)\mathbf{H}^{\dagger}(k)\right). \tag{67}$$

Substituting (67) into (66) results in

$$\hat{\mathbf{z}}_n(0) = \left(\mathbf{H}^{\dagger}(k) - \varepsilon\mathbf{H}^{\dagger}(k)\mathbf{\Delta}_{\mathbf{H}}(k)\mathbf{H}^{\dagger}(k)\right)\mathbf{y}_n(k),$$

which can be further simplifies as

$$\hat{\mathbf{z}}_n(0) = \mathbf{z}_n(0) + \boldsymbol{\eta}(k) \tag{68}$$

where $\boldsymbol{\eta}(k)$ is the estimation error of the initial substates

$$\begin{aligned}\boldsymbol{\eta}(k) =& \mathbf{H}^{\dagger}(k)\mathbf{F}(k)\bar{\boldsymbol{\nu}}(k) - \varepsilon\mathbf{H}^{\dagger}(k)\mathbf{\Delta}_{\mathbf{H}}(k)\mathbf{z}_n(0)\\ &- \varepsilon\mathbf{H}^{\dagger}(k)\mathbf{\Delta}_{\mathbf{H}}(k)\mathbf{H}^{\dagger}(k)\mathbf{F}(k)\bar{\boldsymbol{\nu}}(k).\end{aligned}$$

Thus, the estimation error covariance, given $\mathbb{E}\{\bar{\boldsymbol{\nu}}(k)\bar{\boldsymbol{\nu}}(k)\} = \sigma^2\mathbf{I}_{(k+1)Nm}$, assuming mutual independence of the noise sequences $\mathbf{w}_{i,n}$, $\mathbf{v}_n$, $\boldsymbol{\nu}_i(k)$, and initial system state $\mathbf{x}_{-1} \sim \mathcal{N}(\mathbf{0},\mathbf{\Pi}_0)$ for all $n = 1, 2, \cdots$, $i\in\mathcal{N}$, and $k\in[1,K]$, is obtained as

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{\eta}(k)\boldsymbol{\eta}^{\mathrm{T}}(k)\}=\\ &\quad\varepsilon^2\mathbf{H}^{\dagger}(k)\mathbf{\Delta}_{\mathbf{H}}(k)\mathbb{E}\{\mathbf{z}_n(0)\mathbf{z}_n^{\mathrm{T}}(0)\}\mathbf{\Delta}_{\mathbf{H}}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}}\\ &\quad+\sigma^2(\mathbf{I}-\varepsilon\mathbf{H}^{\dagger}(k)\mathbf{\Delta}_{\mathbf{H}}(k))\mathbf{H}^{\dagger}(k)\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}}\\ &\qquad(\mathbf{I}-\varepsilon\mathbf{H}^{\dagger}(k)\mathbf{\Delta}_{\mathbf{H}}(k))^{\mathrm{T}}.\end{aligned} \tag{69}$$

The average of the estimation error covariance in (69), with respect to the uncertainty of the coupling weights is denoted as $\tilde{\mathbf{P}}_n(k) = \mathbb{E}_{\mathbf{U}}\left\{\mathbb{E}\{\boldsymbol{\eta}(k)\boldsymbol{\eta}^{\mathrm{T}}(k)\}\right\}$ which by substituting (62) into (69), we have

$$\tilde{\mathbf{P}}_n(k) = \bar{\mathbf{P}}_{\mathrm{n}}(k) + \mathbb{E}_{\mathbf{U}}\big\{\varepsilon^2\mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\tilde{\boldsymbol{\Pi}}_n\boldsymbol{\Delta}_{\mathbf{H}}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}}\big\}$$

where $\tilde{\boldsymbol{\Pi}}_n = \mathbf{1}_{2N}\mathbf{1}_{2N}^{\mathrm{T}} \otimes \mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\}$ with $\mathbf{x}_n$ representing the state vector in (1) and

$$\begin{aligned}\bar{\mathbf{P}}_{\mathrm{n}}(k) = \mathbb{E}_{\mathbf{U}}\big\{&\varepsilon^2\mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\tilde{\boldsymbol{\Sigma}}_n\boldsymbol{\Delta}_{\mathbf{H}}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}} \\ &+ \sigma^2(\mathbf{I} - \varepsilon\mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))\mathbf{H}^{\dagger}(k)\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}} \\ &(\mathbf{I} - \varepsilon\mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))^{\mathrm{T}}\big\}.\end{aligned} \tag{70}$$

From (64), it has been shown that $\tilde{\mathbf{P}}_n(k)$ is comprised of a fixed and a diverging terms, which completes the proof.

Appendix D
Proof of Theorem 4

A worst-case scenario for privacy in Appendix C occurs when the HBC agent has access to coupling weights of the entire network, resulting in access to the actual value of the observation matrix $\mathbf{H}(k)$. In this scenario, $\boldsymbol{\Delta}_{\mathbf{H}} = \mathbf{0}$, the estimation error covariance matrix in (34) simplifies to

$$\tilde{\mathbf{P}}(k) = \sigma^2\left(\mathbf{H}^{\mathrm{T}}(k)\left(\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)\right)^{-1}\mathbf{H}(k)\right)^{-1} \tag{71}$$

which is the same as the error covariance matrix of an ML estimator [55] with the observation model in (30). Here, we show that although the HBC agent has access to the coupling weights of the entire network, the mean squared estimation error at the HBC agent attempting to estimate substates $\boldsymbol{\alpha}_{j,n}(0)$ and $\boldsymbol{\beta}_{j,n}(0)$, respectively, defined as

$$\begin{aligned}\tilde{\mathcal{E}}_j(k) &= \mathsf{tr}\left((\tilde{\mathbf{e}}_j \otimes \mathbf{I}_m)\tilde{\mathbf{P}}(k)(\tilde{\mathbf{e}}_j^{\mathrm{T}} \otimes \mathbf{I}_m)\right) \\ \tilde{\mathcal{E}}_{N+j}(k) &= \mathsf{tr}\left((\tilde{\mathbf{e}}_{N+j} \otimes \mathbf{I}_m)\tilde{\mathbf{P}}(k)(\tilde{\mathbf{e}}_{N+j}^{\mathrm{T}} \otimes \mathbf{I}_m)\right),\end{aligned}$$

is non-zero, where $\tilde{\mathbf{e}}_j \in \mathbb{R}^{2N}$ is a vector with 1 in the jth entry and zeros elsewhere. The mean squared estimation error $\tilde{\mathcal{E}}_j(k)$ for $j = 1, 2, \cdots, 2N$ is lower-bounded as

$$\tilde{\mathcal{E}}_j(k) = \mathsf{tr}\left((\tilde{\mathbf{e}}_j \otimes \mathbf{I}_m)(\tilde{\mathbf{e}}_j^{\mathrm{T}} \otimes \mathbf{I}_m)\tilde{\mathbf{P}}(k)\right) > \lambda_{\mathsf{min}}\, m$$

where λ_{min} is the minimum eigenvalue of the error covariance $\tilde{\mathbf{P}}(k)$ and m is length of the state vector. Therefore, all agents will have an estimate error greater than zero if we can show that $\lambda_{\mathsf{min}} > 0$. In other words, it is sufficient to show that (71) is invertible. We start by showing the invertibility of $\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)$ where $\mathbf{F}(k) \triangleq (\mathbf{I}_{k+1} \otimes \mathbf{C}_{\alpha})\boldsymbol{\mathcal{F}}(k)$ and

$$\boldsymbol{\mathcal{F}}(k) = \begin{bmatrix} \mathbf{I}_{Nm} & \mathbf{0}_{Nm} & \cdots & \mathbf{0}_{Nm} \\ \phi^{-1}\boldsymbol{\mathcal{C}}_0\mathbf{B} & \mathbf{I}_{Nm} & \cdots & \mathbf{0}_{Nm} \\ \vdots & \vdots & \ddots & \vdots \\ \phi^{-k}\boldsymbol{\mathcal{C}}_{k-1}\mathbf{B} & \phi^{-(k-1)}\boldsymbol{\mathcal{C}}_{k-2}\mathbf{B} & \cdots & \mathbf{I}_{Nm} \end{bmatrix}. \tag{72}$$

To this end, let us consider an arbitrary vector $\mathbf{x} = \left[\mathbf{x}_0^{\mathrm{T}}, \mathbf{x}_1^{\mathrm{T}}, \cdots \mathbf{x}_k^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{(k+1)Nm}$, and form

$$\boldsymbol{\mathcal{F}}(k)\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \phi^{-1}\boldsymbol{\mathcal{C}}_0\mathbf{B}\mathbf{x}_0 + \mathbf{x}_1 \\ \vdots \\ \phi^{-k}\boldsymbol{\mathcal{C}}_{k-1}\mathbf{B}\mathbf{x}_0 + \cdots + \mathbf{x}_k \end{bmatrix} = \mathbf{0}. \tag{73}$$

It follows that the only vector satisfying (73) is the trivial solution $\mathbf{x} = \mathbf{0}$. Thus, $\boldsymbol{\mathcal{F}}(k)$ is a full rank matrix and invertible. Considering the structure of the observation matrix $\mathbf{H}(k)$ and $\tilde{\mathbf{P}}(k)$ in (71), for $\mathbf{H}^{\mathrm{T}}(k)\left(\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)\right)^{-1}\mathbf{H}(k)$ to be invertible $\mathbf{H}(k)$ must have rank greater than or equal to $2mN$. By collecting sufficient information, the observation matrix $\mathbf{H}(k)$ must have at least $2mN$ independent rows, then the HBC agent can estimate the initial substate of the network agents with a non-zero estimation error.

Appendix E
Filtering Performance under the NIP-DKF

Following a same approach to that of the PP-DKF (cf. (17)), we formulate the network-wide state vector estimation error dynamics, given k consensus iterations, as follows

$$\begin{aligned}\bar{\boldsymbol{\mathcal{E}}}_{n|n} &= \left(\mathbf{Q}^k \otimes \mathbf{I}_m\right)\bar{\boldsymbol{\mathcal{E}}}_n + \phi^{k-1}(\mathbf{Q} \otimes \mathbf{I}_m)\boldsymbol{\nu}(k-1) \\ &+ \sum_{s=2}^{k}\phi^{k-s}\left((\mathbf{Q}^s - \mathbf{Q}^{s-1}) \otimes \mathbf{I}_m\right)\boldsymbol{\nu}(k-s)\end{aligned} \tag{74}$$

where $\mathbf{Q}$ is the doubly stochastic consensus weight matrix as introduced in [45]. For notational convenience, we removed the index k from the parameters in the following analysis. Alternatively, (74) can be reformulated as

$$\begin{aligned}\bar{\boldsymbol{\mathcal{E}}}_{n|n} = \bar{\boldsymbol{\mathcal{P}}}\bar{\boldsymbol{\mathcal{E}}}_{n-1|n-1} + \bar{\boldsymbol{\mathcal{Q}}}\bar{\boldsymbol{\Upsilon}}_n - \bar{\boldsymbol{\Omega}}_n + \phi^{k-1}(\mathbf{Q} \otimes \mathbf{I}_m)\boldsymbol{\nu}(k-1) \\ + \sum_{s=2}^{k}\phi^{k-s}\left((\mathbf{Q}^s - \mathbf{Q}^{s-1}) \otimes \mathbf{I}_m\right)\boldsymbol{\nu}(k-s)\end{aligned}$$

where $\bar{\boldsymbol{\Upsilon}}_n = [\mathbf{v}_n^{\mathrm{T}}, \cdots, \mathbf{v}_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Nm}$ and

$$\begin{aligned}\bar{\boldsymbol{\mathcal{P}}} &= \left(\mathbf{Q}^k \otimes \mathbf{I}_m\right)\mathsf{Blockdiag}(\{\mathbf{P}_i\mathbf{A}\}_{i=1}^N) \\ \bar{\boldsymbol{\mathcal{Q}}} &= \left(\mathbf{Q}^k \otimes \mathbf{I}_m\right)\mathsf{Blockdiag}(\{\mathbf{P}_i\}_{i=1}^N) \\ \bar{\boldsymbol{\Omega}}_n &= \left(\mathbf{Q}^k \otimes \mathbf{I}_m\right)\mathsf{Blockdiag}(\{\mathbf{Q}_i\}_{i=1}^N)[\mathbf{w}_{1,n}^{\mathrm{T}}, \cdots, \mathbf{w}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}.\end{aligned}$$

The second-order statistics of all agents, denoted by $\bar{\boldsymbol{\Sigma}}_n = \mathbb{E}\{\bar{\boldsymbol{\mathcal{E}}}_{n|n}\bar{\boldsymbol{\mathcal{E}}}_{n|n}^{\mathrm{T}}\}$, is given by

$$\bar{\boldsymbol{\Sigma}}_n = \bar{\boldsymbol{\mathcal{P}}}\bar{\boldsymbol{\Sigma}}_{n-1}\bar{\boldsymbol{\mathcal{P}}}^{\mathrm{T}} + \bar{\boldsymbol{\mathcal{Q}}}\bar{\mathbf{C}}_{\boldsymbol{\Upsilon}}\bar{\boldsymbol{\mathcal{Q}}}^{\mathrm{T}} + \bar{\mathbf{C}}_{\boldsymbol{\Omega}} + \bar{\boldsymbol{\mathcal{T}}} \tag{75}$$

where $\bar{\mathbf{C}}_{\boldsymbol{\Upsilon}} = \mathbb{E}\{\bar{\boldsymbol{\Upsilon}}_n\bar{\boldsymbol{\Upsilon}}_n^{\mathrm{T}}\}$, and $\bar{\mathbf{C}}_{\boldsymbol{\Omega}} = \mathbb{E}\{\bar{\boldsymbol{\Omega}}_n\bar{\boldsymbol{\Omega}}_n^{\mathrm{T}}\}$. The effect of injected noise is manifested in $\bar{\boldsymbol{\mathcal{T}}}$ which evolves as

$$\bar{\boldsymbol{\mathcal{T}}} = \sum_{s=2}^{k}\phi^{2(k-s)}\tilde{\boldsymbol{\mathcal{T}}}_s + \phi^{2(k-1)}(\mathbf{Q} \otimes \mathbf{I}_m)\mathbf{C}_{\boldsymbol{\nu}}(\mathbf{Q} \otimes \mathbf{I}_m)^{\mathrm{T}}$$

with $\tilde{\boldsymbol{\mathcal{T}}}_s = \left((\mathbf{Q}^{s-1} - \mathbf{Q}^{s-2}) \otimes \mathbf{I}_m\right)\mathbf{C}_{\boldsymbol{\nu}}\left((\mathbf{Q}^{s-1} - \mathbf{Q}^{s-2}) \otimes \mathbf{I}_m\right)^{\mathrm{T}}$. Due to the doubly stochastic matrix $\mathbf{Q}$ and similar to [3], $\bar{\boldsymbol{\mathcal{P}}}$ is stable; therefore, $\bar{\boldsymbol{\Sigma}}_n \to \bar{\boldsymbol{\Sigma}}$ as $n \to \infty$ and

$$\mathbb{E}\{\bar{\boldsymbol{\mathcal{E}}}_{n|n}\} = \bar{\boldsymbol{\mathcal{P}}}\mathbb{E}\{\bar{\boldsymbol{\mathcal{E}}}_{n-1|n-1}\} = \bar{\boldsymbol{\mathcal{P}}}^n\mathbb{E}\{\bar{\boldsymbol{\mathcal{E}}}_{0|0}\}.$$

Since $\bar{\boldsymbol{\mathcal{P}}}$ is stable, we have $\lim_{n\to\infty}\mathbb{E}\{\bar{\boldsymbol{\mathcal{E}}}_{n|n}\}=0$ that indicates the steady-state estimates are unbiased regardless of their initializing values or privacy-preserving perturbations. The effect of injected noise is manifested in terms of $\bar{\boldsymbol{\mathcal{T}}}$, which degrades the steady-state MSE.

References

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, Jun. 2018.

[2] V. Katewa, F. Pasqualetti, and V. Gupta, "On privacy vs. cooperation in multi-agent systems," *Int. J. of Control*, vol. 91, no. 7, pp. 1693–1707, Jul. 2018.

[3] S. P. Talebi and S. Werner, "Distributed Kalman filtering and control through embedded average consensus information fusion," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4396–4403, Oct. 2019.

[4] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4782–4795, Dec. 2006.

[5] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Trans. Autom. Control*, vol. 33, no. 1, pp. 88–94, Jan. 1988.

[6] S. Das and J. M. Moura, "Distributed Kalman filtering with dynamic observations consensus," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4458–4473, Sept. 2015.

[7] R. Olfati-Saber, "Distributed Kalman filtering and sensor fusion in sensor networks," in *Netw. Embedded Sens. Control*, vol. 331. Heidelberg, Germany: Springer, 2006, pp. 157–167.

[8] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decis. and Control*, 2007, pp. 5492–5498.

[9] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decis. and Control*, 2005, pp. 8179–8184.

[10] U. A. Khan and J. M. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[11] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sept. 2010.

[12] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, Jan. 2007.

[13] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th IEEE Int. Symp. Inf. Process. Sensor Networks*, 2005, pp. 63–70.

[14] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. 48th IEEE Conf. Decis. and Control*, 2009, pp. 7036–7042.

[15] S. Das and J. M. Moura, "Consensus + innovations distributed Kalman filter with optimized gains," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 467–481, Jan. 2016.

[16] J. Qin, J. Wang, L. Shi, and Y. Kang, "Randomized consensus-based distributed Kalman filtering over wireless sensor networks," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3794–3801, Aug. 2021.

[17] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *Proc. 45th IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2020, pp. 5895–5899.

[18] T. Yin, Y. Lv, and W. Yu, "Accurate privacy preserving average consensus," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 67, no. 4, pp. 690–694, Apr. 2020.

[19] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.

[20] J. Wang, R. Zhu, and S. Liu, "A differentially private unscented Kalman filter for streaming data in IoT," *IEEE Access*, vol. 6, pp. 6487–6495, Mar. 2018.

[21] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed privacy-preserving data aggregation against dishonest nodes in network systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1462–1470, Apr. 2019.

[22] Q. Li, R. Heusdens, and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," *IEEE Trans. Signal Process.*, vol. 68, pp. 5983–5996, Oct. 2020.

[23] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. 16th Int. Conf. Distrib. Comput. and Netw.*, 2015, pp. 1–10.

[24] Q. Li, M. Coutino, G. Leus, and M. G. Christensen, "Privacy-preserving distributed graph filtering," in *Proc. 28th IEEE Eur. Signal Process. Conf.*, 2021, pp. 2155–2159.

[25] M. Ruan, M. Ahmad, and Y. Wang, "Secure and privacy-preserving average consensus," in *Proc. Workshop Cyber-phys. Syst. Secur. Privacy*, 2017, pp. 123–129.

[26] J. He, L. Cai, and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 4069–4082, Jul. 2020.

[27] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, Jul. 2017.

[28] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus with optimal noise selection," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 203 – 208, 2015.

[29] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4035–4049, Oct. 2019.

[30] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 127–138, Mar. 2019.

[31] X. Wang, J. He, P. Cheng, and J. Chen, "Privacy preserving average consensus with different privacy guarantee," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 5189–5194.

[32] C. Altafini, "A dynamical approach to privacy preserving average consensus," in *Proc. 58th IEEE Conf. Decis. and Control*, 2019, pp. 4501–4506.

[33] A. Moradi, N. K. Venkategowda, and S. Werner, "Coordinated data-falsification attacks in consensus-based distributed Kalman filtering," in *Proc. 8th IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, 2019, pp. 495–499.

[34] N. K. Venkategowda and S. Werner, "Privacy-preserving distributed maximum consensus," *IEEE Signal Process. Lett.*, vol. 27, pp. 1839–1843, Oct. 2020.

[35] Y. Mo and B. Sinopoli, "Secure estimation in the presence of integrity attacks," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1145–1151, Apr. 2015.

[36] I. Jovanov and M. Pajic, "Relaxing integrity requirements for attack-resilient cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 64, no. 12, pp. 4843–4858, Dec. 2019.

[37] Z. Guo, D. Shi, D. E. Quevedo, and L. Shi, "Secure state estimation against integrity attacks: A gaussian mixture model approach," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 194–207, Jan. 2019.

[38] A.-Y. Lu and G.-H. Yang, "Secure state estimation for multiagent systems with faulty and malicious agents," *IEEE Trans. Autom. Control*, vol. 65, no. 8, pp. 3471–3485, Aug. 2019.

[39] L. Su and S. Shahrampour, "Finite-time guarantees for byzantine-resilient distributed state estimation with noisy measurements," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3758–3771, Sep. 2020.

[40] X. Ren, Y. Mo, J. Chen, and K. H. Johansson, "Secure state estimation with byzantine sensors: A probabilistic approach," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3742–3757, Sep. 2020.

[41] A.-Y. Lu and G.-H. Yang, "Distributed secure state estimation in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2875–2882, Jun. 2021.

[42] X. Liu, Y. Mo, and E. Garone, "Local decomposition of Kalman filters and its application for secure state estimation," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 5037–5044, Oct. 2020.

[43] Y. Ni, J. Wu, L. Li, and L. Shi, "Multi-party dynamic state estimation that preserves data and model privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2288–2299, Jan. 2021.

[44] C. Murguia, I. Shames, F. Farokhi, D. Nešić, and H. V. Poor, "On privacy of dynamical systems: An optimal probabilistic mapping approach," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2608–2620, Feb. 2021.

[45] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[46] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5677–5690, Aug. 2018.

[47] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, Nov. 2019.

[48] W. Wang, D. Li, X. Wu, and S. Xue, "Average consensus for switching topology networks with privacy protection," in *Proc. IEEE Chinese Automat. Congr.*, 2019, pp. 1098–1102.

[49] J. Le Ny, "Differentially private Kalman filtering," in *Differential Privacy for Dynamic Data*. Springer, 2020, pp. 55–75.

[50] K. H. Degue and J. Le Ny, "On differentially private Kalman filtering," in *Proc. 5th IEEE Global Conf. Signal and Inf. Process.*, 2017, pp. 487–491.

[51] Y. Song, C. X. Wang, and W. P. Tay, "Privacy-aware kalman filtering," in *Proc. 43rd IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2018, pp. 4434–4438.

[52] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[53] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Comput. Surveys*, vol. 51, no. 3, pp. 1–38, Jun. 2018.

[54] C. Wang, E. K. Au, R. D. Murch, W. H. Mow, R. S. Cheng, and V. Lau, "On the performance of the mimo zero-forcing receiver in the presence of channel estimation error," *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 805–810, Mar. 2007.

[55] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Prentice Hall Signal Process. Ser. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

Ashkan Moradi received the M.Sc. degree in Telecommunication Networks from University of Tehran, Iran, in 2016. He is currently pursuing a Ph.D. degree at the Department of Electronic Systems at the Norwegian University of Science and Technology (NTNU). His expertise and research interests include distributed filtering, estimation, and learning algorithms in resource-constrained networks, with an emphasis on agent privacy and data security. Currently, he is on a research visit at the Technical University of Munich in Germany.

Naveen K. D. Venkategowda (S'12–M'17) received the B.E. degree in electronics and communication engineering from Bangalore University, Bengaluru, India, in 2008, and the Ph.D. degree in electrical engineering from Indian Institute of Technology, Kanpur, India, in 2016. He is currently an Universitetslektor at the Department of Science and Technology, Linköping University, Sweden. From Oct. 2017 to Feb. 2021, he was postdoctoral researcher at the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway. He was a Research Professor at the School of Electrical Engineering, Korea University, South Korea from Aug. 2016 to Sep. 2017. He was a recipient of the TCS Research Fellowship (2011-15) from TCS for graduate studies in computing sciences and the ERCIM Alain Bensoussan Fellowship in 2017.

Sayed Pouria Talebi received his PhD degree in statistical signal processing from Imperial College London, London-U.K., where his main research focus was on the development of quaternion-valued distributed signal processing techniques. He has since been a postdoctoral research fellow at Aalto University, Espoo-Finland, and NTNU, Trondheim-Norway. In addition, he has served as an invited researcher at University of Cambridge, Cambridge-U.K., where his research focus has been on Bayesian inference and adaptive filtering. His current research interests include distributed estimation and control, fractional-order learning systems, optimisation, machine learning, as well as, high-dimensional algebras for control and signal processing applications.

Stefan Werner (SM'07) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998, and the D.Sc. degree (Hons.) in electrical engineering from the Signal Processing Laboratory, Helsinki University of Technology, Espoo, Finland, in 2002. He is currently a Professor at the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Director of IoT@NTNU, and Adjunct Professor with Aalto University in Finland. He was a visiting Melchor Professor with the University of Notre Dame during the summer of 2019 and an Adjunct Senior Research Fellow with the Institute for Telecommunications Research, University of South Australia, from 2014 to 2020. He held an Academy Research Fellowship, funded by the Academy of Finland, from 2009 to 2014. His research interests include adaptive and statistical signal processing, wireless communications, and security and privacy in cyber-physical systems. He is a member of the editorial boards for the EURASIP Journal of Signal Processing and the IEEE Transactions on Signal and Information Processing over Networks.

Distributed Kalman Filtering with Privacy against Honest-but-Curious Adversaries

Ashkan Moradi*, Naveen K. D. Venkategowda†, Sayed Pouria Talebi*, and Stefan Werner*
*Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway
E-mail: {ashkan.moradi, pouria, stefan.werner}@ntnu.no .
†Linköping University, Norrköping, Sweden, E-mail: naveen.venkategowda@liu.se.

***Abstract*—This paper proposes a privacy-preserving distributed Kalman filter (PP-DKF) to protect the private information of individual network agents from being acquired by honest-but-curious (HBC) adversaries. The proposed approach endows privacy by incorporating noise perturbation and state decomposition. In particular, the PP-DKF provides privacy by restricting the amount of information exchanged with decomposition and concealing private information from adversaries through perturbation. We characterize the performance and convergence of the proposed PP-DKF and demonstrate its robustness against perturbation. The resulting PP-DKF improves agent privacy, defined as the mean squared estimation error of private data at the HBC adversary, without significantly affecting the overall filtering performance. Several simulation examples corroborate the theoretical results.**

***Index Terms*—Estimation, privacy, information fusion, average consensus, distributed Kalman filtering, multiagent systems.**

I. Introduction

Distributed Kalman filters (DKFs) have gained increased attention due to their high accuracy and computational efficiency for learning and estimation tasks in multiagent systems [1]–[4]. In general, distributed Kalman filtering techniques are based on agents running local Kalman filters and consensus operations to fuse observation and state information [5]–[7]. Although local cooperation among agents in distributed settings facilitates the fusion process, it causes undesirable information disclosure. Thus, the vulnerability of distributed procedures to potential eavesdroppers turns privacy preservation into an urgent issue to tackle in many applications [8]–[10].

Various methods are present to address privacy issues in distributed consensus operations in the literature. Differential privacy (DP) techniques, for example, use uncorrelated noise sequences within information exchange protocols to protect individual information [10], [11]. Alternatively, more recent noise injection-based methods achieve a better privacy-accuracy trade-off by perturbing the information exchanged with noise [12]–[14]. Further, decomposition-based techniques provide privacy by restricting the amount of information that is shared with other agents [15], [16].

Using DP to protect individual data streams in a system theoretic context where sensor measurements are transmitted to a fusion center was first addressed in [17]. In [18], a general approach is presented to design a differentially private Kalman filter in both cases of perturbation before exchanging information with the fusion center as well as output perturbation that injects noise to the output of the Kalman filter. In addition, the authors in [19] demonstrate that combining the input signals before adding DP noises, except for privacy, enhances the Kalman filtering performance. The privacy-aware Kalman filter proposed in [20] partitions sensor measurements into private and public substates to maximize the estimation error of the private state and minimize that for the public state. Although most literature discusses centralized filtering settings with external adversaries [17]–[20], in the context of distributed filtering applications, honest-but-curious (HBC) adversaries use local information to infer private data. An HBC adversary is a network agent that participates in the filtering process, but is curious and tries to retrieve private information from other agents. Although literature includes studies related to privacy-preserving Kalman filtering techniques, no attention has been paid to a privacy-preserving framework for distributed Kalman filtering strategies.

This paper proposes a privacy-preserving distributed Kalman filter that incorporates both noise injection-based and decomposition-based average consensus techniques to achieve privacy against HBC adversaries. In the proposed PP-DKF, agents decompose their private information into public and private substates, where only the public substate is shared with neighbors. A noise sequence perturbs the public substate before being shared with neighbors to provide an additional layer of protection. The proposed PP-DKF enhances filtering performance when compared to DKFs employing contemporary privacy-preserving techniques, showing that the method is more robust to noise-injection. Additionally, the PP-DKF improves the privacy level for all agents, defined as the mean squared estimation error of private data at the adversary [21].

Mathematical Notations: Scalars, column vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters, while $\mathbf{I}$, and $\mathbf{0}$ represent identity and zero matrices, respectively. The transpose and statistical expectation operators are denoted by $(\cdot)^{\mathsf{T}}$ and $\mathbb{E}\{\cdot\}$, while $\otimes$ denotes the matrix Kronecker product. The trace operator is denoted as $\mathsf{tr}(\cdot)$, $\mathsf{diag}(\mathbf{a})$ denotes diagonal matrix whose diagonals are the elements of vector $\mathbf{a}$, and the $\mathsf{Blockdiag}(\{\mathbf{A}_i\}_{i=1}^{N})$ represents a block diagonal matrix containing $\mathbf{A}_i$s on the main diagonal.

This work was supported in part by the Research Council of Norway.

II. Problem Formulation

We consider a set of N interconnected agents that is modeled as a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with node set $\mathcal{N}$, representing agents, and edge set $\mathcal{E}$, representing communication links. The neighborhood of agent i is denoted by $\mathcal{N}_i$, with cardinality N_i. We revisit the classical DKF to track a dynamic system state through observations from a network of agents [2], [3], [6]. The state-space model is given by

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{v}_n \tag{1}$$

$$\mathbf{y}_{i,n} = \mathbf{H}_i\mathbf{x}_n + \mathbf{w}_{i,n} \tag{2}$$

where for time instant n and agent i, $\mathbf{A}$ denotes the state transition matrix, and $\mathbf{H}_i$ denotes the observation matrix, $\mathbf{y}_{i,n}$ is the local observation, and $\mathbf{w}_{i,n}$, $\mathbf{v}_n$, are observation and process noises, respectively. The process noise and observation noise are mutually independent white Gaussian sequences with covariance matrices $\mathbf{C}_{\mathbf{v}_n}$ and $\mathbf{C}_{\mathbf{w}_{i,n}}$, respectively. The proposed PP-DKF is based on the DKF in [5], which requires agents to share local estimates with neighbors and reach a network-wide consensus by local collaboration. Since the shared data includes private information, we propose a PP-DKF that safeguards the private information of individual agents from being estimated by HBC adversaries.

III. Privacy-Preserving Distributed Kalman Filter

Based on the proposed DKF in [5], the proposed PP-DKF tracks a dynamic system state by updating the local model given by

$$\begin{aligned}\hat{\mathbf{x}}_{i,n|n-1} &= \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}\\ \mathbf{M}_{i,n|n-1} &= \mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^T + \mathbf{C}_{\mathbf{v}_n}\end{aligned} \tag{3}$$

where, for agent i, $\hat{\mathbf{x}}_{i,n|n-1}$ and $\hat{\mathbf{x}}_{i,n|n}$ are the respective *a priori* and *a posteriori* state vector estimates and the covariance information of agent i, at time instant n is denoted by $\mathbf{M}_{i,n|n-1}$. Following the centralized Kalman filter in [6], the local covariance information of agent i at time instant n is updated as

$$\mathbf{\Gamma}_{i,n} = \mathbf{M}^{-1}_{i,n|n-1} + N\mathbf{H}_i^T\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}\mathbf{H}_i. \tag{4}$$

Updating the covariance information requires sharing the local covariance $\mathbf{\Gamma}_{i,n}$ to reach the average consensus among agents as $\mathbf{M}^{-1}_{i,n|n} = \frac{1}{N}\sum_{i\in\mathcal{N}}\mathbf{\Gamma}_{i,n}$. The local covariance is not considered as private information and it can be implemented in a distributed manner by employing an average consensus filter (ACF) with K consensus iterations as

$$\mathbf{M}^{-1}_{i,n|n} = \mathbf{\Gamma}_{i,n}(K) \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \mathbf{\Gamma}_{j,n}(0) = \mathbf{\Gamma}_{j,n}\}$$

where the operation at each consensus iteration k is given as $\mathbf{\Gamma}_{i,n}(k) = q_{ii}\mathbf{\Gamma}_{i,n}(k-1) + \sum_{j\in\mathcal{N}_i} q_{ij}\mathbf{\Gamma}_{j,n}(k-1)$ with consensus weights satisfying $q_{ii} + \sum_{j\in\mathcal{N}_i} q_{ij} = 1$ for each agent i. It is assumed that the conditions for convergence of $\mathbf{M}_{i,n|n}$ for all agents are satisfied, as given in [5]. The updated covariance is then used to evolve the intermediate state vector estimate of agent i at time instant n as

$$\boldsymbol{\psi}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right) \tag{5}$$

where $\mathbf{G}_{i,n} = N\mathbf{M}_{i,n|n}\mathbf{H}_i^T\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}$ is the local gain. Subsequently, the state vector estimate needs to reach the average consensus among agents as $\hat{\mathbf{x}}_{i,n|n} = \frac{1}{N}\sum_{i\in\mathcal{N}}\boldsymbol{\psi}_{i,n}$, which requires agents to share their intermediate state vector estimate $\boldsymbol{\psi}_{i,n}$ among neighbors. Since $\boldsymbol{\psi}_{i,n}$ includes private information, it needs to be protected from adversaries.

To reach the average consensus of intermediate state vector estimates, the PP-DKF instructs each agent i to decompose its initial information $\mathbf{r}_{i,n}(0) = \boldsymbol{\psi}_{i,n}$ into public and private substates $\boldsymbol{\alpha}_{i,n}(0)$ and $\boldsymbol{\beta}_{i,n}(0)$, respectively. The initial substates are chosen such that $\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0) = 2\mathbf{r}_{i,n}(0)$ is satisfied [15]. The public substate $\boldsymbol{\alpha}_{i,n}$ is shared with neighbors, while the private substate $\boldsymbol{\beta}_{i,n}$ evolves internally without being observed by neighbors. We perturb the public substate before sharing with neighbors with a noise sequence $\boldsymbol{\omega}_i(k)$ at the ith agent and kth consensus iteration in order to further protect the private information. The designed noise structure is

$$\boldsymbol{\omega}_i(k) = \phi^k\boldsymbol{\nu}_i(k) - \phi^{k-1}\boldsymbol{\nu}_i(k-1),\ \forall k \geq 1 \tag{6}$$

where $\boldsymbol{\omega}_i(0) = \boldsymbol{\nu}_i(0)$, $\boldsymbol{\nu}_i(k) \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is an independent and identically distributed Gaussian sequence for each $i \in \mathcal{N}$, and $\phi \in (0,1)$ is a common constant. As a result, each agent i updates its substates at the kth consensus iteration by injecting (6) into the public substate before sharing with the neighbors as follows:

$$\begin{cases}\boldsymbol{\alpha}_{i,n}(k+1) = & \boldsymbol{\alpha}_{i,n}(k) + \varepsilon\sum\limits_{j\in\mathcal{N}_i} w_{ij}\left(\tilde{\boldsymbol{\alpha}}_{j,n}(k) - \boldsymbol{\alpha}_{i,n}(k)\right)\\ & +\varepsilon\mathbf{U}_i\left(\boldsymbol{\beta}_{i,n}(k) - \boldsymbol{\alpha}_{i,n}(k)\right)\\ \boldsymbol{\beta}_{i,n}(k+1) = & \boldsymbol{\beta}_{i,n}(k) + \varepsilon\mathbf{U}_i\left(\boldsymbol{\alpha}_{i,n}(k) - \boldsymbol{\beta}_{i,n}(k)\right)\end{cases} \tag{7}$$

where $\tilde{\boldsymbol{\alpha}}_{j,n}(k) = \boldsymbol{\alpha}_{j,n}(k) + \boldsymbol{\omega}_j(k)$ is the received information from the jth neighbor and $\varepsilon \in (0, 1/(\Delta+1)]$ with $\Delta \triangleq \max_{i\in\mathcal{N}} N_i$ is the consensus parameter. The interaction weight is denoted by w_{ij}, while $\mathbf{U}_i \triangleq \mathsf{diag}(\mathbf{u}_i)$ is a diagonal matrix containing the the coupling weight vector of the ith agent. The coupling weight vector $\mathbf{u}_i \in \mathbb{R}^m$ contains independent elements that control the level of contribution of each substate in the updating procedure. In addition, we require a scalar $\eta \in (0,1)$, such that all nonzero $w_{ij} = w_{ji}$ and all elements of $\mathbf{u}_i$ reside in the range $[\eta, 1)$, [15]. After repeating the steps in (7) for a sufficient number of iterations, say K iterations, the local state estimate, $\hat{\mathbf{x}}_{i,n|n}$, is updated as $\hat{\mathbf{x}}_{i,n|n} = \boldsymbol{\alpha}_{i,n}(K)$ for all $i \in \mathcal{N}$. The operations of the proposed PP-DKF is summarized in Algorithm 1.

Theorem 1: The privacy-preserving average consensus operations in Algorithm 1 converges to the exact average consensus value, asymptotically.

$$\lim_{k\to\infty}\mathbb{E}\{\boldsymbol{\alpha}_{i,n}(k)\} = \lim_{k\to\infty}\mathbb{E}\{\boldsymbol{\beta}_{i,n}(k)\} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}_{i,n}. \tag{8}$$

Proof: To show the convergence of the derived privacy-preserving ACF operations to the exact average consensus value, we first show that the sum of all substates is constant,

Algorithm 1: Privacy-Preserving Distributed Kalman Filter

```
**Model update:** For each $i \in \mathcal{N}$
$\hat{\mathbf{x}}_{i,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$
$\mathbf{M}_{i,n|n-1} = \mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^T + \mathbf{C}_{\mathbf{v}_n}$
$\mathbf{\Gamma}_{i,n} = \mathbf{M}^{-1}_{i,n|n-1} + N\mathbf{H}_i^T\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}\mathbf{H}_i$
$\mathbf{M}^{-1}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \mathbf{\Gamma}_{j,n}\}$
$\mathbf{G}_{i,n} = N\mathbf{M}_{i,n|n}\mathbf{H}_i^T\mathbf{C}^{-1}_{\mathbf{w}_{i,n}}$
$\boldsymbol{\psi}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)$
Set $\mathbf{r}_{i,n}(0) = \boldsymbol{\psi}_{i,n}$
**Privacy-Preserving ACF:**
Select $\boldsymbol{\alpha}_{i,n}(0)$ and set $\boldsymbol{\beta}_{i,n}(0) = 2\mathbf{r}_{i,n}(0) - \boldsymbol{\alpha}_{i,n}(0)$
Share $\tilde{\boldsymbol{\alpha}}_{i,n}(0) = \boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\omega}_i(0)$
**for** $k = 1, \ldots, K$ **do**
    Receive $\tilde{\boldsymbol{\alpha}}_{j,n}(k-1)$, $\forall j \in \mathcal{N}_i$
    Update $\boldsymbol{\alpha}_{i,n}(k)$ and $\boldsymbol{\beta}_{i,n}(k)$, as given in (7)
    Share $\tilde{\boldsymbol{\alpha}}_{i,n}(k) = \boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\omega}_i(k)$
**end**
$\hat{\mathbf{x}}_{i,n|n} = \boldsymbol{\alpha}_{i,n}(K)$
```

asymptotically [15]. The sum of all substates at the kth iteration is defined as $\boldsymbol{\zeta}_n(k) \triangleq \sum_{i=1}^N(\boldsymbol{\alpha}_{i,n}(k) + \boldsymbol{\beta}_{i,n}(k))$ where

$$\boldsymbol{\zeta}_n(k) = \boldsymbol{\zeta}_n(0) + \varepsilon\sum_{i=1}^{N}\sum_{l=1}^{k-1} d_i\,\boldsymbol{\omega}_i(l).$$

with $d_i = \sum_{j\in\mathcal{N}_i} w_{ij}$. Given the zero mean and decaying covariance properties of the designed noise (6), $\boldsymbol{\zeta}_n(k)$ converges to $\boldsymbol{\zeta}_n(0)$ in the mean square sense which is $\lim_{k\to\infty}\mathbb{E}\{\|\boldsymbol{\zeta}_n(k) - \boldsymbol{\zeta}_n(0)\|^2\} = \mathbf{0}$. Subsequently, due to the connected network assumption and considering that $\boldsymbol{\alpha}_{i,n}(0) + \boldsymbol{\beta}_{i,n}(0) = 2\boldsymbol{\psi}_{i,n}$, the ith agent substates, $\boldsymbol{\alpha}_{i,n}$ and $\boldsymbol{\beta}_{i,n}$, converge to the desired average consensus value [15], as in (8). ■

IV. Performance Evaluation

With the equivalent network model of $2N$ agents, each private substate corresponds to an agent only attached to its peer in the original network, we evaluate the effects of incorporating privacy-preserving operations on the filtering performance. It is assumed that the imaginary agents have the same observation parameters, $\mathbf{y}_{i,n}$, $\mathbf{H}_i$, and $\mathbf{C}_{\mathbf{w}_i}$, with their original peers. We also assume that agents start privacy-preserving steps with equal substates, $\boldsymbol{\alpha}_{i,n}(0) = \boldsymbol{\beta}_{i,n}(0) = \boldsymbol{\psi}_{i,n}$, so that their intermediate estimation error is equal to

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n - \boldsymbol{\alpha}_{i,n}(0) \qquad i = 1,\cdots,N\\ \boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n - \boldsymbol{\beta}_{i-N,n}(0) \quad i = N+1,\cdots,2N\end{aligned}$$

Based on the local observation in (2) and substituting the intermediate state in (5), the intermediate estimation error of each agent, $\boldsymbol{\epsilon}_{i,n} = \mathbf{x}_n - \boldsymbol{\psi}_{i,n}$, is formulated as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n - \hat{\mathbf{x}}_{i,n|n-1} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i\left(\mathbf{x}_n - \hat{\mathbf{x}}_{i,n|n-1}\right)\\ &- N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{w}_{i,n} \qquad (9)\\ =& \left(\mathbf{I} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i\right)\mathbf{A}\boldsymbol{\epsilon}_{i,n-1|n-1} \qquad (10)\\ &+ \left(\mathbf{I} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i\right)\mathbf{v}_n - \mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{w}_{i,n}.\end{aligned}$$

where $\boldsymbol{\epsilon}_{i,n-1|n-1} = \mathbf{x}_{n-1} - \hat{\mathbf{x}}_{i,n-1|n-1}$. Assuming the stacked vectors organizing all error terms as $\boldsymbol{\mathcal{E}}_n \triangleq [\boldsymbol{\epsilon}^{\mathrm{T}}_{1,n},\cdots,\boldsymbol{\epsilon}^{\mathrm{T}}_{2N,n}]^{\mathrm{T}}$ and $\boldsymbol{\mathcal{E}}_{n-1|n-1} \triangleq [\boldsymbol{\epsilon}^{\mathrm{T}}_{1,n-1|n-1},\cdots,\boldsymbol{\epsilon}^{\mathrm{T}}_{2N,n-1|n-1}]^{\mathrm{T}}$, the network-wide state vector estimation error, $\boldsymbol{\mathcal{E}}_{n|n}$, which is the stacked error after the privacy-preserving ACF operations in (7) with k consensus iterations, is formulated as

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} =& \mathbf{G}^k\boldsymbol{\mathcal{E}}_n + \phi^{k-1}\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-1) \qquad (11)\\ &+ \sum_{s=2}^{k}\phi^{k-s}\left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-s)\end{aligned}$$

where $\boldsymbol{\nu}(k) = [\boldsymbol{\nu}_1^{\mathrm{T}}(k),\cdots,\boldsymbol{\nu}_N^{\mathrm{T}}(k)]^{\mathrm{T}}$, $\boldsymbol{\mathcal{B}} = [\varepsilon\mathbf{W},\mathbf{0}]^{\mathrm{T}}\otimes\mathbf{I}$, and $\mathbf{G}$ is a doubly stochastic matrix given by

$$\mathbf{G} = \begin{bmatrix}\mathbf{M} & \varepsilon\mathbf{U}\\ \varepsilon\mathbf{U} & \mathbf{I} - \varepsilon\mathbf{U}\end{bmatrix} \qquad (12)$$

with $\mathbf{M} \triangleq (\mathbf{I}_N - \varepsilon(\mathbf{D} - \mathbf{W}))\otimes\mathbf{I} - \varepsilon\mathbf{U}$, $\mathbf{U} = \mathsf{Blockdiag}(\{\mathbf{U}_i\}_{i=1}^N)$, $\mathbf{D} = \mathsf{diag}(\{d_i\}_{i=1}^N)$, and $\mathbf{W}$ as the interaction weight matrix consisting all weights w_{ij}. Substituting the network-wide intermediate state vector estimation error $\boldsymbol{\mathcal{E}}_n$ from (10) into (11) results

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} =& \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{E}}_{n-1|n-1} + \boldsymbol{\theta}_n - \boldsymbol{\mu}_n + \phi^{k-1}\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-1)\\ &+ \sum_{s=2}^{k}\phi^{k-s}\left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)\boldsymbol{\mathcal{B}}\boldsymbol{\nu}(k-s) \qquad (13)\end{aligned}$$

where $\boldsymbol{\mathcal{P}} = \mathbf{G}^k\mathsf{Blockdiag}(\{\mathbf{P}_i\mathbf{A}\}_{i=1}^{2N})$ and

$$\begin{aligned}\boldsymbol{\theta}_n &= \mathbf{G}^k\mathsf{Blockdiag}(\{\mathbf{P}_i\}_{i=1}^{2N})[\mathbf{v}_n^{\mathrm{T}},\cdots,\mathbf{v}_n^{\mathrm{T}}]^{\mathrm{T}}\\ \boldsymbol{\mu}_n &= \mathbf{G}^k\mathsf{Blockdiag}(\{\mathbf{Q}_i\}_{i=1}^{2N})[\mathbf{w}_{1,n}^{\mathrm{T}},\cdots,\mathbf{w}_{2N,n}^{\mathrm{T}}]^{\mathrm{T}}\end{aligned}$$

with $\mathbf{P}_i = \mathbf{I} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}\mathbf{H}_i$ and $\mathbf{Q}_i = \mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}^{-1}_{\mathbf{w}_i}$. Since $\mathbf{P}_i$ is stable and $\mathbf{G}$ is doubly stochastic, the block matrix $\boldsymbol{\mathcal{P}}$ is stable; thus, the statistical expectation of any vector norm for $\boldsymbol{\mathcal{E}}_{n|n}$ converges to a stabilizing value, as $n \to \infty$. Taking the statistical expectation of (13) yields

$$\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\} = \boldsymbol{\mathcal{P}}\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n-1|n-1}\} = \boldsymbol{\mathcal{P}}^n\mathbb{E}\{\boldsymbol{\mathcal{E}}_{0|0}\}.$$

Since $\boldsymbol{\mathcal{P}}$ is stable, we have $\lim_{n\to\infty}\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\} = 0$ that indicates the steady-state estimates are unbiased regardless of their initializing values or perturbation sequences.

The second-order statistics of all agents is formulated by defining $\mathbf{\Sigma}_n = \mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\boldsymbol{\mathcal{E}}_{n|n}^{\mathrm{T}}\}$ and given by

$$\begin{aligned}\mathbf{\Sigma}_n =& \boldsymbol{\mathcal{P}}\mathbf{\Sigma}_{n-1}\boldsymbol{\mathcal{P}}^{\mathrm{T}} + \mathbb{E}\{\boldsymbol{\theta}_n\boldsymbol{\theta}_n^{\mathrm{T}}\} + \mathbb{E}\{\boldsymbol{\mu}_n\boldsymbol{\mu}_n^{\mathrm{T}}\}\\ &+ \sum_{s=2}^{k}\phi^{2(k-s)}\boldsymbol{\mathcal{T}}_s + \phi^{2(k-1)}\boldsymbol{\mathcal{B}}\mathbf{C}_{\nu}\boldsymbol{\mathcal{B}}^{\mathrm{T}} \qquad (14)\end{aligned}$$

with $\mathbf{C}_\nu = \mathbb{E}\{\boldsymbol{\nu}(s)\boldsymbol{\nu}^{\mathrm{T}}(s)\}$ at each consensus iteration s and $\boldsymbol{\mathcal{T}}_s = \left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)\boldsymbol{\mathcal{B}}\mathbf{C}_\nu\boldsymbol{\mathcal{B}}^{\mathrm{T}}\left(\mathbf{G}^{s-1} - \mathbf{G}^{s-2}\right)^{\mathrm{T}}$. Since $\mathbf{G}$ is doubly stochastic and $\boldsymbol{\mathcal{P}}$ is stable, $\mathbf{\Sigma}_n \to \mathbf{\Sigma}$ as $n \to \infty$, where $\mathbf{\Sigma}$ is the solution of the discrete-time Lyapunov equation in (14). Compared with the non-private approach, the effect of injected noise is manifested as a rise in the steady-state mean square error (MSE) of Algorithm 1. In the next section, we examine the performance of the derived framework to preserve agent privacy.

V. Privacy Analysis

We consider an HBC agent that can access the interaction weights and exchanged information of its neighbors. To benchmark the privacy of the derived PP-DKF, we consider the MSE associated with the estimates of the initial states $\boldsymbol{\psi}_n = [\boldsymbol{\psi}_{1,n}^{\mathrm{T}}, \cdots, \boldsymbol{\psi}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}$ at the HBC agent as a privacy metric. Without loss of generality, we assume that the Nth agent is an HBC agent that attempts to estimate the initial states of all agents using the accessible information set $\mathcal{I}(k) = \{\boldsymbol{\alpha}_{N,n}(k), \boldsymbol{\beta}_{N,n}(k), \boldsymbol{\omega}_N(k), \mathbf{u}_N, w_{Nj}, \tilde{\boldsymbol{\alpha}}_{j,n}(k) : \forall j \in \mathcal{N}_N\}$ at each consensus iteration k. We introduce the observation vector $\mathbf{y}_n(k)$ that includes the accessible information transferred to the HBC agent at each iteration k as

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{z}_n(k) + \mathbf{C}_\alpha \boldsymbol{\omega}(k) \tag{15}$$

where $\mathbf{C} \triangleq [\mathbf{C}_\alpha, \mathbf{C}_\beta]$ with $\mathbf{C}_\beta = [\mathbf{0}, \mathbf{e}_N]^{\mathrm{T}} \otimes \mathbf{I}$ and $\mathbf{C}_\alpha = [\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_{N_N}}, \mathbf{e}_N]^{\mathrm{T}} \otimes \mathbf{I}$. The canonical basis $\mathbf{e}_i \in \mathbb{R}^N$ is a vector with 1 in the ith entry and zeros elsewhere, while $\mathbf{z}_n(k) \triangleq [\boldsymbol{\alpha}_n^{\mathrm{T}}(k), \boldsymbol{\beta}_n^{\mathrm{T}}(k)]^{\mathrm{T}}$ with the network-wide agent substate vectors given as

$$\boldsymbol{\alpha}_n(k) \triangleq [\boldsymbol{\alpha}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\alpha}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}}$$
$$\boldsymbol{\beta}_n(k) \triangleq [\boldsymbol{\beta}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\beta}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}}.$$

The estimated value of $\mathbf{z}_n(0)$, i.e., $\hat{\mathbf{z}}_n(0) \triangleq [\hat{\boldsymbol{\alpha}}_n^{\mathrm{T}}(0), \hat{\boldsymbol{\beta}}_n^{\mathrm{T}}(0)]^{\mathrm{T}}$, is then used to estimate the initial state of the agents as $\hat{\boldsymbol{\psi}}_n = \frac{1}{2}(\hat{\boldsymbol{\alpha}}_n(0) + \hat{\boldsymbol{\beta}}_n(0))$. Substituting the network-wide substate update equations in (7) into (15) results

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{G}^k\mathbf{z}_n(0) + \mathbf{C}_\alpha \sum_{t=0}^{k-1} \boldsymbol{\mathcal{C}}_{k-1-t}\mathbf{B}\boldsymbol{\omega}(t) + \mathbf{C}_\alpha\boldsymbol{\omega}(k) \tag{16}$$

where $\boldsymbol{\mathcal{C}}_k = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \mathbf{G}^k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}^{\mathrm{T}}$ and $\mathbf{B} = \varepsilon\mathbf{W} \otimes \mathbf{I}$. Since $\boldsymbol{\nu}(k)$ is a zero-mean i.i.d. sequence, the accumulated observation of the HBC agent set-up at consensus iteration k, $\tilde{\mathbf{y}}_n(k) = \sum_{t=0}^{k} \mathbf{y}_n(t)$, is simplified as

$$\tilde{\mathbf{y}}_n(k) = \mathbf{C}(\mathbf{I} - \mathbf{G})^{k+1}(\mathbf{I} - \mathbf{G})^{-1}\mathbf{z}_n(0) + \mathbf{C}_\alpha\tilde{\boldsymbol{\nu}}(k) \tag{17}$$

where $\tilde{\boldsymbol{\nu}}(k) = \sum_{t=0}^{k-1} \phi^t \boldsymbol{\mathcal{C}}_{k-1-t}\mathbf{B}\boldsymbol{\nu}(t) + \phi^k\boldsymbol{\nu}(k)$. Stacking all the available accumulated observations at each consensus iteration k in a vector, $\bar{\mathbf{y}}_n(k) = [\tilde{\mathbf{y}}_n^{\mathrm{T}}(0), \ldots, \tilde{\mathbf{y}}_n^{\mathrm{T}}(k)]^{\mathrm{T}}$, gives

$$\bar{\mathbf{y}}_n(k) = \mathbf{H}(k)\mathbf{z}(0) + \mathbf{F}(k)\bar{\boldsymbol{\nu}}(k) \tag{18}$$

where $\mathbf{H}(k) = (\mathbf{I} \otimes \mathbf{C})[\mathbf{H}_0^{\mathrm{T}}, \mathbf{H}_1^{\mathrm{T}}, \ldots, \mathbf{H}_k^{\mathrm{T}}]^{\mathrm{T}}$ with $\mathbf{H}_k = \sum_{t=0}^{k} \mathbf{G}^t$, $\bar{\boldsymbol{\nu}}(k) = [\boldsymbol{\nu}^{\mathrm{T}}(0), \cdots, \boldsymbol{\nu}^{\mathrm{T}}(k)]^{\mathrm{T}}$, and

$$\mathbf{F}(k) = \begin{bmatrix} \mathbf{C}_\alpha & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_\alpha\boldsymbol{\mathcal{C}}_0\mathbf{B} & \phi\mathbf{C}_\alpha & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_\alpha\boldsymbol{\mathcal{C}}_{k-1}\mathbf{B} & \phi\mathbf{C}_\alpha\boldsymbol{\mathcal{C}}_{k-2}\mathbf{B} & \cdots & \phi^k\mathbf{C}_\alpha \end{bmatrix}.$$

With the perfect observation matrix $\mathbf{H}(k)$ available, the estimate of initial substates $\mathbf{z}_n(0)$ could be modeled as

$$\hat{\mathbf{z}}_n(0) = \mathbf{H}^\dagger(k)(\mathbf{H}(k)\mathbf{z}_n(0) + \mathbf{F}(k)\bar{\boldsymbol{\nu}}(k)) \tag{19}$$

where $\mathbf{H}^\dagger(k)$ is the Moore–Penrose pseudoinverse of $\mathbf{H}(k)$. However, since the HBC agent does not have access to the coupling weight matrix $\mathbf{U}$, it has to estimate the observation matrix $\mathbf{H}(k)$. Following the estimation procedure in [22], the HBC agent estimates the coupling weight matrix as $\hat{\mathbf{U}} = \mathbf{U} + \boldsymbol{\Delta}_{\mathbf{U}}$ where $\boldsymbol{\Delta}_{\mathbf{U}}$ shows its uncertainty to determine the coupling weight matrix $\mathbf{U}$.

An estimate of matrix $\mathbf{G}$ is obtained using uncertainty modeling above as $\hat{\mathbf{G}} = \mathbf{G} + \varepsilon\boldsymbol{\Delta}_{\mathbf{G}_1}$ where $\boldsymbol{\Delta}_{\mathbf{G}_1} = -\boldsymbol{\mathcal{L}}^{\mathrm{T}}\boldsymbol{\Delta}_{\mathbf{U}}\boldsymbol{\mathcal{L}}$ with $\boldsymbol{\mathcal{L}} = [-\mathbf{I}, \mathbf{I}]$. Employing the binomial expansion, the uncertainty of $\hat{\mathbf{G}}^k$ is simplified as $\hat{\mathbf{G}}^k = \mathbf{G}^k + \varepsilon\boldsymbol{\Delta}_{\mathbf{G}_k}$ where

$$\boldsymbol{\Delta}_{\mathbf{G}_k} = \sum_{t=1}^{k} \frac{k!\varepsilon^{t-1}}{(k-t)!t!}\mathbf{G}^{k-t}\boldsymbol{\Delta}_{\mathbf{G}_1}^t \ \ \forall k \geq 2.$$

Thus, estimate of the observation matrix $\mathbf{H}(k)$ is formulated as $\hat{\mathbf{H}}(k) = \mathbf{H}(k) + \varepsilon\boldsymbol{\Delta}_{\mathbf{H}}(k)$ where $\boldsymbol{\Delta}_{\mathbf{H}}(k)$ denotes the uncertainty of the observation matrix, independent of $\mathbf{H}(k)$, and is computed as $\boldsymbol{\Delta}_{\mathbf{H}}(k) = (\mathbf{I} \otimes \mathbf{C})[\mathbf{0}, \boldsymbol{\Delta}_{\mathbf{G}_1}^{\mathrm{T}}, \ldots, \sum_{t=1}^{k} \boldsymbol{\Delta}_{\mathbf{G}_t}^{\mathrm{T}}]^{\mathrm{T}}$. Subsequently, the estimate of the initial substates in (19) is reformulated as $\hat{\mathbf{z}}_n(0) = \hat{\mathbf{H}}^\dagger(k)\bar{\mathbf{y}}_n(k)$ where $\hat{\mathbf{H}}^\dagger(k) = (\mathbf{H}(k) + \boldsymbol{\Delta}_{\mathbf{H}}(k))^\dagger$. The HBC agent is a legitimate agent of the network and knows the distribution of coupling weights. Given a negligible uncertainty in $\hat{\mathbf{H}}(k)$, the pseudo-inverse of $\hat{\mathbf{H}}(k)$ can be approximated by the first order Taylor expansion as $\hat{\mathbf{H}}^\dagger(k) \cong \mathbf{H}^\dagger(k)\left(\mathbf{I} - \varepsilon\boldsymbol{\Delta}_{\mathbf{H}}(k)\mathbf{H}^\dagger(k)\right)$ and subsequently, we have $\hat{\mathbf{z}}_n(0) = \left(\mathbf{H}^\dagger(k) - \varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\mathbf{H}^\dagger(k)\right)\mathbf{y}_n(k)$ which can be further simplified as

$$\hat{\mathbf{z}}_n(0) = \mathbf{z}_n(0) + \boldsymbol{\eta}(k) \tag{20}$$

with the estimation error of the initial substates

$$\begin{aligned}\boldsymbol{\eta}(k) =& \mathbf{H}^\dagger(k)\mathbf{F}(k)\bar{\boldsymbol{\nu}}(k) - \varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\mathbf{z}_n(0) \\ &- \varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\mathbf{H}^\dagger(k)\mathbf{F}(k)\bar{\boldsymbol{\nu}}(k).\end{aligned}$$

For the worst-case scenario, when the HBC agent knows the exact coupling weights of the entire network, i.e., $\boldsymbol{\Delta}_{\mathbf{U}} = \mathbf{0}$, the estimation error covariance $\mathbf{P}(k) = \mathbb{E}\{\boldsymbol{\eta}(k)\boldsymbol{\eta}^{\mathrm{T}}(k)\}$ is computed as

$$\mathbf{P}(k) = \sigma^2\left(\mathbf{H}^{\mathrm{T}}(k)\left(\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)\right)^{-1}\mathbf{H}(k)\right)^{-1}. \tag{21}$$

As a result, the privacy of the jth agent, pertaining to estimate its initial information $\boldsymbol{\psi}_{j,n}$, is defined as

$$\mathcal{E}_j(k) \triangleq \mathrm{tr}\left((\mathbf{e}_j^{\mathrm{T}} \otimes \mathbf{I})\bar{\mathbf{P}}(k)(\mathbf{e}_j \otimes \mathbf{I})\right), \tag{22}$$

where $\bar{\mathbf{P}}(k) = \frac{1}{4}[\mathbf{I}, \mathbf{I}]\mathbf{P}(k)[\mathbf{I}, \mathbf{I}]^{\mathrm{T}}$.

VI. Numerical Results

We consider a connected network with $L = 5$ agents and edge set $\mathcal{E} = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 1)\}$. The proposed PP-DKF is considered in a collaborative target tracking application as given in [5]. To illustrate the benefits of state-decomposition and noise perturbation, characterizing the PP-DKF, we also implemented a pure noise-injection-based privacy-preserving DKF (NIP-DKF), wherein the noise sequence in (6) perturbed the shared messages of the conventional DKF in [5]. If not stated otherwise, $K = 40$ consensus iterations and $\phi = 0.9$ are employed.

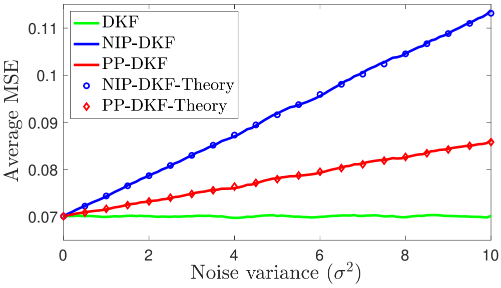


Fig. 1. Average MSE versus noise variance σ^2.

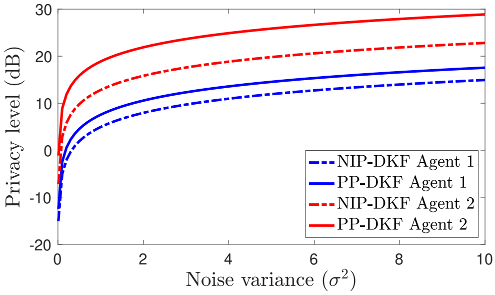


Fig. 2. Privacy metric $\mathcal{E}_j(k)$ versus noise variance σ^2.

Fig. 1 shows the average MSE of the various distributed Kalman filters versus the injected noise variance. We see that the PP-DKF has a better filtering performance than NIP-DKF and achieves an MSE close to the non-private DKF for a broad range of injected noise variances. Also, our theoretical prediction in (14) match the simulation results. The agent privacy $\mathcal{E}_j(k)$ in (22), considering the 5th agent as an HBC agent, is shown in Fig. 2. It shows that injecting more noise results in higher privacy and PP-DKF improves agent privacy compared to NIP-DKF settings. Because of the symmetric topology of the ring network, agents 3 and 4 achieve the same level of privacy as agents 2 and 1, respectively, so they are omitted from Fig.2.

VII. Conclusion

This paper proposed a privacy-preserving distributed Kalman filter that employs decomposition-based and noise injection-based privacy-preserving average consensus techniques to protect private information of agents. It restricts the amount of information exchanged with decomposition and conceals the private data from being estimated by adversaries with perturbation. The convergence and performance of the PP-DKF have been analyzed. Moreover, the achieved privacy level of each agent has been defined as the uncertainty of the honest-but-curious agent in estimating the initial state of other agents. It has been shown that the proposed PP-DKF solution improves privacy and performance of the Kalman filtering operations compared to the DKFs employing contemporary privacy-preserving consensus techniques. Lastly, several simulations verified the obtained theoretical results.

References

[1] V. Katewa, F. Pasqualetti, and V. Gupta, "On privacy vs. cooperation in multi-agent systems," *Int. J. of Control*, vol. 91, no. 7, pp. 1693–1707, Jul. 2018.

[2] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decis. and Control*, 2007, pp. 5492–5498.

[3] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sept. 2010.

[4] U. A. Khan and J. M. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[5] S. P. Talebi and S. Werner, "Distributed Kalman filtering and control through embedded average consensus information fusion," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4396–4403, Oct. 2019.

[6] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decis. and Control*, 2005, pp. 8179–8184.

[7] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. 48th IEEE Conf. Decis. and Control (CDC)*, 2009, pp. 7036–7042.

[8] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed privacy-preserving data aggregation against dishonest nodes in network systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1462–1470, Apr. 2019.

[9] A. Moradi, N. K. Venkategowda, and S. Werner, "Coordinated data-falsification attacks in consensus-based distributed Kalman filtering," in *Proc. 8th IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, 2019, pp. 495–499.

[10] J. He, L. Cai, and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 4069–4082, 2020.

[11] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, Jul. 2017.

[12] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[13] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5677–5690, Aug. 2018.

[14] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 127–138, Mar. 2019.

[15] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, Nov. 2019.

[16] W. Wang, D. Li, X. Wu, and S. Xue, "Average consensus for switching topology networks with privacy protection," in *Proc. IEEE Chinese Automat. Congr. (CAC)*, 2019, pp. 1098–1102.

[17] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.

[18] J. Le Ny, "Differentially private Kalman filtering," in *Differential Privacy for Dynamic Data*. Springer, 2020, pp. 55–75.

[19] K. H. Degue and J. Le Ny, "On differentially private Kalman filtering," in *Proc. 5th IEEE Global Conf. Signal and Inf. Process. (GlobalSIP)*, 2017, pp. 487–491.

[20] Y. Song, C. X. Wang, and W. P. Tay, "Privacy-aware kalman filtering," in *Proc. 43rd IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2018, pp. 4434–4438.

[21] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Comput. Surveys (CSUR)*, vol. 51, no. 3, pp. 1–38, Jun. 2018.

[22] C. Wang, E. K. Au, R. D. Murch, W. H. Mow, R. S. Cheng, and V. Lau, "On the performance of the mimo zero-forcing receiver in the presence of channel estimation error," *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 805–810, Mar. 2007.

Securing the Distributed Kalman Filter Against Curious Agents

Ashkan Moradi[1], Naveen K. D. Venkategowda[2], Sayed Pouria Talebi[1] and Stefan Werner[1]
[1]Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway
[2]Linköping University, Norrköping, Sweden
E-mail: {ashkan.moradi, pouria, stefan.werner}@ntnu.no, naveen.venkategowda@liu.se

***Abstract*—Distributed filtering techniques have emerged as the dominant and most prolific class of filters used in modern monitoring and surveillance applications, such as smart grids. As these techniques rely on information sharing among agents, user privacy and information security have become a focus of concern. In this manuscript, a privacy-preserving distributed Kalman filter (PP-DKF) is derived that maintains privacy by decomposing the information into public and private substates, where only a perturbed version of the public substate is shared among neighbors. The derived PP-DKF provides privacy by restricting the amount of information exchanged with state decomposition and conceals private information by injecting a carefully designed perturbation sequence. A thorough analysis is performed to characterize the privacy-accuracy trade-offs involved in the distributed filter, with privacy defined as the mean squared estimation error of the private information at the honest-but-curious agent. The resulting PP-DKF improves the overall filtering performance and privacy of all agents compared to distributed Kalman filters employing contemporary privacy-preserving average consensus techniques. Several simulation examples corroborate the theoretical results.**

***Index Terms*—Estimation, privacy, information fusion, average consensus, distributed Kalman filtering, multiagent systems.**

I. Introduction

Distributed Kalman filtering algorithms became popular for learning and estimation in multiagent systems [1], [2] due to their high accuracy and computational efficiency [3]–[5]. In general, distributed Kalman filtering techniques are based on agents of a sensor network implementing local Kalman filtering operations using their observed data. Agents then employ consensus techniques to fuse local and neighbor estimates [6]–[8]. However, the local interactions between agents in distributed filtering settings raise concerns regarding privacy and demands for secure distributed filtering [9], [10]. Although local cooperation among agents in distributed filtering facilitates the fusion process, it causes undesirable information disclosures [11]. This vulnerability of distributed filters to potential adversaries has made privacy preservation one of the most pressing subjects in many applications [12]–[18].

The literature contains various methods to address the privacy issues in distributed consensus operations. For example, differential privacy techniques inject uncorrelated noise sequences into information exchange procedures to provide privacy for individual information [13], [14]. In addition, the more recent noise injection-based average consensus techniques achieve an improved privacy-accuracy trade-off by perturbing the information exchanged with noise [15]–[17]. Decomposition-based privacy-preserving techniques, on the other hand, are based on altering the amount of information shared with other agents [19], [20].

In particular, privacy in a system theoretic context, where sensor measurements are transmitted to a fusion center, was first addressed in [9]. The work therein considers the notion of privacy characterized by differential privacy, which protects individual data streams. Subsequently, the work in [21] presents a general approach to design a differentially private Kalman filter in both cases of perturbation before exchanging information with fusion center and output perturbation that injects noise to the output of the Kalman filter. The authors in [22] show that adequately combining the input signals before adding the differential privacy noise can improve the Kalman filter performance.

The privacy-aware centralized Kalman filter proposed in [23] partitions sensor measurements into private and public substates to maximize the estimation error of the private portion while minimizing the estimation error of the public substate. The works in [9], [21]–[23] mainly consider a centralized filtering setting with external adversaries; however, in the context of distributed filtering applications, honest-but-curios adversaries employ local information to infer private data. An honest-but-curious adversary is a legitimate network agent taking part in the filtering process but is curious and attempts to retrieve the private information of other agents. Although considerable research has been devoted to privacy in centralized Kalman filtering solutions, the dilemma of privacy-preserving distributed Kalman filters against honest-but-curious agents has not been appropriately addressed.

In this paper, a privacy-preserving distributed Kalman filtering solution is derived. The derived framework draws upon the ideas from both noise injection and decomposition-based average consensus strategies. In this setting, agents decompose their acquired information into public and private substates, sharing only the perturbed version of their public substate with their neighbors. The private substate evolves internally and will not be shared with neighbors. This process is designed to provide enhanced privacy, defined as the mean squared estimation error of private data at the honest-but-curious agent [24]. In comparison to distributed Kalman filters employing con-

This work was supported in part by the Research Council of Norway.

temporary privacy-preserving average consensus techniques, the PP-DKF derived here exhibits higher robustness against injected noise and accomplishes the filtering process with enhanced performance. The contribution of the work also includes a rigorous mathematical analysis of the convergence and performance of the derived PP-DKF, and formulating a closed-form expression for agent privacy in the presence of an honest-but-curious adversary.

Mathematical Notations: Scalars, column vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters, while $\mathbf{I}$, and $\mathbf{0}$ represent identity and zero matrices, respectively. The transpose and statistical expectation operators are denoted by $(\cdot)^{\mathrm{T}}$ and $\mathbb{E}\{\cdot\}$, while $\otimes$ denotes the matrix Kronecker product. The trace operator is denoted as $\mathsf{tr}(\cdot)$, matrix $\mathsf{diag}(\mathbf{a})$ denotes diagonal matrix whose diagonals are the elements of vector $\mathbf{a}$, and the $\mathsf{Blockdiag}(\{\mathbf{A}_i\}_{i=1}^N)$ represents a block diagonal matrix containing $\mathbf{A}_i$s on the main diagonal. A white Gaussian sequence $\mathbf{x}(k)$ with covariance $\mathbf{\Sigma}$ is represented as $\mathbf{x}(k) \sim \mathscr{N}(\mathbf{0}, \mathbf{\Sigma})$.

II. Problem Formulation

We consider a set of N interconnected agents concerned with a common task. The agents and their connections are modeled as a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ with node set $\mathcal{N}$, representing agents, and edge set $\mathcal{E}$, representing communication links. The neighborhood of agent i, denoted by $\mathcal{N}_i$, is the set of agents that agent i receives information from, which does not include agent i itself. The cardinality of the set $\mathcal{N}_i$ is denoted by N_i.

We revisit the classical distributed Kalman filtering problem of tracking a dynamic system state through observations from a network of agents [3], [4], [7]. The state-space model representing the state vector evolution and local observation function is given by

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{v}_n \tag{1}$$

$$\mathbf{y}_{i,n} = \mathbf{H}_i\mathbf{x}_n + \mathbf{w}_{i,n} \tag{2}$$

where, $\mathbf{A}$ denotes the state transition matrix and $\mathbf{H}_i$ is the ith agent observation matrix. For time instant n and agent i, $\mathbf{y}_{i,n}$ is the local observation, while $\mathbf{w}_{i,n}$ and $\mathbf{v}_n$ are observation and process noises, respectively. The process and observation noises are zero-mean Gaussian sequences with joint covariance matrices given by

$$\mathbb{E}\left\{\begin{bmatrix} \mathbf{v}_n \\ \mathbf{w}_{i,n} \end{bmatrix} \begin{bmatrix} \mathbf{v}_l^{\mathrm{T}} & \mathbf{w}_{j,l}^{\mathrm{T}} \end{bmatrix}\right\} = \begin{bmatrix} \mathbf{C}_{\mathbf{v}_n} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\mathbf{w}_{i,n}}\delta_{i,j} \end{bmatrix} \delta_{n,l}$$

where $\delta_{n,l}$ denotes the Kronecker delta function. The proposed PP-DKF is implemented based on the distributed Kalman filter (DKF) in [6] that requires agents to exchange local estimates with neighbors, and through local collaboration, to reach a network-wide consensus. Since the shared data includes private information, we propose a PP-DKF that prevents an honest-but-curious adversaries from estimating the private information of individual agents. An honest-but-curious agent is a legitimate agent of the network that is curious about private data from other agents.

III. Privacy-Preserving Distributed Kalman Filter

Considering the framework established in the distributed Kalman filtering [6], each agent implements a model update as

$$\begin{aligned} \hat{\mathbf{x}}_{i,n|n-1} &= \mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1} \\ \mathbf{M}_{i,n|n-1} &= \mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}} + \mathbf{C}_{\mathbf{v}_n} \end{aligned} \tag{3}$$

where for agent i and time instant n, $\hat{\mathbf{x}}_{i,n|n-1}$ and $\hat{\mathbf{x}}_{i,n|n}$ are the respective *a priori* and *a posteriori* estimates of the state vector. The ith agent error covariance information at time instant n is denoted by $\mathbf{M}_{i,n|n-1}$ which following the centralized Kalman filter operations in [7] is updated as

$$\mathbf{M}_{i,n|n}^{-1} = \mathbf{M}_{i,n|n-1}^{-1} + \sum_{j\in\mathcal{N}} \mathbf{H}_j^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{j,n}}^{-1}\mathbf{H}_j = \frac{1}{N}\sum_{j\in\mathcal{N}} \mathbf{\Gamma}_{j,n}. \tag{4}$$

The expression in (4) can be approximated through average consensus filters (ACFs) after a local update as

$$\mathbf{\Gamma}_{i,n} = \mathbf{M}_{i,n|n-1}^{-1} + N\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}\mathbf{H}_i.$$

The local covariance information $\mathbf{\Gamma}_{i,n}$ is not considered private, and it can be shared among neighbors to update the *a posteriori* covariance information. To this end, the covariance information $\mathbf{M}_{i,n|n}^{-1}$ is updated via an ACF by averaging the local covariance information $\mathbf{\Gamma}_{i,n}$ among neighbors. The ACF operations is represented with the following schematic [6]:

$$\mathbf{S}_{i,n}(k) \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i \cup i : \mathbf{S}_{j,n}(0)\}$$

where $\mathbf{S}_{j,n}(0)$, $j \in \mathcal{N}_i \cup i$ are the initial inputs to the ACF at node i, and $\mathbf{S}_{i,n}(k)$ is the output at node i after k iterations. The iterative operation of the consensus filter is given by

$$\mathbf{S}_{i,n}(k) = q_{ii}\mathbf{S}_{i,n}(k-1) + \sum_{j\in\mathcal{N}_i} q_{ij}\mathbf{S}_{j,n}(k-1)$$

where $\mathbf{Q} = [q_{ij}]$ is a doubly stochastic consensus weight matrix [25]. It is assumed that the conditions for convergence of $\mathbf{M}_{i,n|n}$ for all agents are satisfied (see [6]).

The updated covariance information is employed to calculate an intermediate state estimate update using the sensors observation as

$$\boldsymbol{\psi}_{i,n} = \hat{\mathbf{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right) \tag{5}$$

where $\mathbf{M}_{i,n|n}^{-1}$ is used to formulate the update gain $\mathbf{G}_{i,n} = N\mathbf{M}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}$. To improve the state estimation, agents share their intermediate state estimate $\boldsymbol{\psi}_{i,n}$ with their neighbors to reach the average consensus. The intermediate state estimate, $\boldsymbol{\psi}_{i,n}$, reveals information regarding the observations and current state vector of an agent, which is considered private. Thus, to avoid information disclosure, the average consensus of intermediate state estimates should be implemented in a privacy-preserving manner. To this end, a privacy-preserving average consensus mechanism is designed to protect the intermediate state estimates while having minimal impact on the filtering process.

Before sharing the intermediate state estimate with neighbors, the ith agent decomposes the initial state $\boldsymbol{\psi}_{i,n}(0) =$

$_{i,n}$ into public and private substates $\boldsymbol{\alpha}_{i,n}(0)$ and $\boldsymbol{\beta}_{i,n}(0)$, satisfying $\boldsymbol{\alpha}_{i,n}(0)+\boldsymbol{\beta}_{i,n}(0)=2\boldsymbol{\psi}_{i,n}(0)$, [19]. The public substate, $\boldsymbol{\alpha}_{i,n}$, is shared with neighbors, while the private substate, $\boldsymbol{\beta}_{i,n}$, evolves internally and will not be observed by neighbors. Although the private substate remains invisible to neighbors, it directly affects the evolution of the public substate. To provide an additional protection layer to the initial state of agent i, we perturb its public substate, at the kth consensus iteration, by noise sequence $\boldsymbol{\omega}_i(k)$. The perturbation-noise is a zero-mean Gaussian sequence, mutually and temporally independent among different agents, with time-dependent covariance such that

$$\boldsymbol{\omega}_i(k)\sim\mathscr{N}(\mathbf{0},\sigma_k^2\mathbf{I}),\ \ \forall i=1,2,\cdots,N. \tag{6}$$

In order to guarantee the convergence of the overall PP-DKF operations, the variance σ_k^2 is chosen to be exponentially decaying with respect to the consensus iteration k [10], [15]. Thus, as the number of consensus iterations increases, the shared data of the ith agent converges toward the average consensus value, which is common among all agents. Hence, regarding the perturbation sequence (6), the PP-DKF injects noise with higher variance to the initial substates, while substates approaching the average consensus value are perturbed with less noise. The substate updates at each agent, and consensus iteration k, are given by

$$\begin{cases}\boldsymbol{\alpha}_{i,n}(k+1)=&\boldsymbol{\alpha}_{i,n}(k)+\varepsilon\mathbf{U}_i(k)\left(\boldsymbol{\beta}_{i,n}(k)-\boldsymbol{\alpha}_{i,n}(k)\right)\\ &+\varepsilon\displaystyle\sum_{j\in\mathcal{N}_i}w_{ij}(k)\left(\tilde{\boldsymbol{\alpha}}_{j,n}(k)-\boldsymbol{\alpha}_{i,n}(k)\right)\\ \boldsymbol{\beta}_{i,n}(k+1)=&\boldsymbol{\beta}_{i,n}(k)+\varepsilon\mathbf{U}_i(k)\left(\boldsymbol{\alpha}_{i,n}(k)-\boldsymbol{\beta}_{i,n}(k)\right)\end{cases} \tag{7}$$

where $\tilde{\boldsymbol{\alpha}}_{j,n}(k)=\boldsymbol{\alpha}_{j,n}(k)+\boldsymbol{\omega}_j(k)$ is the received information from the jth neighbor, $w_{ij}(k)$ denotes the interaction weight between agent i and j at consensus iteration k, and $\mathbf{U}_i(k)\triangleq\mathsf{diag}(\mathbf{u}_i(k))$ is a diagonal matrix containing the ith agent's coupling weight vector $\mathbf{u}_i(k)\in\mathbb{R}^m$ with independent elements that controls the level of contribution of each substate in the updating procedure. The consensus parameter ε resides in the range $(0,1/(\Delta+1)]$ where $\Delta\triangleq\max_{i\in\mathcal{N}}N_i$. For $k=0$, all weights $w_{ij}(0)$ and each elements of $\mathbf{u}_i(0)$ are allowed to be arbitrarily chosen from the set of all real numbers, while satisfying $w_{ij}(0)=w_{ji}(0)$, $\forall i,j$. For $k>0$, a scalar $\eta\in(0,1)$ is required, such that all non-zero $w_{ij}(k)$ and all elements of $\mathbf{u}_i(k)$ reside in the range $[\eta,1)$, [19]. The operations of the proposed PP-DKF at each agent are summarized in Algorithm 1.

To investigate the convergence of the derived privacy-preserving ACF operations to the exact average consensus value, one can show that the sum of all substates is constant, asymptotically [19]. The sum of all substates at the kth iteration is defined as $\boldsymbol{\zeta}_n(k)\triangleq\sum_{i=1}^{N}(\boldsymbol{\alpha}_{i,n}(k)+\boldsymbol{\beta}_{i,n}(k))$ where

$$\boldsymbol{\zeta}_n(k)=\boldsymbol{\zeta}_n(0)+\varepsilon\sum_{i=1}^{N}d_{ii}\left.\sum_{l=1}^{k-1}\boldsymbol{\omega}_i(l)\right). \tag{8}$$

Algorithm 1 Privacy-Preserving Distributed Kalman Filter

```
Initialization: For each agent $i\in\mathcal{N}$
  $\hat{\mathbf{x}}_{i,0|0}=\mathbb{E}\{\mathbf{x}_0\}$
  $\mathbf{M}_{i,0|0}=\mathbb{E}\left\{(\mathbf{x}_0-\mathbb{E}\{\mathbf{x}_0\})(\mathbf{x}_0-\mathbb{E}\{\mathbf{x}_0\})^{\mathrm{T}}\right\}$
Model update:
  $\hat{\mathbf{x}}_{i,n|n-1}=\mathbf{A}\hat{\mathbf{x}}_{i,n-1|n-1}$
  $\mathbf{M}_{i,n|n-1}=\mathbf{A}\mathbf{M}_{i,n-1|n-1}\mathbf{A}^{\mathrm{T}}+\mathbf{C}_{\mathbf{v}_n}$
Measurement update:
  $\boldsymbol{\Gamma}_{i,n}=\mathbf{M}_{i,n|n-1}^{-1}+N\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}\mathbf{H}_i$
  $\mathbf{M}_{i,n|n}^{-1}\leftarrow\boxed{\text{ACF}}\leftarrow\{\forall j\in\mathcal{N}_i:\boldsymbol{\Gamma}_{j,n}\}$
  $\mathbf{G}_{i,n}=N\mathbf{M}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}$
  $_{i,n}=\hat{\mathbf{x}}_{i,n|n-1}+\mathbf{G}_{i,n}\left(\mathbf{y}_{i,n}-\mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)$
  Set $\boldsymbol{\psi}_{i,n}(0)=\boldsymbol{\psi}_{i,n}$
Privacy-Preserving Mechanism:
  Select $\boldsymbol{\alpha}_{i,n}(0)$, and set $\boldsymbol{\beta}_{i,n}(0)=2\boldsymbol{\psi}_{i,n}(0)-\boldsymbol{\alpha}_{i,n}(0)$
  Generate $\{\boldsymbol{\omega}_i(k),k=0,1,\cdots,K\}$ based on (6)
  Share $\tilde{\boldsymbol{\alpha}}_{i,n}(0)=\boldsymbol{\alpha}_{i,n}(0)+\boldsymbol{\omega}_i(0)$
  for $k=1$ to $K$ do
      Receive $\tilde{\boldsymbol{\alpha}}_{j,n}(k-1)$, $\forall j\in\mathcal{N}_i$
      Update $\boldsymbol{\alpha}_{i,n}(k)$ and $\boldsymbol{\beta}_{i,n}(k)$, as given in (7)
      Share $\tilde{\boldsymbol{\alpha}}_{i,n}(k)=\boldsymbol{\alpha}_{i,n}(k)+\boldsymbol{\omega}_i(k)$,
  end for
  $\hat{\mathbf{x}}_{i,n|n}=\boldsymbol{\alpha}_{i,n}(K)$
```

where d_{ii} is a diagonal element of matrix $\mathbf{D}\triangleq\mathsf{diag}(\{\sum_{j\in\mathcal{N}_i}w_{ij}\}_{i=1}^{N})$, to simplify the analysis, we assume that the interaction weights are time-invariant. Given the zero mean and decaying covariance properties of the designed noise (6), $\boldsymbol{\zeta}_n(k)$ converges to $\boldsymbol{\zeta}_n(0)$ in the mean sense which is

$$\lim_{k\to\infty}\mathbb{E}\{\boldsymbol{\zeta}_n(k)-\boldsymbol{\zeta}_n(0)\}=\mathbf{0}. \tag{9}$$

Due to the connected network assumption and considering that $\boldsymbol{\alpha}_{i,n}(0)+\boldsymbol{\beta}_{i,n}(0)=2\boldsymbol{\psi}_{i,n}(0)$, the ith agent substates, $\boldsymbol{\alpha}_{i,n}$ and $\boldsymbol{\beta}_{i,n}$, converge to the desired average consensus value [19], i.e.,

$$\lim_{k\to\infty}\mathbb{E}\{\boldsymbol{\alpha}_{i,n}(k)\}=\lim_{k\to\infty}\mathbb{E}\{\boldsymbol{\beta}_{i,n}(k)\}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}_{i,n}(0).$$

In practice, due to the finite number of consensus iterations, the convergence in (9) is achieved with a bounded variance that reduces the average consensus accuracy. In the next section, we analyze the impact of this consensus error on the overall performance and convergence conditions of the proposed PP-DKF.

IV. Performance Evaluation

To provide an intuitive analysis and a proper insight into the effects of incorporating the privacy-preserving operations, we consider the equivalent network of $2N$ agents so that each private substate corresponds to an agent only attached to its peer in the original network with the same observation parameters, $\mathbf{y}_{i,n}$, $\mathbf{H}_i$, and $\mathbf{C}_{\mathbf{w}_i}$ (see Fig. 1). It is assumed that agents initialize the privacy-preserving steps with equal

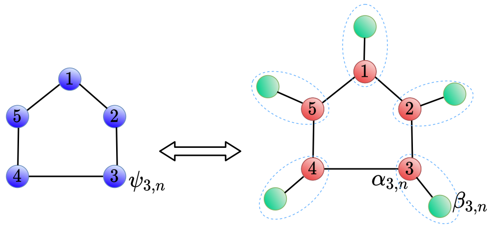


Fig. 1. A ring network topology with $N = 5$ nodes.

substates, so that the intermediate estimation error of agents in the decomposed network is expressed as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n - \boldsymbol{\alpha}_{i,n}(0) \qquad i = 1, \cdots, N\\ \boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n - \boldsymbol{\beta}_{i-N,n}(0) \quad i = N+1, \cdots, 2N\end{aligned}$$

Following the made assumption on the initial substates, $\boldsymbol{\alpha}_{i,n}(0) = \boldsymbol{\beta}_{i,n}(0) = \boldsymbol{\psi}_{i,n}$, the intermediate estimation error of each agent $i \in \{1, 2, \cdots, 2N\}$, employing the local observation in (2), is formulated as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \mathbf{x}_n - \boldsymbol{\psi}_{i,n}\\ =& \mathbf{x}_n - \hat{\mathbf{x}}_{i,n|n-1} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\left(\mathbf{y}_{i,n} - \mathbf{H}_i\hat{\mathbf{x}}_{i,n|n-1}\right)\\ =& \mathbf{x}_n - \hat{\mathbf{x}}_{i,n|n-1} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{H}_i\left(\mathbf{x}_n - \hat{\mathbf{x}}_{i,n|n-1}\right)\\ & - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{w}_{i,n}.\end{aligned} \tag{10}$$

Substituting (1) into (10) and after some algebraic manipulation, we have

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \left(\mathbf{I} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{H}_i\right)\mathbf{A}\boldsymbol{\epsilon}_{i,n-1|n-1}\\ &+ \left(\mathbf{I} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{H}_i\right)\mathbf{v}_n - \mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{w}_{i,n}.\end{aligned} \tag{11}$$

where $\boldsymbol{\epsilon}_{i,n-1|n-1} = \mathbf{x}_{n-1} - \hat{\mathbf{x}}_{i,n-1|n-1}$. Considering the block row vectors organizing all error terms as

$$\begin{aligned}\boldsymbol{\mathcal{E}}_n =& [\boldsymbol{\epsilon}_{1,n}^{\mathrm{T}}, \cdots, \boldsymbol{\epsilon}_{2N,n}^{\mathrm{T}}]^{\mathrm{T}}\\ \boldsymbol{\mathcal{E}}_{n-1|n-1} =& [\boldsymbol{\epsilon}_{1,n-1|n-1}^{\mathrm{T}}, \cdots, \boldsymbol{\epsilon}_{2N,n-1|n-1}^{\mathrm{T}}]^{\mathrm{T}}\end{aligned}$$

the network-wide state vector estimation error of the state-decomposed network, $\boldsymbol{\mathcal{E}}_{n|n}$, which is the stacked error after the privacy-preserving average consensus operations in (7) with k consensus iterations, is expressed by

$$\boldsymbol{\mathcal{E}}_{n|n} = \mathbf{G}^k\boldsymbol{\mathcal{E}}_n + \sum_{s=1}^{k}\mathbf{G}^{s-1}\boldsymbol{\mathcal{B}}\boldsymbol{\omega}(k-s). \tag{12}$$

The stacked perturbation sequences is denoted by $\boldsymbol{\omega}(k) = \left[\boldsymbol{\omega}_1^{\mathrm{T}}(k), \cdots, \boldsymbol{\omega}_N^{\mathrm{T}}(k)\right]^{\mathrm{T}}$, while $\boldsymbol{\mathcal{B}} = [\varepsilon\mathbf{W}, \mathbf{0}]^{\mathrm{T}} \otimes \mathbf{I}$, and $\mathbf{G}$ is a doubly stochastic matrix given by

$$\mathbf{G} = \begin{bmatrix}\mathbf{M} & \varepsilon\mathbf{U}\\ \varepsilon\mathbf{U} & \mathbf{I} - \varepsilon\mathbf{U}\end{bmatrix}$$

with $\mathbf{M} \triangleq (\mathbf{I} - \varepsilon(\mathbf{D} - \mathbf{W})) \otimes \mathbf{I} - \varepsilon\mathbf{U}$. The interaction and coupling weight matrices for the entire network are denoted by $\mathbf{W}(k) \triangleq [w_{ij}(k)]$ and $\mathbf{U}(k) = \mathsf{Blockdiag}(\{\mathbf{U}_i(k)\}_{i=1}^{N})$, respectively. To simplify the state vector error analysis, we assume that the interaction and coupling weight matrices are time-invariant. Alternatively, (12) can be expressed as

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} =& \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{E}}_{n-1|n-1} + \boldsymbol{\mathcal{Q}}\boldsymbol{\Upsilon}_n - \boldsymbol{\Omega}_n\\ &+ \sum_{s=1}^{k}\mathbf{G}^{s-1}\boldsymbol{\mathcal{B}}\boldsymbol{\omega}(k-s)\end{aligned} \tag{13}$$

where

$$\begin{aligned}\boldsymbol{\mathcal{P}} &= \mathbf{G}^k\mathsf{Blockdiag}(\{\mathbf{P}_i\mathbf{A}\}_{i=1}^{2N})\\ \boldsymbol{\mathcal{Q}} &= \mathbf{G}^k\mathsf{Blockdiag}(\{\mathbf{P}_i\}_{i=1}^{2N})\\ \boldsymbol{\Upsilon}_n &= [\mathbf{v}_n^{\mathrm{T}}, \mathbf{v}_n^{\mathrm{T}}, \cdots, \mathbf{v}_n^{\mathrm{T}}]^{\mathrm{T}}\\ \boldsymbol{\Omega}_n &= \mathbf{G}^k\mathsf{Blockdiag}(\{\mathbf{Q}_i\}_{i=1}^{2N})[\mathbf{w}_{1,n}^{\mathrm{T}}, \mathbf{w}_{2,n}^{\mathrm{T}}, \cdots, \mathbf{w}_{2N,n}^{\mathrm{T}}]^{\mathrm{T}}\end{aligned}$$

with $\mathbf{P}_i = \mathbf{I} - N\mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}\mathbf{H}_i$ and $\mathbf{Q}_i = \mathbf{M}_i\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_i}^{-1}$. Following the definition, $\mathbf{P}_i$ is stable and since $\mathbf{G}$ is doubly stochastic, the block matrix $\boldsymbol{\mathcal{P}}$ is stable; therefore, the statistical expectation of any vector norm for $\boldsymbol{\mathcal{E}}_{n|n}$ converges to a stabilizing value as $n \to \infty$. Taking the statistical expectation of (12) yields

$$\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\} = \boldsymbol{\mathcal{P}}\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n-1|n-1}\} = \boldsymbol{\mathcal{P}}^n\mathbb{E}\{\boldsymbol{\mathcal{E}}_{0|0}\}.$$

Once again, since $\boldsymbol{\mathcal{P}}$ is stable, we have $\lim_{n\to\infty}\mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\} = 0$ that indicates the steady-state estimates are unbiased regardless of their initializing values or perturbation sequences.

The recursive expression of the state vector estimation error in (13), is used to formulate the second-order statistics of all agents, denoted by $\boldsymbol{\Sigma}_n = \mathbb{E}\{\boldsymbol{\mathcal{E}}_{n|n}\boldsymbol{\mathcal{E}}_{n|n}^{\mathrm{T}}\}$, as

$$\boldsymbol{\Sigma}_n = \boldsymbol{\mathcal{P}}\boldsymbol{\Sigma}_{n-1}\boldsymbol{\mathcal{P}}^{\mathrm{T}} + \boldsymbol{\mathcal{Q}}\mathbf{C}_{\boldsymbol{\Upsilon}}\boldsymbol{\mathcal{Q}}^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{\Omega}} + \boldsymbol{\mathcal{T}} \tag{14}$$

where $\mathbf{C}_{\boldsymbol{\Upsilon}} = \mathbb{E}\{\boldsymbol{\Upsilon}_n\boldsymbol{\Upsilon}_n^{\mathrm{T}}\}$, $\mathbf{C}_{\boldsymbol{\Omega}} = \mathbb{E}\{\boldsymbol{\Omega}_n\boldsymbol{\Omega}_n^{\mathrm{T}}\}$, and with respect to the noise sequence (6), we have

$$\boldsymbol{\mathcal{T}} = \sum_{s=1}^{k}\sigma_{k-s}^2\mathbf{G}^{s-1}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{B}}^{\mathrm{T}}\left(\mathbf{G}^{s-1}\right)^{\mathrm{T}}.$$

Since $\mathbf{G}$ is doubly stochastic and $\boldsymbol{\mathcal{P}}$ is stable, $\boldsymbol{\Sigma}_n \to \boldsymbol{\Sigma}$ as $n \to \infty$, where $\boldsymbol{\Sigma}$ is the solution of the discrete-time Lyapunov equation in (14). The effect of injected noise is manifested in terms of $\boldsymbol{\mathcal{T}}$, which increases the steady-state mean squared error (MSE) of Algorithm 1 compared to the non-private approach. In the next section, we analyze the performance of the derived framework to preserve agent privacy.

V. Privacy Analysis

We consider an honest-but-curious agent that can access the interaction weights and information shared by its neighbors. To benchmark the privacy of the derived PP-DKF, we consider the MSE associated with the estimates of the initial states $\boldsymbol{\psi}_{i,n}(0)$ at the honest-but-curious agent, as privacy measure. Without loss of generality, it is assumed that the Nth agent is an honest-but-curious agent that employs a maximum likelihood (ML) estimator to estimate the initial states of all agents, $\boldsymbol{\psi}_n(0) = [\boldsymbol{\psi}_{1,n}^{\mathrm{T}}, \cdots, \boldsymbol{\psi}_{N,n}^{\mathrm{T}}]^{\mathrm{T}}$, at time instant n. The honest-but-curious agent has access to the following information set at consensus iteration k

$$\begin{aligned}\mathcal{I}(k) = \{&\boldsymbol{\alpha}_{N,n}(k), \boldsymbol{\beta}_{N,n}(k), \boldsymbol{\omega}_N(k), u_N(k),\\ & w_{Nj}(k), \tilde{\boldsymbol{\alpha}}_{j,n}(k) : \forall j \in \mathcal{N}_N\}.\end{aligned} \tag{15}$$

Proposition 1. *Suppose an honest-but-curious agent has access to messages shared by its neighbors and their corresponding interaction weights. If every agent has at least one regular agent in the neighborhood, an honest-but-curious agent cannot infer private information of any other agent in the network.*

Proof: The proof follows from Theorem 2 in [19] by showing that an arbitrary change in the initial information of the jth agent, $\boldsymbol{\psi}_{j,n}$ to $\bar{\boldsymbol{\psi}}_{j,n}$, remains indistinguishable from the honest-but-curious agent. ■

In the worst case, the honest-but-curious agent also accesses the interaction and coupling weights of the entire network, thereafter it can construct an ML estimator to estimate the private information of the other agents. To construct an ML estimator, we introduce the observation vector $\mathbf{y}_n(k)$ that includes the accessible information transferred from the neighbors to the honest-but-curious agent at each iteration k as

$$\mathbf{y}_n(k) = \mathbf{C}\mathbf{z}_n(k) + \mathbf{C}_\alpha \boldsymbol{\omega}(k)$$

where $\mathbf{C} \triangleq [\mathbf{C}_\alpha, \mathbf{C}_\beta]$ with $\mathbf{C}_\beta = [\mathbf{0}, \mathbf{e}_N]^{\mathrm{T}} \otimes \mathbf{I}$ and

$$\mathbf{C}_\alpha = \Big[\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \cdots, \mathbf{e}_{i_{N_N}}, \mathbf{e}_N\Big]^{\mathrm{T}} \otimes \mathbf{I}.$$

The canonical basis $\mathbf{e}_i$ is a vector with 1 in the ith entry and zeros elsewhere, while $\mathbf{z}_n(k) \triangleq [\boldsymbol{\alpha}_n^{\mathrm{T}}(k), \boldsymbol{\beta}_n^{\mathrm{T}}(k)]^{\mathrm{T}}$ with the network-wide agent substate vectors given as

$$\boldsymbol{\alpha}_n(k) \triangleq [\boldsymbol{\alpha}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\alpha}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}}$$
$$\boldsymbol{\beta}_n(k) \triangleq [\boldsymbol{\beta}_{1,n}^{\mathrm{T}}(k), \cdots, \boldsymbol{\beta}_{N,n}^{\mathrm{T}}(k)]^{\mathrm{T}}.$$

The estimated value of $\mathbf{z}_n(0) \triangleq [\boldsymbol{\alpha}_n^{\mathrm{T}}(0), \boldsymbol{\beta}_n^{\mathrm{T}}(0)]^{\mathrm{T}}$ is employed to estimate agent initial states as $\hat{\boldsymbol{\psi}}_n(0) = \frac{1}{2}(\hat{\boldsymbol{\alpha}}_n(0) + \hat{\boldsymbol{\beta}}_n(0))$.

Since the information of the Nth agent is already known to the honest-but-curious agent, we reduce the state space dimension by removing all entries belonging to the Nth agent form the defined variables and find the estimation error covariance $\tilde{\mathbf{P}}(k)$ instead of $\mathbf{P}(k)$ as it satisfies

$$\mathbf{P}(k) = \begin{bmatrix} \tilde{\mathbf{P}}(k) & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix}.$$

Accordingly, the reduced version of $\mathbf{C}$ and the observation vector $\mathbf{y}_n(k)$ can be expressed as $\tilde{\mathbf{C}} = [\tilde{\mathbf{C}}_\alpha, \tilde{\mathbf{0}}]$ and

$$\tilde{\mathbf{y}}_n(k) = \tilde{\mathbf{C}}\tilde{\mathbf{z}}_n(k) + \tilde{\mathbf{C}}_\alpha \tilde{\boldsymbol{\omega}}(k) \tag{16}$$

where

$$\tilde{\mathbf{z}}_n(k) = [\tilde{\boldsymbol{\alpha}}_n^{\mathrm{T}}(k), \tilde{\boldsymbol{\beta}}_n^{\mathrm{T}}(k)]^{\mathrm{T}}$$
$$\tilde{\mathbf{C}}_\alpha = [\tilde{\mathbf{e}}_{j_1}, \tilde{\mathbf{e}}_{j_2}, \cdots, \tilde{\mathbf{e}}_{j_{N_N}}]^{\mathrm{T}}.$$

Substituting the network-wide state update equations (7) in (16), gives

$$\tilde{\mathbf{y}}_n(k) = \tilde{\mathbf{C}}\tilde{\mathbf{G}}^k \tilde{\mathbf{z}}_n(0) + \tilde{\mathbf{C}}_\alpha \left(\sum_{t=0}^{k-1} \boldsymbol{\mathcal{C}}_{k-1-t} \tilde{\boldsymbol{\mathcal{B}}} \tilde{\boldsymbol{\omega}}(t) + \tilde{\boldsymbol{\omega}}(k) \right) \tag{17}$$

where $\tilde{\boldsymbol{\mathcal{B}}} = \varepsilon \tilde{\mathbf{W}} \otimes \mathbf{I}$, $\boldsymbol{\mathcal{C}}_k = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \tilde{\mathbf{G}}^k \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}^{\mathrm{T}}$, and

$$\tilde{\mathbf{G}} = \begin{bmatrix} \tilde{\mathbf{M}} & \varepsilon\tilde{\mathbf{U}} \\ \varepsilon\tilde{\mathbf{U}} & \mathbf{I} - \varepsilon\tilde{\mathbf{U}} \end{bmatrix}.$$

We can simplify the accumulated observation set of the honest-but-curious agent, up to consensus iteration k, as

$$\begin{bmatrix} \tilde{\mathbf{y}}_n(0) \\ \tilde{\mathbf{y}}_n(1) \\ \vdots \\ \tilde{\mathbf{y}}_n(k) \end{bmatrix} = \mathbf{H}(k)\tilde{\mathbf{z}}_n(0) + \mathbf{F}(k) \begin{bmatrix} \tilde{\boldsymbol{\omega}}(0) \\ \tilde{\boldsymbol{\omega}}(1) \\ \vdots \\ \tilde{\boldsymbol{\omega}}(k) \end{bmatrix} \tag{18}$$

where $\mathbf{H}(k) \triangleq [(\tilde{\mathbf{C}})^{\mathrm{T}}, (\tilde{\mathbf{C}}\tilde{\mathbf{G}})^{\mathrm{T}}, \cdots, (\tilde{\mathbf{C}}\tilde{\mathbf{G}}^k)^{\mathrm{T}}]^{\mathrm{T}}$ and

$$\mathbf{F}(k) = \begin{bmatrix} \tilde{\mathbf{C}}_\alpha & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{\mathbf{C}}_\alpha \boldsymbol{\mathcal{C}}_0 \tilde{\boldsymbol{\mathcal{B}}} & \tilde{\mathbf{C}}_\alpha & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}_\alpha \boldsymbol{\mathcal{C}}_{k-1} \tilde{\boldsymbol{\mathcal{B}}} & \tilde{\mathbf{C}}_\alpha \boldsymbol{\mathcal{C}}_{k-2} \tilde{\boldsymbol{\mathcal{B}}} & \cdots & \tilde{\mathbf{C}}_\alpha \end{bmatrix}. \tag{19}$$

Subsequently, the error covariance of the ML estimator [26], to estimate $\tilde{\mathbf{z}}_n(0)$, with independent noise sequences is obtained by

$$\tilde{\mathbf{P}}(k) = \left(\mathbf{H}^{\mathrm{T}}(k) \left(\mathbf{F}(k) \tilde{\boldsymbol{\Gamma}}(k) \mathbf{F}^{\mathrm{T}}(k) \right)^{-1} \mathbf{H}(k) \right)^{-1} \tag{20}$$

where $\tilde{\boldsymbol{\Gamma}}(k) = \mathsf{diag}\left(\{\sigma_t^2 \mathbf{I}\}_{t=0}^{k} \right)$ contains the perturbation sequence covariances up to consensus iteration k. Since the accessible information of the honest-but-curious agent is expanding, the error covariance $\tilde{\mathbf{P}}(k)$ is monotonically non-increasing, i.e., for $k_1 \leq k_2$, we have $\tilde{\mathbf{P}}(k_2) \leq \tilde{\mathbf{P}}(k_1)$. This implies that error covariance matrix $\tilde{\mathbf{P}}(k)$ converges to a constant matrix $\tilde{\mathbf{P}} = \lim_{k \to \infty} \tilde{\mathbf{P}}(k)$. Let us assume

$$\tilde{\mathbf{P}} = \begin{bmatrix} \tilde{\mathbf{P}}_1 & \tilde{\mathbf{P}}_{12} \\ \tilde{\mathbf{P}}_{21} & \tilde{\mathbf{P}}_{22} \end{bmatrix},$$

then, the error covariance of the ML estimator to estimate $\boldsymbol{\psi}_n(0)$ is given by

$$\bar{\mathbf{P}} = \frac{1}{4} \left(\tilde{\mathbf{P}}_1 + \tilde{\mathbf{P}}_{12} + \tilde{\mathbf{P}}_{21} + \tilde{\mathbf{P}}_{22} \right).$$

Thus, the privacy metric of the ith agent, related to estimate its initial state $\boldsymbol{\psi}_{i,n}(0)$ by the honest-but-curious agent N is defined as

$$\mathcal{E}_i \triangleq \mathsf{tr}\left((\tilde{\mathbf{e}}_i^{\mathrm{T}} \otimes \mathbf{I}) \bar{\mathbf{P}} (\tilde{\mathbf{e}}_i \otimes \mathbf{I}) \right). \tag{21}$$

The derived privacy metric represents the ability of the privacy-preserving strategy to conceal the initial states from being estimated by the honest-but-curious agent. Several simulations verify the privacy performance of the proposed PP-DKF in the next section.

VI. Numerical Results

We consider a ring network topology with $N = 5$ agents shown in Fig. 1. The proposed PP-DKF is considered in a collaborative target tracking application. The state-space model is following the distributed Kalman filter in [6], where the state vector $\mathbf{x}_n = [X_n, Y_n, \dot{X}_n, \dot{Y}_n]^{\mathrm{T}}$ consists of the positions $\{X_n, Y_n\}$ and velocities $\{\dot{X}_n, \dot{Y}_n\}$ in the horizontal and vertical directions, respectively. For comparison purposes, we implement a pure noise-injection based privacy-preserving DKF (NIP-DKF), wherein the noise sequence in (6) perturbs

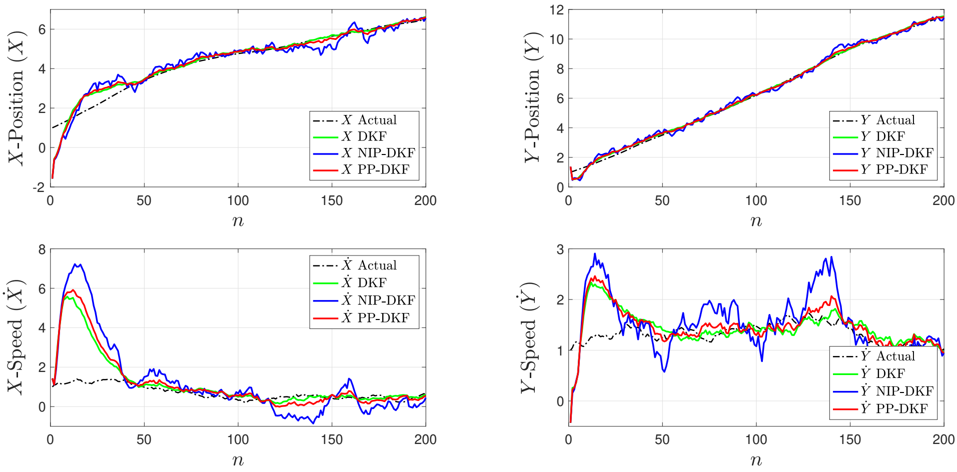


Fig. 2. Tracking performance of the derived PP-DKF with $K = 40$ consensus iterations and noise variance $\sigma^2 = 0.5$.

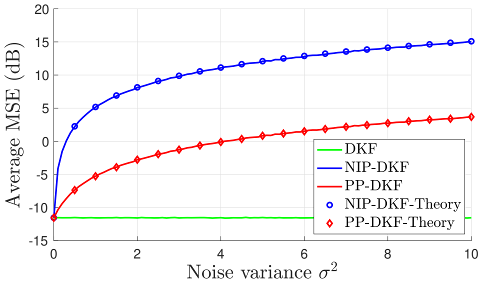


Fig. 3. Average filtering MSE versus injected noise variance σ^2.

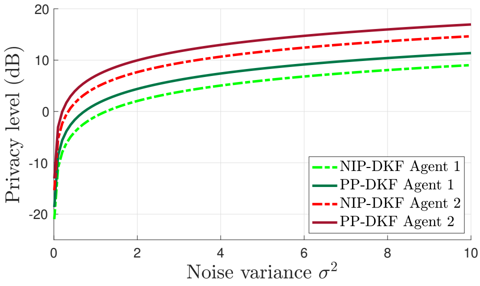


Fig. 4. Privacy metric $\mathcal{E}_i$ versus injected noise variance σ^2.

the shared messages of the conventional DKF [6]. Regarding the perturbation sequence assumptions in (6), we assume $\sigma_k^2 = \frac{\phi^{2k}}{N(k+1)}\sigma^2$ at each consensus iteration, where $\phi = 0.9$, and σ^2 is noise variance that controls the amount of the injected noise.

Fig. 2 shows the performance of the proposed PP-DKF to track the system state compared to the NIP-DKF and non-private distributed Kalman filter (DKF). The proposed PP-DKF performs as well as the non-private distributed Kalman filter and outperforms the NIP-DKF. This means that the estimate produced by PP-DKF is closer to the actual position and speed of the target compared to NIP-DKF. The higher accuracy of PP-DKF to track the position and speed of the target, verifying its robustness to the perturbation noise sequences.

Fig. 3 shows the average MSE of the distributed Kalman filter versus the noise variance parameter σ^2 with $K = 40$ consensus iterations. We see that the perturbation sequence degrades the performance of the privacy-preserving approaches, PP-DKF and NIP-DKF, compared to the conventional DKF [6]. We also see that the proposed PP-DKF significantly outperforms the NIP-DKF method by achieving lower MSE for a broad range of injected noise variances, indicating lower sensitivity of the PP-DKF to the noise variance than the NIP-DKF. This is because the proposed PP-DKF operates by partially obfuscating shared substates. At the same time, the NIP-DKF solution perturbs the entire state before sharing among neighbors, which was the motivation behind the design of our consensus framework.

Fig. 4 shows the privacy metric (21) for $K = 30$ consensus iterations versus the noise variance parameter σ^2, for all agents. We see that injecting a higher amount of noise results in higher privacy, where the privacy level of all agents is significantly improved under the proposed PP-DKF compared to the NIP-DKF. Due to the ring topology, agents 3 and 4 achieve the same privacy level as agents 2 and 1. The improved privacy-accuracy trade-off under the PP-DKF is manifested by achieving lower MSE and higher privacy $\mathcal{E}_i$ for all agents compared to NIP-DKF.

VII. Conclusion

This paper proposed a privacy-preserving distributed Kalman filter that utilizes both decomposition-based and noise injection-based privacy-preserving average consensus techniques to protect network agents disclosing their private information. It provides a private distributed Kalman filter by restricting the amount of information exchanged with decomposition and concealing the private data from being estimated by adversaries with perturbation. The convergence and performance of the derived PP-DKF have been analyzed. The achieved privacy level of all agents, defined as the uncertainty of the honest-but-curious agent to estimate the initial state of other agents, has been characterized in the presence of an honest-but-curious agent. It has been shown that the proposed PP-DKF solution improves privacy and performance of the Kalman filtering operations compared to the DKF employing contemporary privacy-preserving techniques. Lastly, several simulations verified the obtained theoretical results.

References

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, Jun. 2018.

[2] V. Katewa, F. Pasqualetti, and V. Gupta, "On privacy vs. cooperation in multi-agent systems," *Int. J. of Control*, vol. 91, no. 7, pp. 1693–1707, Jul. 2018.

[3] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decis. and Control*, 2007, pp. 5492–5498.

[4] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sept. 2010.

[5] U. A. Khan and J. M. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[6] S. P. Talebi and S. Werner, "Distributed Kalman filtering and control through embedded average consensus information fusion," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4396–4403, Oct. 2019.

[7] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decis. and Control*, 2005, pp. 8179–8184.

[8] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. 48th IEEE Conf. Decis. and Control*, 2009, pp. 7036–7042.

[15] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[9] J. Le Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.

[10] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed privacy-preserving data aggregation against dishonest nodes in network systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1462–1470, Apr. 2019.

[11] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *Proc. 45th IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2020, pp. 5895–5899.

[12] A. Moradi, N. K. Venkategowda, and S. Werner, "Coordinated data-falsification attacks in consensus-based distributed Kalman filtering," in *Proc. 8th IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, 2019, pp. 495–499.

[13] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, Jul. 2017.

[14] J. He, L. Cai, and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 4069–4082, Jul. 2020.

[16] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 127–138, Mar. 2019.

[17] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5677–5690, Aug. 2018.

[18] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4035–4049, Oct. 2019.

[19] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, Nov. 2019.

[20] W. Wang, D. Li, X. Wu, and S. Xue, "Average consensus for switching topology networks with privacy protection," in *Proc. IEEE Chinese Automat. Congr.*, 2019, pp. 1098–1102.

[21] J. Le Ny, "Differentially private Kalman filtering," in *Differential Privacy for Dynamic Data*. Springer, 2020, pp. 55–75.

[22] K. H. Degue and J. Le Ny, "On differentially private Kalman filtering," in *Proc. 5th IEEE Global Conf. Signal and Inf. Process.*, 2017, pp. 487–491.

[23] Y. Song, C. X. Wang, and W. P. Tay, "Privacy-aware kalman filtering," in *Proc. 43rd IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2018, pp. 4434–4438.

[24] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Comput. Surveys*, vol. 51, no. 3, pp. 1–38, Jun. 2018.

[25] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, Jan. 2007.

[26] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Prentice Hall Signal Process. Ser. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

Appendix C

Publications in Chapter 5

P7: V. C. Gogineni, A. Moradi, N. K. D. Venkategowda and S. Werner, "Communication-Efficient and Privacy-Aware Distributed LMS Algorithm," in Proceedings 25th *IEEE International Conference on Information Fusion*, 2022, pp. 1-6.

P8: V. C. Gogineni, A. Moradi, N. K. D. Venkategowda and S. Werner, "Communication-Efficient and Privacy-Aware Distributed Learning," submitted to *IEEE Transactions on Signal and Information Processing over Networks*, pp. 1-13, 2023.

P8 is awaiting publication and is not included
P7 is available at https://doi.org/10.23919/FUSION49751.2022.9841380

Communication-Efficient and Privacy-Aware Distributed LMS Algorithm

Vinay Chakravarthi Gogineni[⋆], Ashkan Moradi[⋆], Naveen K. D. Venkategowda[§], Stefan Werner[⋆]
[⋆]Dept. of Electronic Systems, Norwegian University of Science and Technology-NTNU, Norway
[§]Dept. of Science and Technology, Linköping University, Sweden
E-mails: {vinay.gogineni, ashkan.moradi, stefan.werner}@ntnu.no, naveen.venkategowda@liu.se

***Abstract*—This paper presents a private-partial distributed least mean square (PP-DLMS) algorithm that offers energy efficiency while preserving privacy and is suitable for applications with limited resources and strict security requirements. The proposed PP-DLMS allows every agent to exchange only a fraction of their perturbed data with neighbors during the collaboration process to minimize communication costs and guarantee privacy simultaneously. In order to understand how partial-sharing of perturbed data affects the learning performance, we conduct mean convergence analysis. Moreover, to investigate the privacy-preserving properties of the proposed algorithm, we characterize agent privacy in the presence of an honest-but-curious (HBC) adversary. Analytical results show that the proposed PP-DLMS is resilient against an HBC adversary by providing a fair energy-privacy trade-off compared to the conventional LMS algorithm. Numerical simulations corroborate the analytical findings.**

***Index Terms*—Distributed learning, energy-efficiency, privacy-preservation, average consensus, multiagent systems.**

I. Introduction

In the past decade, distributed computing systems have played a significant role in advancing signal processing and machine learning over multiagent networks [1]–[5]. The distributed network structure facilitates local communication between agents and their neighbors, thus enhancing the learning performance and robustness against dynamic changes in network topology. The local interactions among agents are realized via radio communication, which consumes large amounts of power and bandwidth. Local interactions are not only energy-intensive but also vulnerable to potential adversaries [6]. Thus, a distributed learning procedure that reduces the communication load as much as possible without significantly impairing the privacy of agents and overall estimation performance is always preferred.

Cryptography-based methods can provide secure communication between agents. However, they add substantial communication overhead and require considerable amounts of power [7]–[9], prohibiting their use in resource-constrained networks. Furthermore, cryptographic techniques are ineffective against privacy theft by dishonest network agents. Instead, low-complexity methods like noise injection-based mechanisms are attractive alternatives for preserving the privacy of individual agents [10]–[17]. In this category, differential-privacy techniques inject uncorrelated noise sequences into the information exchanged to ensure data privacy [10], [11]. The privacy-accuracy trade-off was improved in [15]–[18] by injecting correlated noise sequences with decaying variances into the exchanged information. Meanwhile, decomposition-based privacy-preserving techniques divide the private information into two substates, of which only one is shared among agents, hence making inference more difficult for adversaries [19], [20].

Distributed computing systems are often associated with limited computational and power resources, so resource-intensive local interactions should be minimized. This can be accomplished by performing dimensionality reduction [21] and 1-bit quantization [22] on the information before exchanging. Although these methods reduce communication costs, they are time-consuming and add additional computational burden to agents. Employing a probabilistic communication strategy is also an alternative solution to reduce local communication among agents [23]. Furthermore, partial-sharing concepts proposed in [27]–[29] reduce the consumption of resources by allowing agents to share only a fraction of information during each inter-agent interaction. The ease of implementation has made partial-sharing concepts popular in distributed learning. These communication-efficient methods, however, have not been investigated for privacy protection. To this end, in this paper, we propose a distributed learning framework that simultaneously attains both energy efficiency and privacy preservation.

This paper presents a private-partial distributed LMS (PP-DLMS) algorithm that enables agents to participate in local interactions by sharing only a fraction of their perturbed information, thus reducing resource consumption as well as preserving privacy. To investigate the impact of partial-sharing of perturbed data on the performance of distributed learning, we analyze the mean convergence and study the privacy of agents in the presence of an honest-but-curious (HBC) adversary. The HBC agent is a legitimate agent in the network that is curious about the private information of other agents. Since an HBC agent is a member of the network, it has access to the information exchanged in the neighborhood as well as to the information of the partial-sharing-based communication mechanism. As a result, the network becomes more vulnerable to information leakage. The privacy analysis shows that the proposed PP-DLMS provides a fair energy-privacy trade-off against HBC adversaries. Finally, we provide

The first two authors contributed equally to this work.

This work was supported by the Research Council of Norway.

numerical simulations that corroborate our analytical findings.

Mathematical notation: Scalars are denoted by lowercase letters, column vectors by bold lowercase, and matrices by bold uppercase. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-1}$ denote the transpose and inverse operators, respectively. The symbol $\mathbf{1}_K$ represents the $K \times 1$ column vector with all entries equal to one and $\mathbf{I}_K$ is the $K \times K$ identity matrix. The right Kronecker product of two matrices is denoted by $\otimes$, while $\lambda_i(\mathbf{A})$ denotes the ith eigenvalue of matrix $\mathbf{A}$.

II. Background and Problem Formulation

Consider a sensor network modeled as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ represents the agents of the network and $\mathcal{E}$ is the set of edges that represent bidirectional communication links between the nodes, i.e., $(k,l) \in \mathcal{E}$ if nodes k and l are connected. Additionally, the set $\mathcal{N}_k$ indicates the neighborhood of the node k that includes itself and the cardinality of the set $\mathcal{N}_k$ is denoted by $|\mathcal{N}_k|$, while $K = |\mathcal{N}|$ is the number of agents in the network. At time instant n and agent k, the input signal $\mathbf{x}_{k,n}$ and desired signal $y_{k,n}$ are assumed to be described as

$$y_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}}\, \mathbf{w}^{\star} + \epsilon_{k,n}, \tag{1}$$

where $\mathbf{w}^{*} \in \mathbb{R}^L$ is an optimal parameter vector to be estimated, $\mathbf{x}_{k,n} = [x_{k,n}, x_{k,n-1}, \ldots, x_{k,n-L+1}]^{\mathrm{T}}$ is the input signal vector, and the observation noise $\epsilon_{k,n}$ is a zero-mean Gaussian random sequence. The estimate of $\mathbf{w}^{\star}$ at time instant n, i.e., $\mathbf{w}_n$ is chosen so that it minimizes

$$\mathcal{J}_n = \frac{1}{K} \sum_{k \in \mathcal{N}} \mathbb{E}[e_{k,n}^2], \tag{2}$$

where $e_{k,n} = y_{k,n} - \hat{y}_{k,n}$ with $\hat{y}_{k,n}$ as the estimated filter output at agent k. At every time instant n, $\mathbf{w}_n$ can be updated via steepest-descent approach as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\eta}{2} \nabla \mathcal{J}_n = \mathbf{w}_n + \eta \sum_{k \in \mathcal{N}} e_{k,n} \mathbf{x}_{k,n}, \tag{3}$$

where η is the step size. The operation in (3) can be modeled as $\mathbf{w}_{n+1} = \frac{1}{K} \sum\limits_{k \in \mathcal{N}} \boldsymbol{\psi}_{k,n+1}$ with

$$\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_n + \mu\, e_{k,n}\, \mathbf{x}_{k,n}, \tag{4}$$

being the intermediate estimate of $\mathbf{w}^{\star}$ at node k and time instant n, and $\mu = \eta K$ is the new step size. The average of the intermediate estimate $\boldsymbol{\psi}_{k,n+1}$ across the entire network can be evaluated in a distributed manner using an average consensus filter (ACF) [24]–[26].

In the process of obtaining an average consensus, agents exchange local information $\boldsymbol{\psi}_{k,n+1}$ with their neighbors that contains node-sensitive information and might be exploited by potential adversaries. To protect the node-sensitive information from being inferred by adversaries, agents exchange perturbed versions of their private information [15]–[17]. Thus, the state of the ACF after m consensus iterations is

$$\mathbf{h}_{k,(m)} = \sum_{l \in \mathcal{N}_k} a_{lk} \tilde{\mathbf{h}}_{l,(m-1)}, \tag{5}$$

where a_{lk} is the consensus weight between agents l and k, $\tilde{\mathbf{h}}_{l,(m-1)} = \mathbf{h}_{l,(m-1)} + \boldsymbol{\omega}_{l,(m-1)}$ is the perturbed local information with $\mathbf{h}_{l,(0)} = \boldsymbol{\psi}_{l,n+1}$, and $\boldsymbol{\omega}_{l,(m-1)}$ is the perturbation noise at agent l and $(m-1)$th consensus iteration [15]. The perturbation noise at agent l and consensus iteration m is given by

$$\boldsymbol{\omega}_{l,(m)} = \begin{cases} \boldsymbol{\nu}_{l,(0)}, & m = 0 \\ \phi^m \boldsymbol{\nu}_{l,(m)} - \phi^{m-1} \boldsymbol{\nu}_{l,(m-1)}, & \text{otherwise,} \end{cases} \tag{6}$$

where constant $\phi \in (0,1)$ is same for all agents, and $\boldsymbol{\nu}_{l,(m)} \in \mathbb{R}^L$ is a zero-mean Gaussian sequence with $\mathbb{E}[\boldsymbol{\nu}_{l,(m)} \boldsymbol{\nu}_{l,(m)}^{\mathrm{T}}] = \sigma_\nu^2 \mathbf{I}_L$. If $\mathbf{A}$ with $[\mathbf{A}]_{l,k} = a_{lk}$ is a doubly stochastic matrix that satisfies the conditions stated in [25] and the perturbation noise follows (6), all agents reach consensus on the exact average, given by

$$\lim_{m \to \infty} \mathbf{h}_{k,(m)} = \frac{1}{K} \sum_{l \in \mathcal{N}} \mathbf{h}_{l,(0)}, \tag{7}$$

asymptotically.

III. PP-DLMS Algorithm

As shown in (5), the collaboration between agents is vital for distributed learning. Privacy-preserving distributed learning techniques are no exception. However, although collaboration among agents improves learning accuracy, it is resource-intensive. As nodes in sensor networks have limited battery power, reducing the inter-node communication overhead is essential while maintaining inter-node cooperation benefits. By promoting partial-sharing [27]–[29] among agents in privacy-preserving distributed learning systems, we aim to achieve both privacy and energy efficiency in a single framework.

In the proposed PP-DLMS, during each consensus iteration m, every agent shares only a portion of the perturbed version of its private information with neighbors (i.e., M out of L entries in $\mathbf{h}_{k,(m)}$) to reduce the communication load while maintaining privacy. The entry selection procedure at each agent k is characterized by a diagonal selection matrix of size $L \times L$, the main diagonal of which consists of M numbers of ones and $L - M$ numbers of zeros. The selection matrix of agent k at time instant n and consensus iteration m is denoted by $\mathbf{S}_{k,n,(m)}$, where the position of ones indicates which entries of the private information are to be shared with neighbors. The selection of M out of L entries can be made stochastically, or, sequentially as in [27], [28]. We adopt a coordinated partial-sharing scheme, which is a special case of sequential and stochastic partial-sharing methods [28]. In coordinated partial-sharing, all agents are initialized with the same selection matrices, i.e., $\mathbf{S}_{1,0,(0)} = \mathbf{S}_{2,0,(0)} \cdots \mathbf{S}_{K,0,(0)}$. Since we are using the coordinated partial-sharing, we drop node index in $\mathbf{S}_{k,n,(m)}$ and continue with $\mathbf{S}_{n,(m)}$. Additionally, the selection matrix at the current consensus iteration, i.e., $\mathbf{S}_{n,(m)}$, can be obtained by applying a right-circular shift operation on the main diagonal elements of the selection matrix during the previous consensus iteration, i.e., $\mathbf{S}_{n,(m-1)}$. We also consider $\mathbf{S}_{n,(0)} = \mathbf{S}_{n-1,(m)}$ at each time index n. This process has an entry-sharing probability of $p = \frac{M}{L}$ because each entry will be

Algorithm 1: Private-Partial DLMS (PP-DLMS)

```
• For each agent $k \in \mathcal{N}$
Initialize: $\mathbf{S}_{n,(0)}, \tau$,
$\hat{y}_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}} \mathbf{w}_{k,n}$
$e_{k,n} = y_{k,n} - \hat{y}_{k,n}$
Local Update:
        $\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_{k,n} + \mu\, \mathbf{x}_{k,n}\, e_{k,n}$
Average Consensus Update:
Set $\mathbf{h}_{k,(0)} = \boldsymbol{\psi}_{k,n+1}$
For $m = 1$ to $T$
Perturb the local data $\tilde{\mathbf{h}}_{k,(m-1)} = \mathbf{h}_{k,(m-1)} + \boldsymbol{\omega}_{k,(m-1)}$
Share $\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{k,(m-1)}$
Receive $\left\{\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} : \forall l \in \mathcal{N}_k^-\right\}$
 $\mathbf{h}_{k,(m)} = a_{kk} \tilde{\mathbf{h}}_{k,(m-1)}$
    $+\sum_{l \in \mathcal{N}_k^-} a_{lk} \left(\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{k,(m-1)}\right)$
$\mathbf{S}_{n,(m)} = \text{circularshift}\left(\mathbf{S}_{n,(m-1)}, \tau\right)$
Endfor
$\mathbf{w}_{k,n+1} = \mathbf{h}_{k,(T)}$
```

shared M times during L subsequent iterations. By using the selection matrices, the privacy-preserving average consensus state update at each agent k can be expressed alternatively as

$$\begin{aligned}\mathbf{h}_{k,(m)} &= a_{kk} \tilde{\mathbf{h}}_{k,(m-1)} \\ &+\sum_{l \in \mathcal{N}_k^-} a_{lk} \big(\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{l,(m-1)}\big),\end{aligned} \tag{8}$$

where $\mathcal{N}_k^-$ indicates the neighborhood of node k excluding itself. As a result of partial information sharing, agents do not have access to the portion of the information that was not shared. However, by allowing each node to use its own internal information instead of the unshared information of neighboring agents, this challenge can be solved. At each agent k, we therefore substitute $(\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{k,(m-1)}$ in the place of $(\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{l,(m-1)}$ for each $l \in \mathcal{N}_k^-$ as

$$\begin{aligned}\mathbf{h}_{k,(m)} &= a_{kk} \tilde{\mathbf{h}}_{k,(m-1)} \\ &+\sum_{l \in \mathcal{N}_k^-} a_{lk} \left(\mathbf{S}_{n,(m-1)} \tilde{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) \tilde{\mathbf{h}}_{k,(m-1)}\right).\end{aligned} \tag{9}$$

After a sufficient number of consensus iterations, say T, the parameter vector $\mathbf{w}_{k,n}$ is updated to $\mathbf{w}_{k,n+1} = \mathbf{h}_{k,(T)}$. The workflow of the proposed PP-DLMS is summarized in Algorithm 1.

IV. Performance Analysis

In this section, we examine the impact of partial sharing of information on convergence and privacy.

A. Network Global Model

At each time instant n, we define the optimal model parameter vector $\mathbf{w}_{net}^\star = \mathbf{1}_K \otimes \mathbf{w}^\star$, estimated model parameter vector $\mathbf{w}_{net,n} = \mathrm{col}\{\mathbf{w}_{1,n}, \mathbf{w}_{2,n}, \ldots, \mathbf{w}_{K,n}\}$, input data matrix $\mathbf{X}_n = \mathrm{blockdiag}\{\mathbf{x}_{1,n}, \mathbf{x}_{2,n}, \ldots, \mathbf{x}_{K,n}\}$, observation noise vector $\boldsymbol{\epsilon}_{net,n} = \mathrm{col}\big\{\epsilon_{1,n}, \epsilon_{2,n}, \ldots, \epsilon_{K,n}\big\}$, and private information

$$\begin{aligned}\mathbf{h}_{(0)} &= \mathrm{col}\{\mathbf{h}_{1,(0)}, \mathbf{h}_{2,(0)}, \ldots, \mathbf{h}_{K,(0)}\} \\ &= \mathrm{col}\{\boldsymbol{\psi}_{1,n}, \boldsymbol{\psi}_{2,n}, \ldots, \boldsymbol{\psi}_{K,n}\},\end{aligned} \tag{10}$$

where the column-wise stacking and block diagonalization operations are represented by $\mathrm{col}\{\cdot\}$ and $\mathrm{blockdiag}\{\cdot\}$, respectively. Using the above definitions, data model and error vector at network-level are

$$\begin{aligned}\mathbf{y}_n &= \mathrm{col}\{y_{1,n}, y_{2,n}, \ldots, y_{K,n}\} = \mathbf{X}_n^{\mathrm{T}} \mathbf{w}_{net}^\star + \boldsymbol{\epsilon}_n \\ \mathbf{e}_n &= \mathrm{col}\big\{e_{1,n}, e_{2,n}, \ldots, e_{K,n}\big\} = \mathbf{y}_n - \mathbf{X}_n^{\mathrm{T}} \mathbf{w}_{net,n}.\end{aligned} \tag{11}$$

According to definitions in (11), the average consensus state update in (9), and

$$\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_{k,n} + \mu\, \mathbf{x}_{k,n}\, e_{k,n}, \tag{12}$$

the network-level model of the PP-DLMS can be stated as

$$\mathbf{w}_{net,n+1} = \boldsymbol{\mathcal{B}}_n \big(\mathbf{w}_{net,n} + \mu \mathbf{X}_n\, \mathbf{e}_n\big) + \mathbf{c}_n \tag{13}$$

with

$$\boldsymbol{\mathcal{B}}_n = \prod_{i=0}^{m-1} \boldsymbol{\mathcal{B}}_{n,(i)} \text{ and } \mathbf{c}_n = \sum_{i=0}^{m-1} \Big(\prod_{j=i}^{m-1} \boldsymbol{\mathcal{B}}_{n,(j)}\Big) \boldsymbol{\omega}_{(i)}, \tag{14}$$

where $\boldsymbol{\mathcal{B}}_{n,(m)} = \mathbf{A} \otimes \mathbf{S}_{n,(m)} + \mathbf{I}_K \otimes (\mathbf{I}_L - \mathbf{S}_{n,(m)})$, $\boldsymbol{\omega}_{(i)} = \mathrm{col}\{\boldsymbol{\omega}_{1,(i)}, \boldsymbol{\omega}_{2,(i)}, \ldots, \boldsymbol{\omega}_{K,(i)}\}$, and the network-level perturbation noise vector is given by

$$\boldsymbol{\omega}_{(i)} = \begin{cases} \boldsymbol{\nu}_{(0)}, & i = 0 \\ \phi^i \boldsymbol{\nu}_{(i)} - \phi^{i-1} \boldsymbol{\nu}_{(i-1)}, & \text{otherwise}, \end{cases} \tag{15}$$

where $\boldsymbol{\nu}_{(i)} = \mathrm{col}\{\boldsymbol{\nu}_{1,(i)}, \boldsymbol{\nu}_{2,(i)}, \ldots, \boldsymbol{\nu}_{K,(i)}\}$. In order to obtain the convergence condition for PP-DLMS, we assume the following:

- **A1.** For all $k \in \mathcal{N}$, the input signal vector $\mathbf{x}_{k,n}$ is drawn from a WSS multivariate random sequence with correlation matrix $\mathbf{R}_k = \mathrm{E}[\mathbf{x}_{k,n} \mathbf{x}_{k,n}^{\mathrm{T}}]$; in addition, the input signal vectors $\mathbf{x}_{k,n}$ and $\mathbf{x}_{l,m}$ are independent for all $k \neq l$ and $n \neq m$.
- **A2.** The noise process $\epsilon_{k,n}$ is assumed to be zero-mean i.i.d. and independent of any other quantity.
- **A3.** For all $k \in \mathcal{N}$, the selection matrix $\mathbf{S}_{n,(m)}$ is assumed to be independent of any other data.

B. First-order Convergence

Considering $\tilde{\mathbf{w}}_{net,n} = \mathbf{w}_{net}^\star - \mathbf{w}_{net,n}$, and using the fact that $\mathbf{w}_{net}^\star = \boldsymbol{\mathcal{B}}_n \mathbf{w}_{net}^\star$ (since $\boldsymbol{\mathcal{B}}_{n,(m)} \mathbf{w}_{net}^\star = \mathbf{w}_{net}^\star$ for all m), then form (13), $\tilde{\mathbf{w}}_{net,n+1}$ can be recursively expressed as

$$\tilde{\mathbf{w}}_{net,n+1} = \boldsymbol{\mathcal{B}}_n \big(\mathbf{I}_{LK} - \mu \mathbf{X}_n \mathbf{X}_n^{\mathrm{T}}\big) \tilde{\mathbf{w}}_{net,n} - \mu \boldsymbol{\mathcal{B}}_n \mathbf{X}_n \boldsymbol{\epsilon}_{net,n} - \mathbf{c}_n. \tag{16}$$

Applying expectation $\mathbb{E}[\cdot]$ on the both sides of (16) and using the assumptions $\mathbf{A1}-\mathbf{A3}$, we obtain

$$\mathbb{E}[\tilde{\mathbf{w}}_{net,n+1}] = \mathbb{E}[\boldsymbol{\mathcal{B}}_n]\big(\mathbf{I}_{LK}-\mu\boldsymbol{\mathcal{R}}\big)\mathbb{E}[\tilde{\mathbf{w}}_{net,n}], \qquad (17)$$

where $\boldsymbol{\mathcal{R}} = \mathbb{E}[\mathbf{X}_n\mathbf{X}_n^{\mathrm{T}}] = \text{blockdiag}\{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K\}$. From (17), one can see that $\lim_{n\to\infty}\mathbb{E}[\tilde{\mathbf{w}}_{net,n}]$ attains finite value if and only if $\|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\big(\mathbf{I}_{LK}-\mu\boldsymbol{\mathcal{R}}\big)\| < 1$ for all n, where $\|\cdot\|$ is any matrix norm. Here, we use the block maximum norm of the matrix, i.e., $\|\cdot\|_{b,\infty}$ in [30], to obtain the mean convergence condition. From the properties of block maximum norm, one can obtain

$$\|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\big(\mathbf{I}_{LK}-\mu\boldsymbol{\mathcal{R}}\big)\|_{b,\infty} \leq \|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\|_{b,\infty}\|\mathbf{I}_{LK}-\mu\boldsymbol{\mathcal{R}}\|_{b,\infty}\cdot$$

Additionally, we have

$$\|\mathbb{E}[\boldsymbol{\mathcal{B}}_n]\|_{b,\infty} = \|\prod_{i=0}^{m-1}\mathbb{E}\left[\mathbf{B}_{n,(i)}\right]\|_{b,\infty} \leq \prod_{i=0}^{m-1}\|\mathbb{E}[\mathbf{B}_{n,(i)}]\|_{b,\infty} \leq 1,$$

and using the similar procedure in [27], [28], one can prove that

$$\|\mathbb{E}[\mathbf{B}_{n,(i)}]\|_{b,\infty} = \|p(\mathbf{A}\otimes\mathbf{I}_L)+(1-p)\mathbf{I}_{LK}\|_{b,\infty} \leq 1\cdot$$

By using [31, Lemma D. 5], it is seen that $\mathbb{E}\big[\tilde{\mathbf{w}}_{net,n}\big]$ converges under the condition $\rho\big(\mathbf{I}_{LK}-\mu\boldsymbol{\mathcal{R}}\big) < 1$, or, equivalently, $\forall k, i : |1-\mu\lambda_i(\mathbf{R}_k)| < 1$, where $\rho(\cdot)$ denotes the spectral radius of the argument matrix. As a result, we obtain the mean convergence condition as

$$0 < \mu < \frac{2}{\max\limits_{\forall i,k}\{\lambda_i(\mathbf{R}_k)\}}\cdot \qquad (18)$$

Accordingly, as long as the step size μ satisfies (18), the operations will converge in the mean.

C. Privacy Analysis

This section examines the privacy of agents in the presence of an HBC agent. The HBC agent is an adversary, but a legitimate agent of the network that has access to information associated with the selection of elements in the partial sharing process and consequently increases the likelihood of information leakage. Let us assume that agent k is an HBC agent trying to estimate the private information of other agents at each time instant n, i.e., $\mathbf{h}_{l,(0)} = \boldsymbol{\psi}_{l,n+1}$ for $l \in \mathcal{N}\setminus\{k\}$. The privacy of agent l is defined as the mean squared estimation error at the adversary attempting to infer the private information as

$$\mathcal{E}_{l,(m)} \triangleq \text{tr}\left(\mathbb{E}[(\hat{\mathbf{h}}_{l,(m)}-\mathbf{h}_{l,(0)})(\hat{\mathbf{h}}_{l,(m)}-\mathbf{h}_{l,(0)})^{\mathrm{T}}]\right) \qquad (19)$$

where $\hat{\mathbf{h}}_{l,(m)}$ denotes the estimate of the private information $\mathbf{h}_{l,(0)}$ after m consensus iterations at the adversary.

The HBC agent has access to its own information and the information exchanged in the neighborhood at each consensus iteration m, i.e., $\{\mathbf{h}_{k,(m)}, \mathbf{S}_{n,(m)}, \mathbf{S}_{n,(m)}\tilde{\mathbf{h}}_{l,(m)}\}$, for $l \in \mathcal{N}_k^-$. Since the HBC agent already knows its own information, the corresponding entries are removed from $\boldsymbol{\omega}_{(m)}, \boldsymbol{\nu}_{(m)}, \mathbf{h}_{(0)}, \text{and}, \boldsymbol{\mathcal{B}}_{n,(m)}$, and denote the quantities with reduced dimensions as $\breve{\boldsymbol{\omega}}_{(m)}, \breve{\boldsymbol{\nu}}_{(m)}, \breve{\mathbf{h}}_{(0)}, \text{and}, \breve{\boldsymbol{\mathcal{B}}}_{n,(m)}$, respectively. From (9), the network-level consensus operation with reduced dimensions can be stated as

$$\tilde{\mathbf{h}}_{(m)} = \Big(\prod_{i=0}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(i)}\Big)\breve{\mathbf{h}}_{(0)} + \sum_{i=0}^{m}\Big(\prod_{j=i}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(j)}\Big)\breve{\boldsymbol{\omega}}_{(i)}\cdot \qquad (20)$$

Without loss of generality, we consider the case where agent K is an HBC agent. At the HBC agent, let $\boldsymbol{\theta}_{(m)} = \mathbf{C}\tilde{\mathbf{h}}_{(m)}$ be the observation vector that comprises the information captured at mth consensus iteration with $\mathbf{C} = \bar{\mathbf{C}}^{\mathrm{T}}\otimes\mathbf{I}_L$ where columns of $\bar{\mathbf{C}} \in \mathbb{R}^{(K-1)\times|\mathcal{N}_K^-|}$ consist of the canonical vectors corresponding to neighbors of agent K. The canonical vector corresponding to agent l, $\mathbf{e}_l \in \mathbb{R}^{K-1}$, is a vector with 1 in the lth entry and zeros elsewhere. Then, following similar procedure as in [15] and substituting (6) in (9), observation model at the HBC agent, after m consensus iterations, is described as

$$\boldsymbol{\vartheta}_{(m)} = \mathbf{H}_{(m)}\breve{\mathbf{h}}_{(0)} + \mathbf{F}_{(m)}\boldsymbol{\upsilon}_{(m)} \qquad (21)$$

where $\boldsymbol{\vartheta}_{(m)} = \text{col}\{\boldsymbol{\theta}_{(0)}, \cdots, \boldsymbol{\theta}_{(m)}\}$, $\mathbf{H}_{(m)} = \text{col}\{\boldsymbol{\mathcal{H}}_{(0)}, \cdots, \boldsymbol{\mathcal{H}}_{(m)}\}$ with $\boldsymbol{\mathcal{H}}_{(m)} = \mathbf{C}\prod_{i=0}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(i)}$, $\breve{\mathbf{h}}_{(0)} = \text{col}\{\mathbf{h}_{1,(0)}, \cdots, \mathbf{h}_{K-1,(0)}\}$, $\boldsymbol{\upsilon}_{(m)} = \text{col}\{\breve{\boldsymbol{\nu}}_{(0)}, \cdots, \breve{\boldsymbol{\nu}}_{(m)}\}$, and

$$\mathbf{F}_{(m)} = \begin{bmatrix} \mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(0)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{F}_{(1),(0)} & \phi\mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{F}_{(2),(0)} & \phi\mathbf{C}\mathbf{F}_{(2),(1)} & \phi^2\mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{F}_{(m),(0)} & \phi\mathbf{C}\mathbf{F}_{(m),(1)} & \phi^2\mathbf{C}\mathbf{F}_{(m),(2)} & \cdots & \phi^m\mathbf{C}\breve{\boldsymbol{\mathcal{B}}}_{n,(m)} \end{bmatrix}$$

with $\mathbf{F}_{(m),(i)} = \prod_{t=i+1}^{m}\breve{\boldsymbol{\mathcal{B}}}_{n,(t)}(\breve{\boldsymbol{\mathcal{B}}}_{n,(i)}-\mathbf{I})$. Using the model in (21) the HBC agent can obtain the maximum likelihood (ML) estimate of $\breve{\mathbf{h}}_{(0)}$, with associated error covariance

$$\mathbf{P}_{(m)} = \Big(\mathbf{H}_{(m)}^{\mathrm{T}}\big(\mathbf{F}_{(m)}\boldsymbol{\Gamma}\mathbf{F}_{(m)}^{\mathrm{T}}\big)^{-1}\mathbf{H}_{(m)}\Big)^{-1} \qquad (22)$$

where $\boldsymbol{\Gamma} = \mathbb{E}\{\boldsymbol{\upsilon}_{(m)}\boldsymbol{\upsilon}_{(m)}^{\mathrm{T}}\} = \sigma_\nu^2\mathbf{I}$. As the HBC agent collects more information from neighbors, the mean squared error of the ML estimator decreases and the privacy metric (19) at each agent k is obtained as

$$\mathcal{E}_{k,(m)} = \text{tr}\left((\mathbf{e}_k^{\mathrm{T}}\otimes\mathbf{I}_L)\mathbf{P}_{(m)}(\mathbf{e}_k\otimes\mathbf{I}_L)\right). \qquad (23)$$

V. Numerical Simulations

To demonstrate the effectiveness of PP-DLMS, we conducted simulations for identifying an unknown system of length $L = 32$. For this, we considered a network of $K = 5$ agents with the adjacency matrix of

$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix},$$

as in [15]. The input signal $x_{k,n}$ and observation noise sequence $\epsilon_{k,n}$, were drawn from zero-mean Gaussian distribution with variance $\sigma_x^2 = 1$ and $\sigma_\epsilon^2 \in \mathcal{U}(0.008, 0.03)$ where $\mathcal{U}(\cdot)$ is the uniform distribution. The average consensus weights

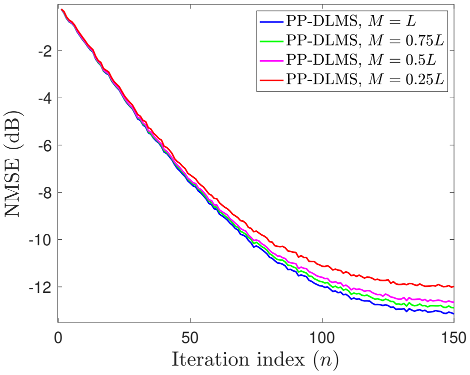


Fig. 1: Network-level MSE (in dB) versus time.

are non-negative coefficients and were obtained through the Metropolis rule [25]. The ACF was iterated for $T = 40$ iterations to approximate the required averages and the perturbation noise sequence at each agent follows (6) with $\phi = 0.9$. The proposed PP-DLMS was simulated under coordinated partial-sharing scheme for different values of M (say $0.75L$, $0.5L$, $0.25L$) and the network-level MSE (NMSE) was considered as the performance metric. The results were obtained against the injected noise variance σ_ν^2, by averaging over 500 independent experiments.

Firstly, the learning curves (i.e., NMSE in dB vs iteration index n) for perturbation noise variance $\sigma_\nu^2 = 5$ are shown in the Fig. 1. Next, for different values of σ_ν^2, the steady-state NMSE is displayed in Fig. 2. From these plots, it can be observed that the proposed PP-DLMS scheme simultaneously achieves energy efficiency and privacy at the cost of a slight degradation in the NMSE. This performance degradation is inversely proportional to the amount of information shared during the average consensus operations. The degradation in performance increases with less information shared at each iteration, smaller M, resulting in a larger NMSE.

Finally in the presence of an HBC agent, agent 5 in the network, the privacy metric (23) versus σ_ν^2 for different values of M is illustrated in Fig. 3. A similar breach of privacy occurs with agent 4 as in [15], and agent 3 obtains identical privacy as agent 1 due to symmetric topology, they are omitted in Fig. 3. From Fig. 3, it can be seen that the proposed PP-DLMS provides a reasonable privacy-energy trade-off. For the case of sharing $M = 0.75L$, the algorithm achieves the same level of privacy as in the case of full information sharing. In the case of sharing less information, $M = 0.5L$ and $M = 0.25L$, the level of privacy decreases, however since smaller portions of information are shared at each consensus iteration, the HBC agent must collect information for more consensus iterations to accurately estimate the private information of other agents.

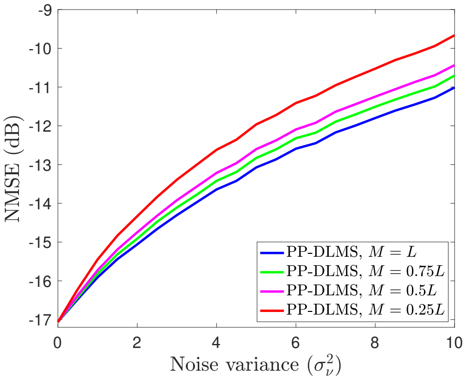


Fig. 2: Network-level MSE (in dB) for different values of M, i.e., the portion of the shared information, versus injected noise variance σ_ν^2.

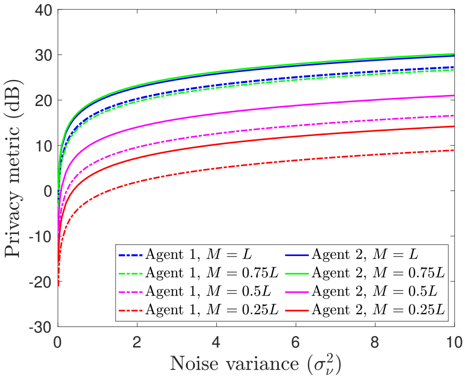


Fig. 3: Agent privacy (in dB) for different values of M versus injected noise variance σ_ν^2.

VI. Conclusions

This paper proposed an energy-efficient and privacy-preserving distributed LMS algorithm. By allowing each agent to share only a fragment of perturbed local information with its neighbors, the proposed private-partial distributed LMS (PP-DLMS) simultaneously achieved both energy-efficiency and privacy-preservation. A mean-convergence analysis of the proposed PP-DLMS algorithm has been conducted to examine the impact of partial-sharing of information on the estimation performance. Further, agent privacy has been characterized in the presence of an honest-but-curious (HBC) adversary, in order to investigate the privacy-preserving properties of the proposed algorithm. Analytical results revealed that the PP-DLMS is resilient to the perturbation sequence and provides a fair energy-privacy trade-off against HBC agents. Numerical simulations have validated the analytical findings.

References

[1] J. B Predd, S. B Kulkarni, and H. V Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Process. Mag.*, Vol. 23, no. 4, pp. 56–69, Jul., 2006.

[2] T.H. Chang, M. Hong, H.T. Wai, X. Zhang, and S. Lu, “Distributed learning in the nonconvex world: From batch data to streaming and beyond,” *IEEE Signal Process. Mag.*, Vol. 37, no. 3, pp. 26–38, May, 2020.

[3] T.H. Chang, M. Hong, and X. Wang, “Multi-agent distributed optimization via inexact consensus ADMM,” *IEEE Trans. Signal Process.*, Vol. 63, no. 2, pp. 482–497, Jan., 2014.

[4] K. Yuan, B. Ying, X. Zhao, and A. H Sayed, “Exact diffusion for distributed optimization and learning—Part I: Algorithm development,” *IEEE Trans. Signal Process.*, Vol. 67, no. 3, pp. 708–723, Feb., 2018.

[5] K. Yuan, B. Ying, X. Zhao, and A. H Sayed, “Exact diffusion for distributed optimization and learning—Part II: Convergence analysis,” *IEEE Trans. Signal Process.*, Vol. 67, no. 3, pp. 724–739, Feb., 2018.

[6] Q. Li, J. S. Gundersen, R. Heusdens, and M. G. Christensen, “Privacy-preserving distributed processing: metrics, bounds and algorithms,” *IEEE Trans. Inf. Forensics Security.*, Vol. 16, no. 3, pp. 2090–2103, Jan., 2021.

[7] R. L. Lagendijk, Z. Erkin, and M. Barni, “Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation,” *IEEE Signal Process. Mag.*, Vol. 30, no. 1, pp. 82–105, Jan., 2013.

[8] K. Kogiso, and T. Fujita, “Cyber-security enhancement of networked control systems using homomorphic encryption,” *Proc. 54th IEEE Conf. Decis. and Control*, pp. 6836–6843, 2015.

[9] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, “Multiparty computation from somewhat homomorphic encryption,” *Springer Annu. Cryptology Conf.*, pp. 643–662, 2012.

[10] J. He and L. Cai and X. Guan, “Differential private noise adding mechanism and its application on consensus algorithm,” *IEEE Trans. Signal Process.*, Vol. 68, pp. 4069-4082, Jul., 2020.

[11] E. Nozari, P. Tallapragada, J. Cortés, “Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design,” *Elsevier Automatica*, Vol. 81, pp. 221–231, Jul., 2017.

[12] N. K. D. Venkategowda, S. Werner, “Privacy-preserving distributed maximum consensus,” *IEEE Signal Process. Lett.*, Vol. 27, pp. 1839–1843, Oct., 2020.

[13] A. Moradi, N. K. Venkategowda, S. Werner, “Coordinated data- falsification attacks in consensus-based distributed Kalman filtering,” *Proc. 8th IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, pp. 495–499, 2019.

[14] A. Moradi, N. K. Venkategowda, S. P. Talebi, S. Werner, “Distributed Kalman Filtering with Privacy against Honest-but-Curious Adversaries,” *Proc. 55th IEEE Asilomar Conf. Signals, Syst., Computers*, pp. 790–794, 2021.

[22] S. Xie, H. Li, “Distributed LMS estimation over networks with quantized communications,” *Int. J. Control*, Vol. 86, no. 3, pp. 478–492, Apr., 2013.

[15] Y. Mo and R.M. Murray, “Privacy preserving average consensus,” *IEEE Trans. Autom. Control*, Vol. 62, no. 2, pp. 753–765, Feb., 2017.

[16] J. He, L. Cai, X. Guan, “Preserving data-privacy with added noises: Optimal estimation and privacy analysis,” *IEEE Trans. Inf. Theory*, Vol. 64, no. 8, pp. 5677–5690, Aug., 2018.

[17] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, “Privacy-preserving average consensus: privacy analysis and algorithm design,” *IEEE Trans. Signal Inf. Process. Netw.*, Vol. 5, no. 1, pp. 127–138, Mar., 2019.

[18] A. Moradi, N. K. Venkategowda, S. P. Talebi, S. Werner, “Securing the Distributed Kalman Filter Against Curious Agents,” *Proc. 24th IEEE Int. Conf. Inf. Fusion*, pp. 1–7, 2021.

[19] Y. Wang, “Privacy-preserving average consensus via state decomposition,” *IEEE Trans. Autom. Control*, Vol. 64, no. 11, pp. 4711–4716, Nov., 2019.

[20] W. Wang, D. Li, X. Wu, and S. Xue, “Average consensus for switching topology networks with privacy protection,” *Proc. IEEE Chinese Automat. Congr.*, pp. 1098–1102, 2019.

[21] S. Chouvardas, K. Slavakis, and S. Theodoridis, “Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction,” *IEEE J. Sel. Topics Signal Process.*, Vol. 7, no. 2, pp. 257–273, Apr., 2013.

[23] C. G. Lopes, and A. H. Sayed, “Diffusion adaptive networks with changing topologies,” *IEEE Int. Conf. Acoust., Speech Signal Process.*, pp. 3285–3288, 2008.

[24] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Distributed LMS for consensus-based in-network adaptive processing,” *IEEE Trans. Signal Process.*, Vol. 57, no. 6, pp. 2365–2382, June, 2009.

[25] L. Xiao, S. Boyd and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. Int. Conf. Info. Process. in Sensor Networks*, 2005, pp. 63–70.

[26] L. Xiao, Stephen Boyd, “Fast linear iterations for distributed averaging,” *Syst. & Control Lett.*, Vol. 53, no. 1, pp. 65-78, 2004.

[27] R. Arablouei, S. Werner, Y. F. Huang and K. Doğançay, “Distributed least mean-square estimation with partial diffusion,” *IEEE Trans. Signal Process.*, Vol. 62, no. 2, pp. 472–484, Jan., 2013.

[28] R. Arablouei,K. Doğançay, S. Werner and Y. F. Huang, “Adaptive distributed estimation based on recursive least-squares and partial diffusion,” *IEEE Trans. Signal Process.*, Vol. 62, no. 14, pp. 3510–3522, Jul., 2014.

[29] V. C. Gogineni and M. Chakraborty, “Partial diffusion affine projection algorithm over clustered multitask networks,” in *Proc. IEEE Int. Symp. Circuits and Syst.*, 2019, pp. 1-5.

[30] A. H. Sayed, “*Adaptation, learning, and optimization over networks*,” Found. Trends Mach. Learn., vol. 7, no. 4–5, pp. 311–801, 2014.

[31] A. H . Sayed, “Diffusion adaptation over networks,” in *Academic Press Library in Signal Process.,* vol. 3, pp. 322-453, Elsevier, 2014.

Appendix D

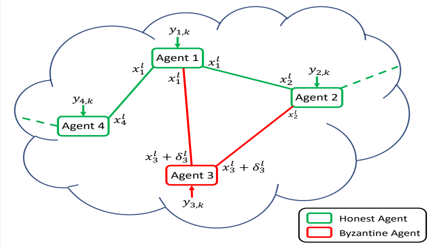
Publication in Chapter 6

P6: A. Moradi, N. K. D. Venkategowda and S. Werner, “Total Variation based Distributed Kalman Filtering for Resiliency Against Byzantines,” in *IEEE Sensors Journal*, pp. 1-11, Jan. 2023.

Total Variation based Distributed Kalman Filtering for Resiliency Against Byzantines

Ashkan Moradi[1], Naveen K. D. Venkategowda[2], *Member, IEEE*, Stefan Werner[1], *Fellow, IEEE*

***Abstract*—This paper proposes a distributed Kalman filter (DKF) with enhanced robustness against Byzantine adversaries. A Byzantine agent is a legitimate network agent that, unlike an honest agent, manipulates information before sharing it with neighbors to impair the overall system performance. In contrast to the literature, the DKF is modeled as a distributed optimization problem where resiliency against Byzantine agents is accomplished by employing a total variation (TV) penalty term. We utilize a distributed subgradient algorithm to compute the state estimate and error covariance matrix updates of the DKF. Additionally, we prove that the proposed suboptimal solution converges to a neighborhood of the optimal centralized solution of the Kalman filter (KF) with a bounded radius when Byzantine agents are present. Numerical simulations corroborate the theoretical findings and demonstrate the robustness of the proposed DKF against Byzantine attacks.**



***Index Terms*—Multiagent network, Kalman filtering, Distributed optimization, Byzantine attack, attack robustness**

I. Introduction

DISTRIBUTED filtering techniques have found widespread use in diverse applications such as environmental monitoring, smart grids, and state estimation [1]–[4]. Due to the lack of a fusion center in distributed Kalman filtering scenarios, agents rely on local interactions to complete a common task across the network [5], [6]. However, the local collaboration renders distributed Kalman filtering susceptible to security attacks.

Attacks on multi-agent networks can be classified as either active or passive; for example, a passive attack can be an eavesdropper intercepting a communication link between agents in order to obtain information [7]. On the other hand, active attacks include denial-of-service attacks (DoS) and data falsification attacks. During DoS attacks, agents cannot exchange information due to link blockages [8]. In contrast, in data falsification attacks, false information is injected into the network [9] by either external adversaries or malicious agents, also termed Byzantine agents, to degrade the overall system performance. Data falsification attacks can be performed independently by each Byzantine agent or designed cooperatively in order to maximize system degradation [9].

Data falsification attacks, in general, have been extensively studied to analyze the impact of malicious behaviors on distributed filtering and estimation [10]–[18]. One approach to reducing the impact of malicious adversaries on the network performance is to detect them and counteract their actions by implementing correction measures [19]–[21]. For example, [22] proposed a defense strategy for a distributed recursive filter by detecting adversarial attacks based on changes in innovation signals of agents and redesigning their gains. Several studies have been proposed in the literature to design an optimal data falsification attack from the perspective of an adversary that evades detection [23]–[26]. For example, the authors in [23] and [24] propose stealthy linear data falsification attacks in remote state estimation scenarios assuming K-L divergence and χ^2 detectors, respectively. Furthermore, the integrity attack also includes stealthy attack strategies, which inject false data into the network without being detected [27]–[29]. In contrast, [25], [26] mainly focus on designing attacks to ensure that the probability of detection does not exceed a given threshold. These have shown that relying on attack detection to limit the impact of adversaries has limited utility in the presence of stealth attacks. Hence, there is a need for a robust algorithm that can operate effectively even when unidentified attacks occur [30].

To that end, works in [31], [32] propose using the statistics of innovation signals to re-design the consensus weights of agents in distributed signal detection and filtering scenarios to minimize the impact of Byzantine agents. A Byzantine-resilient distributed state estimation algorithm is proposed in [33], which allows agents to update state estimates locally by selecting the best subset of neighbors to be effective in updating the state estimate. To reduce computational resources, in [34], [35], distributed state estimation approaches provide

[*] This work was supported in part by the Research Council of Norway.

[1] A. Moradi and S. Werner are with the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway, E-mail: {ashkan.moradi, stefan.werner}@ntnu.no. S. Werner is also with the Department of Signal Processing and Acoustics, Aalto University, Finland

[2] N. K. D. Venkategowda is with Linköping University, Norrköping, Sweden, E-mail: naveen.venkategowda@liu.se.

resilience against measurement attacks by assigning adaptive weights to received measurements from neighbors. By assigning smaller weights to measurements whose norm exceeds a certain threshold, they would have a smaller impact on updating state estimates. The studies in [36], [37] investigate the problem of multi-sensor estimation under undetectable attacks. From the perspective of an adversary, authors in [36], [37] design the attack to maximize the estimation error of the network. Moreover, the gains of the estimator are re-designed in order to mitigate the impact of the designed optimal attack. In addition, a secure state estimation strategy with triple-loop local state observers is proposed in [38], while in [39], the secure state estimation problem is solved by a local observer that achieves robustness against sensor attacks by employing the median of its local estimates.

Homomorphic encryption schemes have been proposed to further ensure the confidentiality of the signals sent over the network [40]. In [41], the authors propose employing additively homomorphic encryption, which enables the cloud server and security module to integrate the information of multiple parties while maintaining data privacy. However, the authors in [42] propose a modified encoding and decoding scheme that, unlike the previous work in [43], does not negatively affect estimation performance in the absence of attacks and further protects data integrity in multi-sensor networks. Moreover, utilizing randomization-based methods to disrupt and mislead attackers in their malicious activities is a less resource-intensive method to mitigate the impact of adversarial attacks in the network [44]. Furthermore, to improve network resistance in the presence of adversarial attacks, [45], [46] introduced a redundancy-based approach for CPSs at different levels of communication, channels, software, and hardware. Redundant subsystems serve as backups or parallel integrity verification units to reduce the effect of malfunctioning behaviors in the network [47]. However, an approach based on redundancy demands strict network requirements and can only tolerate a few Byzantine adversaries. Accordingly, the authors in [48] reduce the stringent requirements of redundancy to only a group of agents and make them resistant to attacks. Generally, these approaches reduce the impact of adversarial attacks on the network. Still, they require more local computations and information transfer in the network, which is undesirable in resource-constrained situations.

The Kalman filtering algorithm has been modeled as an optimization problem. However, this optimization-based approach has not been analyzed in adversarial situations or adapted for robustness in the presence of Byzantine agents. Therefore, contrary to the literature, we propose a distributed Kalman filtering algorithm modeled as an optimization problem with total variation-based constraints that provides robustness to coordinated Byzantine attacks. First, we design the filtering algorithm by adapting the framework proposed in [49] to model the Kalman filtering operations as a solution to an optimization problem and using the TV-norm penalty in the objective function to enforce resiliency against data-falsification attacks in [50]–[52]. Then, we solve the TV-norm-penalized optimization problem using a distributed subgradient algorithm that updates the state estimate for all agents through local collaborations. Furthermore, we model the error covariance update of agents as a TV-norm-penalized optimization problem, which is solved by a similar subgradient approach in the presence of Byzantine agents. Moreover, we show that the proposed TV-norm penalized optimization problem corresponding to the state estimate update results in the same solution as the centralized Kalman filter (CKF). In addition, in the presence of Byzantine agents, we show that the proposed suboptimal solution for the state estimate update, obtained by the subgradient algorithm, converges to a bounded neighborhood of the optimal solution. Finally, we provide numerical simulations to demonstrate the resiliency against Byzantine behavior by obtaining lower filtering mean square error (MSE).

The remainder of this article is organized as follows. Section II presents the problem formulation and provides background information. Section III proposes a DKF with a TV-norm penalized objective function that is robust against Byzantine agents. Section IV presents the convergence of the proposed TV-norm-penalized distributed optimization problem to a bounded neighborhood of the CKF solution. Finally, numerical results are provided in Section V to demonstrate the resiliency against Byzantines, and Section VI concludes the article.

Mathematical Notation: Scalars are denoted by lowercase letters, column vectors by bold lowercase, and matrices by bold uppercase. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-1}$ denote the transpose and inverse operators, respectively. The symbol $\mathbf{1}_m$ represents the $m \times 1$ column vector with all entries equal to one, and $\mathbf{I}_m$ is the $m \times m$ identity matrix. The trace operator is denoted as $\mathrm{tr}(\cdot)$, whereas the greater than and less than symbols in the scalar inequalities are represented by $>$ and $<$, respectively. A positive semidefinite matrix $\mathbf{A}$ is denoted by $\mathbf{A} \succcurlyeq 0$ and $\mathbf{A} \succcurlyeq \mathbf{B}$ indicates that $\mathbf{A} - \mathbf{B}$ is a positive semidefinite matrix. The element-wise sign function is represented by $\mathsf{sign}(\cdot)$ where given $x > 0$, $\mathsf{sign}(x) = 1$ and $\mathsf{sign}(x) = -1$ when $x < 0$. In case of $x = 0$, the value of $\mathsf{sign}(x)$ can be any arbitrary value within $[-1, 1]$. The half vectorization of a symmetric matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$is denoted by $\mathsf{vec}_h(\mathbf{M}) \in \mathbb{R}^{m(m+1)/2}$, where $\mathsf{vec}_h(\mathbf{M}) = [M_{1,1}, \cdots, M_{1,m}, M_{2,2}, \cdots, M_{2,m}, \cdots, M_{m,m}]^{\mathrm{T}}$ with M_{ij} as the ijth element of $\mathbf{M}$. The operator of $\mathsf{vec}_h^{-1}(\cdot)$ denotes the inverse function of $\mathsf{vec}_h(\cdot)$, i.e., $\mathsf{vec}_h^{-1}(\mathsf{vec}_h(\mathbf{M})) = \mathbf{M}$. The stacked vector $\mathbf{x} = [\mathbf{a}]_{i=1}^{N} \in \mathbb{R}^{Nm}$ corresponds to N times stacking the smaller vector $\mathbf{a} \in \mathbb{R}^m$ together.

II. Background and Problem Formulation

Consider a network modeled as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ represents agents of the network and $\mathcal{E}$ is the set of edges that represent communication links between agents, i.e., $(i, j) \in \mathcal{E}$ if nodes i and j are connected. Additionally, the set $\mathcal{N}_i$ specifies the neighborhood of node i and does not include the node itself. The cardinality of the set $\mathcal{N}_i$ is denoted by $|\mathcal{N}_i|$, while $N = |\mathcal{N}|$ is the number of agents in the network.

A. Distributed Kalman Filter (DKF)

We revisit the DKF problem that is modeled as a maximum likelihood estimation problem and represents the relationship between a KF [5] and an optimization problem [49]. The state-space model characterizes the state vector evolution and observation vectors and is given by

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \tag{1}$$
$$\mathbf{y}_{i,k} = \mathbf{H}_i\mathbf{x}_k + \mathbf{v}_{i,k} \tag{2}$$

where for time instant k, $\mathbf{F} \in \mathbb{R}^{m\times m}$ denotes the state transition matrix, $\mathbf{H} = [\mathbf{H}_1^{\mathrm{T}}, \cdots, \mathbf{H}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Nn\times m}$ denotes the network observation matrix, $\mathbf{y}_k = [\mathbf{y}_{1,k}^{\mathrm{T}}, \cdots, \mathbf{y}_{N,k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Nn}$ is the network observation vector, and $\mathbf{w}_k \in \mathbb{R}^m$ and $\mathbf{v}_k = [\mathbf{v}_{1,k}^{\mathrm{T}}, \cdots, \mathbf{v}_{N,k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Nn}$, are process and observation noises, respectively. The process noise $\mathbf{w}_k$ and observation noise $\mathbf{v}_k$ are zero-mean white Gaussian noise processes with a covariance matrices $\mathbf{Q} \in \mathbb{R}^{m\times m}$ and $\mathbf{R} \in \mathbb{R}^{Nn\times Nn}$, respectively, where $\mathbf{R} \triangleq \mathsf{diag}(\{\mathbf{R}_i\}_{i=1}^N)$ and $\mathbf{R}_i = \mathbb{E}\{\mathbf{v}_{i,k}\mathbf{v}_{i,k}^{\mathrm{T}}\} \in \mathbb{R}^{n\times n}$. We assume that the pair $(\mathbf{F}, \mathbf{H})$ is observable and observation noise sequences are uncorrelated. Every agent estimates the state of the network by processing its local and neighboring information. A local estimate for each agent must be provided in a way that the local mean squared error of the agent is minimized.

B. Byzantine Attack Strategy

We assume a distributed setting in which a subset of agents $\mathcal{B}$ are Byzantines, i.e., $\mathcal{B} \subset \mathcal{N}$, and unlike honest agents, they share the manipulated version of their local estimates. In order to update the *a posteriori* state estimate, agents need information exchange with neighbors; we, therefore, assume that Byzantine agents falsify their state estimate before sharing it with neighbors at each iteration. The shared state estimate can be modeled as

$$\tilde{\mathbf{x}}_{i,k}^l = \begin{cases} \mathbf{x}_{i,k}^l + \boldsymbol{\delta}_i^l & i \in \mathcal{B} \\ \mathbf{x}_{i,k}^l & i \notin \mathcal{B} \end{cases} \tag{3}$$

where at agent i and iteration l, $\mathbf{x}_{i,k}^l$ denotes the state estimate and $\boldsymbol{\delta}_i^l \in \mathbb{R}^m$ is the perturbation sequence of the Byzantine agent. To maximize the attack stealthiness, as shown in [53], [54], we consider the perturbation sequence to be a zero-mean Gaussian sequence with covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{m\times m}$. Moreover, in order to maximize the damage caused by the Byzantine attack, we assume that Byzantines design a coordinated attack with covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}\{\boldsymbol{\delta}^l(\boldsymbol{\delta}^l)^{\mathrm{T}}\} \in \mathbb{R}^{Nm\times Nm}$ where $\boldsymbol{\delta}^l = [(\boldsymbol{\delta}_1^l)^{\mathrm{T}}, \cdots, (\boldsymbol{\delta}_N^l)^{\mathrm{T}}]^{\mathrm{T}}$ is the network-wide perturbation sequence and $\boldsymbol{\delta}_i^l = \mathbf{0}$ if $i \notin \mathcal{B}$.

III. Byzantine-Robust Distributed Kalman Filter (BR-DKF)

We consider a network of N agents and assume each agent runs a local KF without sending information to a fusion center. Instead, agents exchange information with their neighbors to develop their optimal estimates. The communication network is considered as graph $\mathcal{G}$ with adjacency and Laplacian matrices $\mathbf{E}$ and $\mathbf{L}$, respectively. Each agent $i \in \mathcal{N}$ updates its local estimate by employing the local observation vector in (2). Similar to the centralized case in [49], the DKF also requires two steps of prediction and correction, where for each agent i and time instant k, the prediction updates are modeled as

$$\hat{\mathbf{x}}_{i,k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{i,k-1} \tag{4}$$
$$\mathbf{P}_{i,k|k-1} = \mathbf{F}\mathbf{P}_{i,k-1}\mathbf{F}^{\mathrm{T}} + \mathbf{Q} \tag{5}$$

with $\hat{\mathbf{x}}_{i,k-1}$ and $\mathbf{P}_{i,k-1} = \mathbb{E}\{\mathbf{e}_{i,k-1}\mathbf{e}_{i,k-1}^{\mathrm{T}}\}$ being the state estimate and error covariance matrix at time instant $k-1$, and $\mathbf{e}_{i,k-1} = \mathbf{x}_{k-1} - \hat{\mathbf{x}}_{i,k-1}$. The *a priori* state estimate and error covariance are denoted by $\hat{\mathbf{x}}_{i,k|k-1}$ and $\mathbf{P}_{i,k|k-1} = \mathbb{E}\{\mathbf{e}_{i,k|k-1}\mathbf{e}_{i,k|k-1}^{\mathrm{T}}\}$, respectively, with $\mathbf{e}_{i,k|k-1} = \mathbf{x}_k - \hat{\mathbf{x}}_{i,k|k-1}$.

The correction steps of the DKF can be modeled as the solution of a constrained optimization problem [49]; in particular, the *a posteriori* state estimates can be obtained by solving the optimization problem

$$\min_{\{\mathbf{x}_{i,k}\}_{i=1}^N} \sum_{i=1}^N f_i(\mathbf{x}_{i,k}) \quad \text{s. t. } \mathbf{x}_{i,k} = \mathbf{x}_{j,k}, \ \forall j \in \mathcal{N}_i, i = 1, 2, \cdots, N \tag{6}$$

where the local objective function $f_i(\mathbf{x}_{i,k})$ is given by

$$f_i(\mathbf{x}_{i,k}) = \frac{1}{2}\Big((\mathbf{y}_{i,k} - \mathbf{H}_i\mathbf{x}_{i,k})^{\mathrm{T}}\mathbf{R}_i^{-1}(\mathbf{y}_{i,k} - \mathbf{H}_i\mathbf{x}_{i,k}) \tag{7}$$
$$+ \frac{1}{N}(\mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k|k-1})^{\mathrm{T}}\mathbf{P}_{i,k|k-1}^{-1}(\mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k|k-1})\Big)$$

and the constraints enforce consensus across all the agents in the network. The distributed Kalman filtering problem can be solved by any distributed algorithm that finds the optimal solutions in (6), i.e., $\mathbf{x}_{i,k}^*$ for each $i \in \mathcal{N}$. Subsequently, the *a posteriori* state estimates of agents are obtained as $\hat{\mathbf{x}}_k = [\hat{\mathbf{x}}_{1,k}^{\mathrm{T}}, \cdots, \hat{\mathbf{x}}_{N,k}^{\mathrm{T}}]^{\mathrm{T}}$ where $\hat{\mathbf{x}}_{i,k} = \mathbf{x}_{i,k}^*$.

Motivated by [50], [51], the constraints in (6) can be approximated by a TV-norm penalty which also endows robustness to data falsification attacks. In the absence of a Byzantine agent in the network, the TV-norm-penalized problem of (6) can be formulated as

$$\mathbf{x}_{c_k}^* = \min_{\{\mathbf{x}_{i,k}\}_{i=1}^N} \sum_{i=1}^N \left(f_i(\mathbf{x}_{i,k}) + \frac{\lambda_{\text{tv}}}{2}\sum_{j\in\mathcal{N}_i} \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|_1\right) \tag{8}$$

where $\mathbf{x}_{c_k}^* = [(\mathbf{x}_{1,k}^*)^{\mathrm{T}}, \cdots, (\mathbf{x}_{N,k}^*)^{\mathrm{T}}]^{\mathrm{T}}$ and λ_{tv} is a penalty parameter. Due to the penalty parameter λ_{tv}, estimates $\mathbf{x}_{i,k}$ and $\mathbf{x}_{j,k}$ are forced to be close. The larger the λ_{tv}, the closer $\mathbf{x}_{i,k}$ and $\mathbf{x}_{j,k}$ become. However, the TV-norm penalty allows for some pairs of $\mathbf{x}_{i,k}$ and $\mathbf{x}_{j,k}$ to be different, which is crucial when Byzantine agents are present in the network.

We solve the optimization problem in (8) with a subgradient method [51], and derive the state estimate update at each agent $i \in \mathcal{N}$ as

$$\mathbf{x}_{i,k}^{l+1} = \mathbf{x}_{i,k}^l - \alpha_k\left(\nabla_{\mathbf{x}_{i,k}} f_i(\mathbf{x}_{i,k}^l) + \lambda_{\text{tv}}\sum_{j\in\mathcal{N}_i} \mathsf{sign}(\mathbf{x}_{i,k}^l - \mathbf{x}_{j,k}^l)\right) \tag{9}$$

where $\alpha_k > 0$ denotes the step size and $\mathbf{x}_{i,k}^l$ is the state estimate of the subgradient method at agent i and iteration

l. Assuming that a group of agents is conducting Byzantine attacks on the network, i.e., $\mathcal{B} \subset \mathcal{N}$, and by substituting the gradient $\nabla_{\mathbf{x}_{i,k}} f_i(\mathbf{x}_{i,k})$, we obtain

$$\mathbf{x}_{i,k}^{l+1} = \mathbf{x}_{i,k}^{l} - \alpha_k \Bigg(\boldsymbol{\Omega}_{i,k}\mathbf{x}_{i,k}^{l} - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{R}_i} \mathsf{sign}(\mathbf{x}_{i,k}^{l} - \mathbf{x}_{j,k}^{l}) + \lambda_{\text{tv}} \sum_{j \in \mathcal{B}_i} \mathsf{sign}(\mathbf{x}_{i,k}^{l} - \tilde{\mathbf{x}}_{j,k}^{l}) \Bigg) \tag{10}$$

where $\tilde{\mathbf{x}}_{j,k}^{l} = \mathbf{x}_{j,k}^{l} + \boldsymbol{\delta}_j^l$ is the state estimate received from the jth Byzantine neighbor, $\mathcal{R}_i$ and $\mathcal{B}_i$ include honest and Byzantine members of $\mathcal{N}_i$, and

$$\begin{aligned} \boldsymbol{\Omega}_{i,k} &= \mathbf{H}_i^{\mathrm{T}} \mathbf{R}_i^{-1} \mathbf{H}_i + \frac{1}{N} \mathbf{P}_{i,k|k-1}^{-1} \\ \boldsymbol{\theta}_{i,k} &= \mathbf{H}_i^{\mathrm{T}} \mathbf{R}_i^{-1} \mathbf{y}_{i,k} + \frac{1}{N} \boldsymbol{\Omega}_{i,k|k-1} \hat{\mathbf{x}}_{i,k|k-1} \end{aligned} \tag{11}$$

with $\boldsymbol{\Omega}_{i,k|k-1} = \mathbf{P}_{i,k|k-1}^{-1}$. Regardless of the state estimate received from neighbors, the value of $\mathsf{sign}(\mathbf{x}_{i,k}^{l} - \tilde{\mathbf{x}}_{j,k}^{l})$ is restricted to $[-1, 1]$. Thus, the last term in (10) limits the effect of perturbed data received from a Byzantine agent, so that the state estimate update is more resistant to Byzantine attacks.

Similarly, the error covariance update also requires designing an optimization problem to obtain the average consensus of the information matrices $N\boldsymbol{\Omega}_{i,k}$ throughout the network. To this end, we propose the following optimization problem that derives the error covariance update

$$\begin{aligned} &\min_{\{\boldsymbol{\zeta}_i\}_{i=1}^N} \sum_{i=1}^{N} \|\boldsymbol{\zeta}_i - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,k})\|_F^2 \\ &\text{s. t.} \qquad \boldsymbol{\zeta}_i = \boldsymbol{\zeta}_j, \ \ \forall j \in \mathcal{N}_i, i = 1, 2, \cdots, N. \end{aligned} \tag{12}$$

The optimal solution of (12) is denoted by $\boldsymbol{\zeta}^* = [(\boldsymbol{\zeta}_1^*)^{\mathrm{T}}, \cdots, (\boldsymbol{\zeta}_N^*)^{\mathrm{T}}]^{\mathrm{T}}$ which returns the average of $\mathsf{vec}_h(N\boldsymbol{\Omega}_{i,k})$ throughout the entire network. Subsequently, the error covariance matrix can be updated as $\mathbf{P}_{i,k} = (\mathsf{vec}_h^{-1}(\boldsymbol{\zeta}_i^*))^{-1}$. Motivated by the TV-norm-penalized optimization problem in (8), we modify the optimization problem in (12) as

$$\boldsymbol{\zeta}^* = \min_{\{\boldsymbol{\zeta}_i\}_{i=1}^N} \sum_{i=1}^{N} \Bigg(g_i(\boldsymbol{\zeta}_i) + \frac{\lambda_{\text{tv}}}{2} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{\zeta}_i - \boldsymbol{\zeta}_j\|_1 \Bigg) \tag{13}$$

where $g_i(\boldsymbol{\zeta}_i) = \|\boldsymbol{\zeta}_i - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,k})\|_F^2$. Employing a similar subgradient approach as in (9), results in

$$\boldsymbol{\zeta}_i^{l+1} = \boldsymbol{\zeta}_i^{l} - \gamma_k \Bigg(\nabla_{\boldsymbol{\zeta}_i} g_i(\boldsymbol{\zeta}_i^l) + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\boldsymbol{\zeta}_i^l - \boldsymbol{\zeta}_j^l) \Bigg) \tag{14}$$

where $\gamma_k > 0$ denotes the step size and the update equation in (14) is simplified as

$$\boldsymbol{\zeta}_i^{l+1} = \boldsymbol{\zeta}_i^{l} - \gamma_k \Bigg(\boldsymbol{\zeta}_i^l - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,k}) + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\boldsymbol{\zeta}_i^l - \boldsymbol{\zeta}_j^l) \Bigg) \tag{15}$$

After a large enough number of iterations, say l^*, the suboptimal solutions in (10) and (15) converge to $(\mathbf{x}_{i,k}^{l^*}, \boldsymbol{\zeta}_i^{l^*})$ and the filtering *a posteriori* state estimate and error covariance matrix can be updated as

$$\begin{aligned} \hat{\mathbf{x}}_{i,k} &= \mathbf{x}_{i,k}^{l^*} \\ \mathbf{P}_{i,k} &= (\mathsf{vec}_h^{-1}(\boldsymbol{\zeta}_i^*))^{-1}. \end{aligned}$$

Algorithm 1 Byzantine-Robust DKF (BR-DKF)

```
• For each agent $i \in \mathcal{N}$
• Initialize $\hat{\mathbf{x}}_{i,0}$ and $\mathbf{P}_{i,0}$
for all $k > 0$ do
    $\hat{\mathbf{x}}_{i,k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{i,k-1}$
    $\mathbf{P}_{i,k|k-1} = \mathbf{F}\mathbf{P}_{i,k-1}\mathbf{F}^{\mathrm{T}} + \mathbf{Q}$
    $\boldsymbol{\Omega}_{i,k|k-1} = \mathbf{P}_{i,k|k-1}^{-1}$
    $\boldsymbol{\Omega}_{i,k} = \mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{H}_i + \frac{1}{N}\boldsymbol{\Omega}_{i,k|k-1}$
    $\boldsymbol{\theta}_{i,k} = \mathbf{H}_i^{\mathrm{T}}\mathbf{R}_i^{-1}\mathbf{y}_{i,k} + \frac{1}{N}\boldsymbol{\Omega}_{i,k|k-1}\hat{\mathbf{x}}_{i,k|k-1}$
    Set $\mathbf{x}_{i,k}^1 = \mathbf{0}$ and $\boldsymbol{\zeta}_i^1 = \mathbf{0}$
    for $l = 1$ to $l^*$ do
        Share $\mathbf{x}_{i,k}^l + \boldsymbol{\delta}_i^l$ with neighbors if $i \in \mathcal{B}$
        $\mathbf{x}_{i,k}^{l+1} = \mathbf{x}_{i,k}^l - \alpha_k \left( \boldsymbol{\Omega}_{i,k}\mathbf{x}_{i,k}^l - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\mathbf{x}_{i,k}^l - \tilde{\mathbf{x}}_{j,k}^l) \right)$
        $\boldsymbol{\zeta}_i^{l+1} = \boldsymbol{\zeta}_i^l - \gamma_k \left( \boldsymbol{\zeta}_i^l - \mathsf{vec}_h(N\boldsymbol{\Omega}_{i,k}) + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\boldsymbol{\zeta}_i^l - \boldsymbol{\zeta}_j^l) \right)$
    end for
    $\hat{\mathbf{x}}_{i,k} = \mathbf{x}_{i,k}^{l^*}$
    $\mathbf{P}_{i,k} = (\mathsf{vec}_h^{-1}(\boldsymbol{\zeta}_i^{l^*}))^{-1}$
end for
```

Assuming that Byzantine agents manipulate only state estimates, i.e., falsify the state estimate $\mathbf{x}_{i,k}^l$ at each iteration l, Algorithm 1 summarizes detailed operations of the proposed BR-DKF. It can be seen in Algorithm 1 that two additional $\mathsf{sign}(\cdot)$ operations are computed at each iteration l, compared to conventional consensus-based DKFs. The complexity of $\mathsf{sign}(\cdot)$ operator is dominated by the complexity of $O(m^2)$ for the multiplication of $\boldsymbol{\Omega}_{i,k}\mathbf{x}_{i,k}^l$ at each iteration l. As a result, the local computational complexity of the proposed method is the same as that of the conventional consensus-based DKF algorithms.

IV. Performance Analysis

In this section, we demonstrate that the TV-norm-penalized problem in (8) yields a feasible solution when the penalty parameter λ_{tv} is sufficiently large. We also show that the suboptimal solution in (10) converges to a neighborhood of the optimal solution of the problem in (8) with a bounded radius when Byzantine agents are in the network. To assist in future calculations, we define $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N \times |\mathcal{E}|}$ as the node-edge incidence matrix where for each edge $e = (i, j) \in \mathcal{E}$ with $i < j$, we set $a_{ei} = 1$ and $a_{je} = -1$, otherwise, the elements of $\mathbf{A}$ remain zero. In the following Theorem, we establish the optimality of the proposed solution in (8) to yield the same solution as the centralized solution $\hat{\mathbf{x}}_k^*$ in [49]. We provide a lower bound threshold for the penalty parameter λ_{tv} that guarantees convergence of the solution in (8) to the centralized solution in [49].

Theorem 1: Given that the network topology is connected, if $\lambda_{\text{tv}} \geq \lambda_0$ where

$$\lambda_0 = \frac{\sqrt{N}}{\sigma_{\min}(\mathbf{A})} \max_{\forall k} \max_{i \in \mathcal{N}} \|\boldsymbol{\Omega}_{i,k}\mathbf{x}_{i,k}^* - \boldsymbol{\theta}_{i,k}\|_\infty \tag{16}$$

with $\sigma_{\min}(\mathbf{A})$ being the minimum non-zero singular value of $\mathbf{A}$, $\mathbf{\Omega}_{i,k}$ and $\boldsymbol{\theta}_{i,k}$ defined in (11); then, for the optimal solution $\mathbf{x}^*_{c_k}$ in (8) and the optimal solution of the CKF problem $\hat{\mathbf{x}}^*_k$ in [49], we have $\mathbf{x}^*_{c_k} = [\hat{\mathbf{x}}^*_k]_{i=1}^N$·

Proof: The proof begins with stating the fact that for each $i \in \mathcal{N}$, the optimal solution $\mathbf{x}^*_{c_k} = [\mathbf{x}^*_{i,k}]_{i=1}^N$ satisfies the optimality condition

$$\nabla_{\mathbf{x}_{i,k}} f_i(\mathbf{x}^*_{i,k}) + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\mathbf{x}^*_{i,k} - \mathbf{x}^*_{j,k}) = \mathbf{0}\cdot \tag{17}$$

Let us assume $\mathbf{s}_{ij} = \mathsf{sign}(\mathbf{x}^*_{i,k} - \mathbf{x}^*_{j,k})$ and $\boldsymbol{\nu}_{i,k} = \nabla_{\mathbf{x}_{i,k}} f_i(\mathbf{x}^*_{i,k})$;[1] then knowing that $\mathbf{s}_{ji} = -\mathbf{s}_{ij}$, we have

$$\boldsymbol{\nu}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i, i<j} \mathbf{s}_{ij} - \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i, i>j} \mathbf{s}_{ij} = \mathbf{0}\cdot \tag{18}$$

Assuming $\boldsymbol{\nu}_k = [\boldsymbol{\nu}^{\mathrm{T}}_{1,k}, \cdots, \boldsymbol{\nu}^{\mathrm{T}}_{N,k}]^{\mathrm{T}}$ and $\mathbf{s} = [\mathbf{s}^{\mathrm{T}}_1, \cdots, \mathbf{s}^{\mathrm{T}}_{|\mathcal{E}|}]^{\mathrm{T}}$ with $\mathbf{s}_t = \mathbf{s}_{ij}$ for each $(i,j) \in \mathcal{E}$, we have

$$\boldsymbol{\nu}_k + \lambda_{\text{tv}} \mathbf{A}\mathbf{s} = \mathbf{0}\cdot \tag{19}$$

Now the problem is to show that (19) has at least one solution $\mathbf{s}^*$ and due to the structure of $\mathbf{s}$ its elements are within $[-1, 1]$ or $\|\mathbf{s}\|_\infty \leq 1$. The rank of $\mathbf{A}$ is $N-1$ with the column null space of one vector, i.e., $\mathbf{1}_N$, since $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ is bidirectionally connected. In addition, the optimality condition of the centralized solution in [49] satisfies

$$\sum_{i \in \mathcal{N}} \boldsymbol{\nu}_{i,k} = \sum_{i \in \mathcal{N}} \nabla_{\mathbf{x}_{i,k}} f_i(\mathbf{x}^*_{i,k}) = \mathbf{0} \tag{20}$$

which means $\lambda_{\text{tv}}\mathbf{A}$ and $\boldsymbol{\nu}_k$ share the same null space and have the same rank that consequently, states that we will have at least one solution for (19). In order to find the solution that satisfies $\|\mathbf{s}\|_\infty \leq 1$, we consider the least-squares solution as $\mathbf{s} = -\frac{1}{\lambda_{\text{tv}}} \mathbf{A}^\dagger \boldsymbol{\nu}_k$ where $\dagger$ denotes the pseudo inverse. The least-squares solution is bounded as

$$\|\mathbf{s}\|_2 = \frac{1}{\lambda_{\text{tv}}} \|\mathbf{A}^\dagger \boldsymbol{\nu}_k\|_2\cdot \tag{21}$$

Then, we have

$$\|\mathbf{s}\|_2 \leq \frac{1}{\lambda_{\text{tv}}} \sigma_{\max}(\mathbf{A}^\dagger) \|\boldsymbol{\nu}_k\|_2 \leq \frac{1}{\lambda_{\text{tv}} \sigma_{\min}(\mathbf{A})} \|\boldsymbol{\nu}_k\|_2 \tag{22}$$

where $\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ represent the maximum and minimum non-zero singular values of the argument matrix, respectively. Since $\|\mathbf{s}\|_\infty \leq \|\mathbf{s}\|_2$ and $\|\boldsymbol{\nu}_k\|_2 \leq \sqrt{N} \|\boldsymbol{\nu}_k\|_\infty$, we have

$$\|\mathbf{s}\|_\infty \leq \frac{\sqrt{N}}{\lambda_{\text{tv}} \sigma_{\min}(\mathbf{A})} \|\boldsymbol{\nu}_k\|_\infty = \frac{\sqrt{N}}{\lambda_{\text{tv}} \sigma_{\min}(\mathbf{A})} \max_{i \in \mathcal{N}} |\boldsymbol{\nu}_{i,k}|\cdot \tag{23}$$

Thus, $\|\mathbf{s}\|_\infty \leq 1$ if $\lambda_{\text{tv}} \geq \frac{\sqrt{N}}{\sigma_{\min}(\mathbf{A})} \max_{i \in \mathcal{N}} |\boldsymbol{\nu}_{i,k}|$ for each k, which results in the requirement of $\lambda_{\text{tv}} \geq \lambda_0$ where

$$\begin{aligned} \lambda_0 &= \frac{\sqrt{N}}{\sigma_{\min}(\mathbf{A})} \max_{\forall k} \max_{i \in \mathcal{N}} \|\nabla_{\mathbf{x}^*_{c_i}} f_i(\mathbf{x}^*_{c_i})\|_\infty \\ &= \frac{\sqrt{N}}{\sigma_{\min}(\mathbf{A})} \max_{\forall k} \max_{i \in \mathcal{N}} \|\mathbf{\Omega}_{i,k} \mathbf{x}^*_{i,k} - \boldsymbol{\theta}_{i,k}\|_\infty \end{aligned}$$

that completes the proof. ■

After showing the convergence of the proposed method to the desired centralized case, we need to theoretically analyze the performance of the proposed solution in the presence of Byzantines. The following theorem shows that the proposed suboptimal solution in (10) converges to a neighborhood of the optimal centralized solution within a bounded radius despite the presence of Byzantine agents.

Theorem 2: Given the assumptions in Theorem 1 and $\lambda_{\text{tv}} \geq \lambda_0$, at each agent $i \in \mathcal{N}$ and the presence of Byzantine agents, the solution proposed in (10) stays in the neighborhood of the optimal solution $\mathbf{x}^*_{c_k} = [\mathbf{x}^*_{i,k}]_{i=1}^N$ in (8) with radius

$$\lim_{l \to \infty} \mathbb{E}_l \{\|\mathbf{x}^{l+1}_{i,k} - \mathbf{x}^*_{i,k}\|^2\} \leq \frac{\Delta_0}{1 - \|\mathbf{\Delta}\|} \tag{24}$$

where $\mathbf{\Delta} = \left(1 + 2\alpha_k^2 \|\mathbf{\Omega}_{i,k}\|^2 + 2\varepsilon\alpha_k\right)\mathbf{I} - 2\alpha_k \mathbf{\Omega}_{i,k}$, $\Delta_0 = \lambda_{\text{tv}}^2 \alpha_k (4\alpha_k + \frac{1}{\varepsilon})(4|\mathcal{R}_i|^2 + |\mathcal{B}_i|^2)m$, and the step size α_k satisfies

$$\alpha_k \leq \min_{i \in \mathcal{N}} \left\{ \frac{\lambda_{\min}(\mathbf{\Omega}_{i,k}) - \varepsilon}{\|\mathbf{\Omega}_{i,k}\|^2} \right\}\cdot \tag{25}$$

Proof: The proof begins by computing the gap between the optimal solution in (8) and the proposed solution in (10) after l iterations as follows

$$\begin{aligned} \mathbb{E}_l \{\|\mathbf{x}^{l+1}_{i,k} - \mathbf{x}^*_{i,k}\|^2\} &= \mathbb{E}_l \{\|\mathbf{x}^l_{i,k} - \mathbf{x}^*_{i,k} \\ &- \alpha_k \big(\mathbf{\Omega}_{i,k} \mathbf{x}^l_{i,k} - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\mathbf{x}^l_{i,k} - \mathbf{x}^l_{j,k})\big)\|^2\}\cdot \end{aligned} \tag{26}$$

In this case, (26) can be further simplified as

$$\begin{aligned} &\mathbb{E}_l \{\|\mathbf{x}^{l+1}_{i,k} - \mathbf{x}^*_{i,k}\|^2\} = \mathbb{E}_l \{\|\mathbf{x}^l_{i,k} - \mathbf{x}^*_{i,k}\|^2\} \\ &\underbrace{+ \alpha_k^2\, \mathbb{E}_l \{\|\mathbf{\Omega}_{i,k} \mathbf{x}^l_{i,k} - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\mathbf{x}^l_{i,k} - \mathbf{x}^l_{j,k})\|^2\}}_{\beta_1} \\ &\underbrace{- 2\alpha_k < \mathbf{x}^l_{i,k} - \mathbf{x}^*_{i,k}, \mathbf{\Omega}_{i,k} \mathbf{x}^l_{i,k} - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{N}_i} \mathsf{sign}(\mathbf{x}^l_{i,k} - \mathbf{x}^l_{j,k}) >}_{\beta_2} \end{aligned} \tag{27}$$

Considering the optimality condition for the optimal solution $\mathbf{x}^*_{c_k}$ in (8) as

$$\mathbf{\Omega}_{i,k} \mathbf{x}^*_{i,k} - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{R}_i} \mathsf{sign}(\mathbf{x}^*_{i,k} - \mathbf{x}^*_{j,k}) = \mathbf{0}, \tag{28}$$

we have

$$\begin{aligned} \beta_1 =& \mathbb{E}_l \{\|\mathbf{\Omega}_{i,k} \mathbf{x}^l_{i,k} - \boldsymbol{\theta}_{i,k} + \lambda_{\text{tv}} \sum_{j \in \mathcal{R}_i} \mathsf{sign}(\mathbf{x}^l_{i,k} - \mathbf{x}^l_{j,k}) \\ &+ \lambda_{\text{tv}} \sum_{j \in \mathcal{B}_i} \mathsf{sign}(\mathbf{x}^l_{i,k} - \tilde{\mathbf{x}}^l_{j,k}) - \mathbf{\Omega}_{i,k} \mathbf{x}^*_{i,k} + \boldsymbol{\theta}_{i,k} \\ &- \lambda_{\text{tv}} \sum_{j \in \mathcal{R}_i} \mathsf{sign}(\mathbf{x}^*_{i,k} - \mathbf{x}^*_{j,k})\|^2\} \qquad (29) \\ =& \mathbb{E}_l \{\|\mathbf{\Omega}_{i,k} (\mathbf{x}^l_{i,k} - \mathbf{x}^*_{i,k}) + \lambda_{\text{tv}} \sum_{j \in \mathcal{B}_i} \mathsf{sign}(\mathbf{x}^l_{i,k} - \tilde{\mathbf{x}}^l_{j,k}) \\ &+ \lambda_{\text{tv}} \sum_{j \in \mathcal{R}_i} \big(\mathsf{sign}(\mathbf{x}^l_{i,k} - \mathbf{x}^l_{j,k}) - \mathsf{sign}(\mathbf{x}^*_{i,k} - \mathbf{x}^*_{j,k})\big)\|^2\}\cdot \end{aligned}$$

[1] Throughout the article, we remove the index k from $\mathbf{s}_{ij}$ in order to simplify the notation.

Due to the inequality $(a+b)^2 \le 2a^2+2b^2$, we have

$$\begin{aligned}\beta_1 \le\ & 2\mathbb{E}_l\{\|\boldsymbol{\Omega}_{i,k}(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})\|^2\}\\ &+2\lambda_{\text{tv}}^2\mathbb{E}_l\{\|\sum_{j\in\mathcal{R}_i}\big(\mathsf{sign}(\mathbf{x}^l_{i,k}-\mathbf{x}^l_{j,k})-\mathsf{sign}(\mathbf{x}^*_{i,k}-\mathbf{x}^*_{j,k})\big)\\ &+\sum_{j\in\mathcal{B}_i}\mathsf{sign}(\mathbf{x}^l_{i,k}-\tilde{\mathbf{x}}^l_{j,k})\|^2\} \qquad (30)\\ \le\ & 2\mathbb{E}_l\{\|\boldsymbol{\Omega}_{i,k}(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})\|^2\}\\ &+4\lambda_{\text{tv}}^2\underbrace{\mathbb{E}_l\{\|\sum_{j\in\mathcal{B}_i}\mathsf{sign}(\mathbf{x}^l_{i,k}-\tilde{\mathbf{x}}^l_{j,k})\|^2\}}_{\le|\mathcal{B}_i|^2m}\\ &+4\lambda_{\text{tv}}^2\underbrace{\mathbb{E}_l\{\|\sum_{j\in\mathcal{R}_i}\big(\mathsf{sign}(\mathbf{x}^l_{i,k}-\mathbf{x}^l_{j,k})-\mathsf{sign}(\mathbf{x}^*_{i,k}-\mathbf{x}^*_{j,k})\big)\|^2\}}_{\le4|\mathcal{R}_i|^2m}\cdot\end{aligned}$$

Since for the matrix norm $\|\cdot\|$, we have[2]

$$\mathsf{tr}(\mathbf{AB}) \le \min\{\|\mathbf{A}\|\mathsf{tr}(\mathbf{B}),\|\mathbf{B}\|\mathsf{tr}(\mathbf{A})\} \tag{31}$$

where $\mathbf{A}$ and $\mathbf{B}$ are positive semi-definite and $\|\mathbf{AB}\| \le \|\mathbf{A}\|\|\mathbf{B}\|$, we can show that

$$\|\boldsymbol{\Omega}_{i,k}(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})\|^2 \le \|\boldsymbol{\Omega}_{i,k}\|^2\|\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k}\|^2,$$

and subsequently

$$\beta_1 \le 2\|\boldsymbol{\Omega}_{i,k}\|^2\|\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k}\|^2+4\lambda_{\text{tv}}^2(4|\mathcal{R}_i|^2+|\mathcal{B}_i|^2)m\cdot \tag{32}$$

Additionally, we have

$$\begin{aligned}-2\times\beta_2 =\ & -2<\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k},\boldsymbol{\Omega}_{i,k}\mathbf{x}^l_{i,k}-\boldsymbol{\theta}_{i,k}\\ &+\lambda_{\text{tv}}\sum_{j\in\mathcal{N}_i}\mathsf{sign}(\mathbf{x}^l_{i,k}-\mathbf{x}^l_{j,k})\\ &-\boldsymbol{\Omega}_{i,k}\mathbf{x}^*_{i,k}+\boldsymbol{\theta}_{i,k}-\lambda_{\text{tv}}\sum_{j\in\mathcal{R}_i}\mathsf{sign}(\mathbf{x}^*_{i,k}-\mathbf{x}^*_{j,k})>\\ =\ & -2<\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k},\boldsymbol{\Omega}_{i,k}(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})>\\ &-2<\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k},\lambda_{\text{tv}}\sum_{j\in\mathcal{B}_i}\mathsf{sign}(\mathbf{x}^l_{i,k}-\tilde{\mathbf{x}}^l_{j,k})>\\ &-2<\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k},\lambda_{\text{tv}}\sum_{j\in\mathcal{R}_i}\big(\mathsf{sign}(\mathbf{x}^l_{i,k}-\mathbf{x}^l_{j,k})\\ &\quad-\mathsf{sign}(\mathbf{x}^*_{i,k}-\mathbf{x}^*_{j,k})\big)>\cdot \qquad (33)\end{aligned}$$

The inequality $-2ab \le \varepsilon a^2+\frac{b^2}{\varepsilon}$ for each $\varepsilon \ge 0$ gives

$$\begin{aligned}-2<&\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k},\lambda_{\text{tv}}\sum_{j\in\mathcal{B}_i}\mathsf{sign}(\mathbf{x}^l_{i,k}-\tilde{\mathbf{x}}^l_{j,k})>\\ &\le\varepsilon\|\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k}\|^2+\frac{\lambda_{\text{tv}}^2}{\varepsilon}|\mathcal{B}_i|^2m \qquad (34)\end{aligned}$$

and

$$\begin{aligned}-2<&\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k},\lambda_{\text{tv}}\sum_{j\in\mathcal{R}_i}\big(\mathsf{sign}(\mathbf{x}^l_{i,k}-\mathbf{x}^l_{j,k})\\ &-\mathsf{sign}(\mathbf{x}^*_{i,k}-\mathbf{x}^*_{j,k})\big)>\\ &\le\varepsilon\|\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k}\|^2+\frac{4\lambda_{\text{tv}}^2}{\varepsilon}|\mathcal{R}_i|^2m\cdot \qquad (35)\end{aligned}$$

After substituting (29) and (33) in (27), we get

$$\begin{aligned}&\mathbb{E}_l\{\|\mathbf{x}^{l+1}_{i,k}-\mathbf{x}^*_{i,k}\|^2\}\\ &\quad\le\big(1+2\alpha_k^2\|\boldsymbol{\Omega}_{i,k}\|^2+2\varepsilon\alpha_k\big)\,\mathbb{E}_l\{\|\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k}\|^2\}\\ &\quad\quad-2\alpha_k\mathbb{E}_l\{(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})^{\mathrm{T}}\boldsymbol{\Omega}_{i,k}(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})\}\\ &\quad\quad+\lambda_{\text{tv}}^2\alpha_k(4\alpha_k+\frac{1}{\varepsilon})(4|\mathcal{R}_i|^2+|\mathcal{B}_i|^2)m \qquad (36)\\ &=\mathbb{E}_l\{(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})^{\mathrm{T}}\Big(\big(1+2\alpha_k^2\|\boldsymbol{\Omega}_{i,k}\|^2+2\varepsilon\alpha_k\big)\mathbf{I}-2\alpha_k\boldsymbol{\Omega}_{i,k}\Big)\\ &\quad\quad(\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k})\}+\lambda_{\text{tv}}^2\alpha_k(4\alpha_k+\frac{1}{\varepsilon})(4|\mathcal{R}_i|^2+|\mathcal{B}_i|^2)m\cdot\end{aligned}$$

To guarantee that the error is decreasing with each iteration, we must have

$$\big(1+2\alpha_k^2\|\boldsymbol{\Omega}_{i,k}\|^2+2\varepsilon\alpha_k\big)\,\mathbf{I}-2\alpha_k\boldsymbol{\Omega}_{i,k}\preccurlyeq\mathbf{I} \tag{37}$$

that yields

$$2\alpha_k\left(\alpha_k\|\boldsymbol{\Omega}_{i,k}\|^2\mathbf{I}+\varepsilon\mathbf{I}-\boldsymbol{\Omega}_{i,k}\right)\preccurlyeq0 \tag{38}$$

Since $\alpha_k \ge 0$ and by assuming $\bar{\alpha}_k=\alpha_k\|\boldsymbol{\Omega}_{i,k}\|^2+\varepsilon$, we only need to have

$$\boldsymbol{\Omega}_{i,k}-\bar{\alpha}_k\mathbf{I}\succcurlyeq0 \tag{39}$$

which requires $\bar{\alpha}_k \le \lambda_j(\boldsymbol{\Omega}_{i,k})$ for each $j=1,2,\cdots,m$ that means

$$\alpha_k \le \frac{\lambda_{\min}(\boldsymbol{\Omega}_{i,k})-\varepsilon}{\|\boldsymbol{\Omega}_{i,k}\|^2}\cdot$$

Thus, to ensure that the error gap $\mathbb{E}_l\{\|\mathbf{x}^{l+1}_i-\mathbf{x}^*_i\|^2\}$ is bounded for all agents, the step size must satisfy

$$0\le\alpha_k\le\min_{i\in\mathcal{N}}\left\{\frac{\lambda_{\min}(\boldsymbol{\Omega}_{i,k})-\varepsilon}{\|\boldsymbol{\Omega}_{i,k}\|^2}\right\} \tag{40}$$

where $0\le\varepsilon\le\lambda_{\min}(\boldsymbol{\Omega}_{i,k})$. Defining

$$\begin{aligned}\boldsymbol{\Delta}&=\big(1+2\alpha_k^2\|\boldsymbol{\Omega}_{i,k}\|^2+2\varepsilon\alpha_k\big)\,\mathbf{I}-2\alpha_k\boldsymbol{\Omega}_{i,k}\\ \Delta_0&=\lambda_{\text{tv}}^2\alpha_k(4\alpha_k+\frac{1}{\varepsilon})(4|\mathcal{R}_i|^2+|\mathcal{B}_i|^2)m\end{aligned}$$

and assuming that α_k satisfies (40), we get $\|\boldsymbol{\Delta}\|\le1$. Now, employing (31), the error gap in (36) turns into

$$\mathbb{E}_l\{\|\mathbf{x}^{l+1}_{i,k}-\mathbf{x}^*_{i,k}\|^2\}\le\|\boldsymbol{\Delta}\|\,\mathbb{E}_l\{\|\mathbf{x}^l_{i,k}-\mathbf{x}^*_{i,k}\|^2\}+\Delta_0, \tag{41}$$

which simplifies as

$$\mathbb{E}_l\{\|\mathbf{x}^{l+1}_{i,k}-\mathbf{x}^*_{i,k}\|^2\}\le\|\boldsymbol{\Delta}\|^{l+1}\,\mathbb{E}_l\{\|\mathbf{x}^0_{i,k}-\mathbf{x}^*_{i,k}\|^2\}+\Delta_0\sum_{s=0}^{l}\|\boldsymbol{\Delta}\|^s\cdot \tag{42}$$

As a result of $\|\boldsymbol{\Delta}\|\le1$, the error gap becomes

$$\lim_{l\to\infty}\mathbb{E}_l\{\|\mathbf{x}^{l+1}_{i,k}-\mathbf{x}^*_{i,k}\|^2\}\le\frac{\Delta_0}{1-\|\boldsymbol{\Delta}\|} \tag{43}$$

asymptotically, which completes the proof. ■

Remark 1: The error gap in (43) illustrates that the BR-DKF restricts the impact of attack amplitude completely due to the $\mathsf{sign}(\cdot)$ terms; however, the number of Byzantine agents in the network still affects the error bound in (43) by altering Δ_0.

Remark 2: This work provides the analysis for an undirected graph topology, and analyzing the algorithm with a directed graph topology requires a new analysis, which is beyond the scope of this work.

[2] The matrix norm $\|\cdot\|$ is defined as $\|\mathbf{A}\|\triangleq\sigma_{\max}(\mathbf{A})$ with $\sigma_{\max}(\cdot)$ representing the largest singular value of the argument matrix.

V. Simulation Results

The performance of the proposed Byzantine-Robust DKF (BR-DKF) is illustrated by considering two network topologies, including a network of $N = 5$ agents with the edge set of $\mathcal{E} = \{(1, 2), (2, 3), (3, 5), (4, 5), (5, 1)\}$, same as [15], shown in Fig. 1, and a randomly generated undirected connected network with $N = 25$ agents with the topology shown in Fig. 6. The discrete-time system and agent parameters are considered similar to the work in [49], and are given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.4 & 0.9 & 0 & 0 \\ -0.9 & 0.4 & 0 & 0 \\ 0 & 0 & 0.5 & 0.8 \\ 0 & 0 & -0.8 & 0.5 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k,$$

$$\mathbf{y}_{i,k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_{i,k},$$

where the state noise covariance $\mathbf{Q} = 0.1\mathbf{I}$, and the observation noise covariance $\mathbf{R}_i = \text{diag}(0.1, 0.2, 0.3, 0.1)$. To benchmark our proposed algorithm, we evaluate the following scenarios: the centralized KF (CKF), distributed KF (DKF) [49], DKF subject to Byzantine attack (B-DKF), and the proposed BR-DKF subject to Byzantine attack.

Considering Byzantines as B agents with the largest node degree in the graph topology, the corresponding perturbation covariances are designed following the optimization problem $\mathcal{P}_1$ in [16]. In problem $\mathcal{P}_1$, the steady-state network mean squared error (NMSE) is maximized by designing the covariance of the perturbation sequences at the Byzantine agents. The NMSE is defined as

$$\text{NMSE} \triangleq \limsup_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} \text{tr}(\mathbf{P}_{i,k}),$$

where $\mathbf{P}_{i,k}$ is the error covariance of the DKF in [16] at agent i and time instant k. Accordingly, the optimization problem to design the perturbation covariances is modeled as

$$\begin{aligned} \max_{\boldsymbol{\Sigma}}.\quad & \text{NMSE} \\ \text{s. t.}\quad & \sum_{j\in\mathcal{B}} \text{tr}(\boldsymbol{\Sigma}_j) \le \eta, \\ & \boldsymbol{\Sigma} \succcurlyeq 0, \end{aligned}$$

where the first constraint limits the total power of the falsification sequences and satisfies the detection-avoidance target with parameter η. The second constraint ensures that the perturbation covariance $\boldsymbol{\Sigma}$ is positive semidefinite. As a result, the proposed algorithm is examined under the worst-case scenario of an attack that maximizes the network MSE.

In the first scenario, we consider the network in Fig. 1 comprising $N = 5$ agents, of which $B = 2$ are Byzantine agents, taken as the agents with the highest node degree. We plot the average MSE across agents, i.e.,

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_k - \hat{\mathbf{x}}_{i,k})^{\mathrm{T}} (\mathbf{x}_k - \hat{\mathbf{x}}_{i,k}). \tag{44}$$

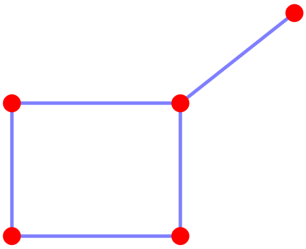


Fig. 1. Network topology with $N = 5$ agents.

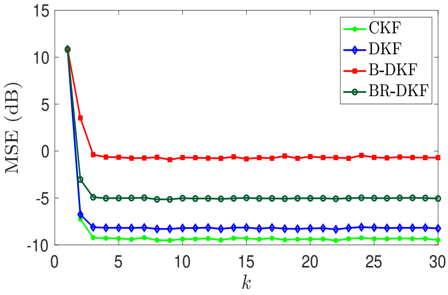


Fig. 2. MSE versus filtering time index k in the network with $N = 5$ agents.

In the absence of Byzantines, the parameters of α_k, γ_k, and λ_{tv} of the BR-DKF are tuned to obtain the nearest possible MSE to the DKF algorithm. Even without a Byzantine attack, the BR-DKF does not reach the same performance as the DKF method; this is because the sign$(\cdot)$ terms in the updating process restrict the actual values of the state estimate. Here, Byzantine agents conduct a coordinated data-falsification attack where $\boldsymbol{\Sigma}_i$ denotes the covariance matrix of perturbation sequences of agent $i \in \mathcal{B}$.

Fig. 2 shows the MSE in (44) versus the filtering time index k in a network of $N = 5$ agents. The number of iterations for the state estimate and the error covariance updates is set to $l^* = 25$ and the results are averaged over 2000 Monte Carlo experiments. The BR-DKF achieves lower MSE than the B-DKF under the same Byzantine attack, which demonstrates its robustness. There is a performance gap between centralized and distributed Kalman filters, even without Byzantine agents, which is due to the number of iterations in the subgradient solution. By increasing the number of l^*, the performance of the DKF will approach the CKF asymptotically.

Fig. 3 shows how the actual state of the network, with $m = 4$, is closely estimated by various filtering methods. Tracking performance for different filtering settings is illustrated in shaded colors for all agents in the network, and the average of the estimate for all agents is shown as a solid line. We see that the proposed BR-DKF method estimates the actual state elements with a smaller variance than the B-DKF method.

Fig. 4 shows the MSE versus the percentage of Byzantine agents in the network. The BR-DKF method is significantly

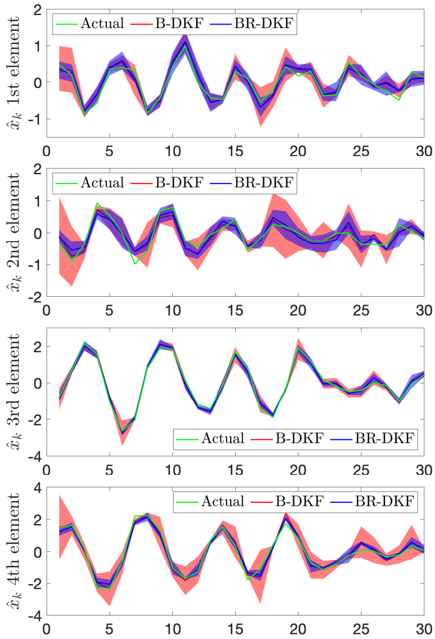


Fig. 3. State estimation accuracy for the different elements of the state in a network of $N = 5$.

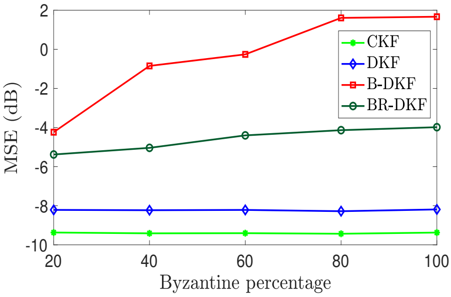


Fig. 4. Steady-state MSE versus percentage of the Byzantine agents in the network with $N = 5$ agents.

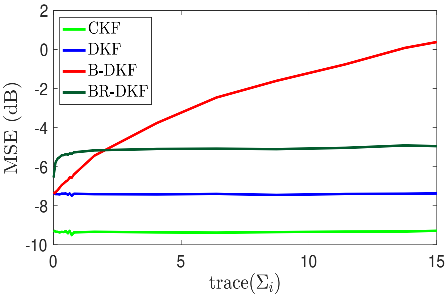


Fig. 5. Steady-state MSE versus trace of the Byzantine agent attack covariance in the network with $N = 5$ agents.

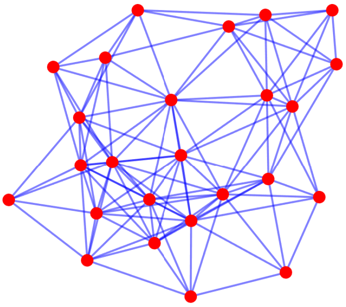


Fig. 6. Network topology with $N = 25$ agents.

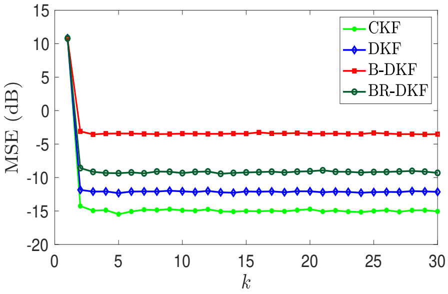


Fig. 7. MSE versus filtering time index k in a network $N = 25$ agents.

less sensitive to the number of Byzantines in the networks than the B-DKF method. Fig. 5 shows the MSE versus the trace of perturbation sequence covariance of individual Byzantine agents. As shown, even without injecting any noise by the Byzantine agent, the MSE in BR-DKF does not reach the DKF method; this is because the sign$(\cdot)$ terms in the updating process limit the actual value of the state estimates. Upon starting the Byzantine attack, the obtained MSE under the B-DKF increases dramatically as more noise is injected, but the obtained MSE under the BR-DKF does not change. This is due to the restriction that the sign$(\cdot)$ term provides, and as stated in *Remark 1*, the number of Byzantine agents is the only factor impacting the steady-state MSE in BR-DKF.

In the second scenario, we consider a network of $N = 25$ agents as in Fig. 6, including $B = 5$ Byzantine agents that are chosen as network agents with the highest node degree. A similar tuning is made to the step size parameters in order to ensure the smallest difference in MSE for DKF and BR-DKF algorithms in the absence of an attack. In Fig. 7, the MSE in (44) is plotted versus the filtering time index k for different filtering approaches. The subgradient solution for the state and error covariance are iterated for $l^* = 25$ iterations. Under the same Byzantine attack, the proposed BR-DKF obtains a lower MSE than the B-DKF, which verifies its robustness against Byzantine behaviors.

Similar to the previous scenario, the estimation accuracy for

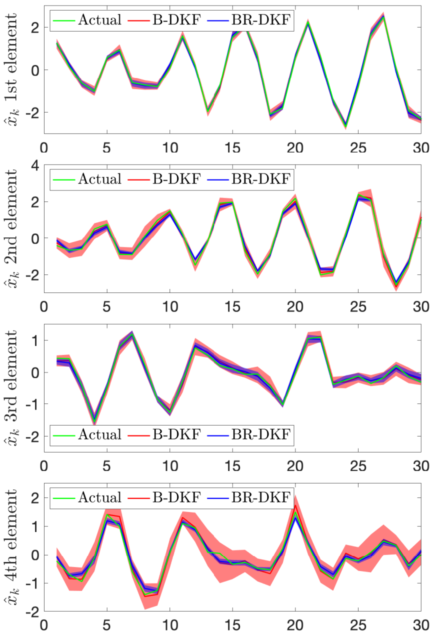


Fig. 8. State estimation accuracy for the different elements of the state in a network of $N = 25$.

different state vector elements, with $m = 4$, is shown in Fig. 8. The estimated values of agents are plotted in shaded colors and their average of the estimated values in solid colors. It can be seen that the proposed BR-DKF reduces the variance of the estimated values and can robustly track the actual state of the network with higher accuracy than the B-DKF algorithm.

Fig. 9 illustrates the obtained MSE versus the percentage of Byzantine agents in the network for different algorithms. A similar trend is observed, showing that the greater the percentage of Byzantine agents in the network, the higher the MSE, while the BR-DKF sensitivity to the Byzantine percentage is significantly less than the B-DKF. In Fig. 10, the MSE is illustrated versus the trace of the perturbation covariance of individual Byzantine agents, which shows that under the BR-DKF, as the trace of attack covariance is low, sign$(\cdot)$ terms in the state estimate update equations constrain the actual values and degrade the MSE compared to the DKF. When Byzantines inject more noise, the performance of the BR-DKF is not degraded, while under the B-DKF algorithm, the MSE increases significantly as more noise is injected. This confirms the resilience of the BR-DKF to the coordinated data falsification attack.

Simulation results are provided for a stable state matrix $\mathbf{F}$, spectral radius less than one, while the algorithm also performs efficiently for unstable state matrices. To verify the stability of the proposed algorithm using an unstable state matrix, in Fig. 11 and Fig. 12, we plot the MSE versus the trace of perturbation covariance for the case where only $\mathbf{F}$ is different and is considered as

$$\mathbf{F} = \begin{bmatrix} 0.6 & 0.9 & 0 & 0 \\ -0.9 & 0.6 & 0 & 0 \\ 0 & 0 & 0.7 & 0.8 \\ 0 & 0 & -0.8 & 0.7 \end{bmatrix}. \tag{45}$$

It can be seen that the trend of changing MSE versus trace of the perturbation covariance in different algorithms remains the same.

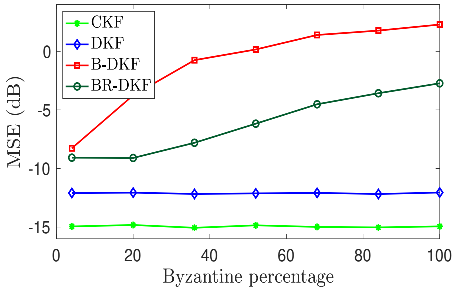


Fig. 9. Steady-state MSE versus percentage of the Byzantine agents in the network with $N = 25$ agents.

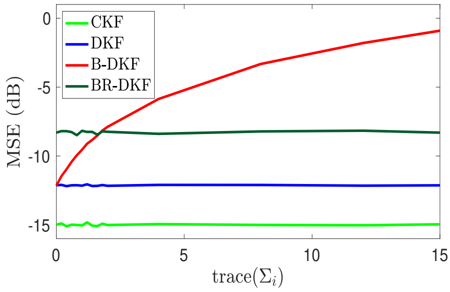


Fig. 10. Steady-state MSE versus trace of the Byzantine agent attack covariance in the network with $N = 25$ agents.

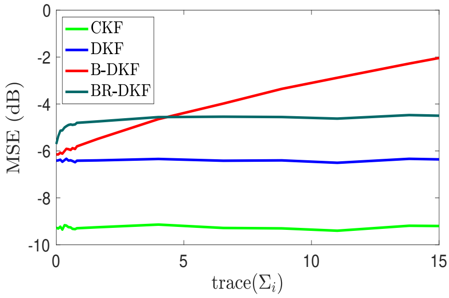


Fig. 11. Steady-state MSE versus trace of the Byzantine agent attack covariance, with unstable state matrix, in the network with $N = 5$ agents.

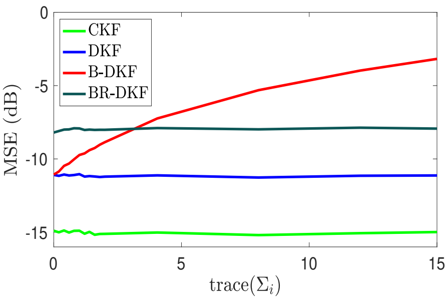


Fig. 12. Steady-state MSE versus trace of the Byzantine agent attack covariance, with unstable state matrix, in the network with $N = 25$ agents.

VI. Conclusion

This paper proposed a distributed Kalman filter (DKF) with resiliency against Byzantine attacks. Considering the Byzantine agent as a network member that alters information before exchanging it with neighbors, we investigated DKF operations from the perspective of distributed optimization. The resulting optimization-based DKF solution improved the robustness of the filtering operations against Byzantine behaviors by employing a TV-norm penalty term for the objective function. We utilized a distributed subgradient algorithm to derive a suboptimal solution to update the state estimate and error covariance matrix of the proposed Byzantine robust DKF (BR-DKF). Furthermore, we demonstrated that the proposed suboptimal solution converges to a neighborhood of the optimal centralized solution with a bounded radius in the presence of the Byzantine agents. Numerical simulations corroborated the theoretical findings and demonstrated the robustness of the proposed BR-DKF against Byzantine behaviors. In future research, the impact of time-varying and directed graph topologies on the performance of the proposed algorithm will be investigated.

References

[1] Y. Liu, B. Wang, W. Ye, X. Ning, and B. Gu, "Global estimation method based on spatial–temporal Kalman filter for dpos," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3748–3756, Feb. 2021.

[2] C. Li and H. Wang, "Distributed frequency estimation over sensor network," *IEEE Sensors J.*, vol. 15, no. 7, pp. 3973–3983, July 2015.

[3] Y. Yu, "Consensus-based distributed linear filter for target tracking with uncertain noise statistics," *IEEE Sensors J.*, vol. 17, no. 15, pp. 4875–4885, Aug. 2017.

[4] Y. Chen, Q. Zhao, Z. An, P. Lv, and L. Zhao, "Distributed multi-target tracking based on the K-MTSCF algorithm in camera networks," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5481–5490, July 2016.

[5] R. Olfati, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. 48th IEEE Conf. Decis. and Control*, 2009, pp. 7036–7042.

[6] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sept. 2010.

[7] D. Kapetanovic, G. Zheng, and F. Rusek, "Physical layer security for massive MIMO: An overview on passive eavesdropping and active attacks," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 21–27, June 2015.

[8] R. K. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, Oct. 2002.

[9] A. Vempaty, L. Tong, and P. K. Varshney, "Distributed inference with byzantine data: State-of-the-art review on data falsification attacks," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 65–75, Sept. 2013.

[10] W. Yang, W. Luo, and X. Zhang, "Distributed secure state estimation under stochastic linear attacks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2036–2047, July-Sept. 1 2021.

[11] W. Yang, Y. Zhang, G. Chen, C. Yang, and L. Shi, "Distributed filtering under false data injection attacks," *Elsevier Automatica*, vol. 102, pp. 34–44, April 2019.

[12] L. An and G.-H. Yang, "Distributed secure state estimation for cyber–physical systems under sensor attacks," *Elsevier Automatica*, vol. 107, pp. 526–538, Sept. 2019.

[13] H. Wu, B. Zhou, and C. Zhang, "Secure distributed estimation against data integrity attacks in internet-of-things systems," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–14, 2021.

[14] H. Song, D. Ding, H. Dong, and Q.-L. Han, "Distributed maximum correntropy filtering for stochastic nonlinear systems under deception attacks," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3733–3744, May 2022.

[15] A. Moradi, N. K. D. Venkategowda, S. P. Talebi, and S. Werner, "Privacy-preserving distributed Kalman filtering," *IEEE Trans. Signal Process.*, vol. 70, pp. 3074–3089, June 2022.

[16] A. Moradi, N. K. Venkategowda, and S. Werner, "Coordinated data-falsification attacks in consensus-based distributed Kalman filtering," in *Proc. 8th IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, 2019, pp. 495–499.

[17] A. Moradi, N. K. Venkategowda, S. P. Talebi, and S. Werner, "Distributed Kalman filtering with privacy against honest-but-curious adversaries," in *Proc. 55th IEEE Asilomar Conf. Signals, Syst., Comput.*, 2021, pp. 790–794.

[18] A. Moradi, N. K. D. Venkategowda, S. P. Talebi, and S. Werner, "Securing the distributed Kalman filter against curious agents," in *Proc. 24th IEEE Int. Conf. Inf. Fusion*, 2021, pp. 1–7.

[19] M. N. Kurt, Y. Yılmaz, and X. Wang, "Distributed quickest detection of cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2015–2030, Aug. 2018.

[20] M. N. Kurt, Y. Yilmaz, and X. Wang, "Real-time detection of hybrid and stealthy cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 498–513, Feb. 2019.

[21] M. Aktukmak, Y. Yilmaz, and I. Uysal, "Sequential attack detection in recommender systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3285–3298, Apr. 2021.

[22] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "Recursive filtering of distributed cyber-physical systems with attack detection," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 10, pp. 6466–6476, Oct. 2021.

[23] C.-Z. Bai, V. Gupta, and F. Pasqualetti, "On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6641–6648, Dec. 2017.

[24] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Control Netw. Syst.*, vol. 4, no. 1, pp. 4–13, March 2016.

[25] Y. Chen, S. Kar, and J. M. Moura, "Optimal attack strategies subject to detection constraints against cyber-physical systems," *IEEE Control Netw. Syst.*, vol. 5, no. 3, pp. 1157–1168, Sept. 2017.

[26] X.-X. Ren, G. Yang, and X.-G. Zhang, "Statistical-based optimal-stealthy attack under stochastic communication protocol: An application to networked pmsm systems," *IEEE Trans. Ind. Electron.*, 2022.

[27] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2618–2624, Sept. 2016.

[28] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, "False data injection on state estimation in power systems—attacks, impacts, and defense: A survey," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 411–423, Apr. 2017.

[29] F. Li and Y. Tang, "False data injection attack for cyber-physical systems with resource constraint," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 729–738, Feb. 2020.

[30] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[31] B. Kailkhura, S. Brahma, and P. K. Varshney, "Data falsification attacks on consensus-based detection systems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 145–158, March 2016.

[32] X. He, X. Ren, H. Sandberg, and K. H. Johansson, “How to secure distributed filters under sensor attacks,” *IEEE Trans. Autom. Control*, vol. 67, no. 6, pp. 2843–2856, 2022.

[33] L. An and G.-H. Yang, “Byzantine-resilient distributed state estimation: A min-switching approach,” *Elsevier Automatica*, vol. 129, p. 109664, July 2021.

[34] Y. Chen, S. Kar, and J. M. Moura, “Resilient distributed estimation: Sensor attacks,” *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3772–3779, Sept. 2018.

[35] Y. Chen, S. Kar, and J. M. F. Moura, “Resilient distributed parameter estimation with heterogeneous data,” *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4918–4933, Oct. 2019.

[36] H. Song, P. Shi, C.-C. Lim, W.-A. Zhang, and L. Yu, “Attack and estimator design for multi-sensor systems with undetectable adversary,” *Elsevier Automatica*, vol. 109, p. 108545, Nov. 2019.

[37] H. Song, P. Shi, W.-A. Zhang, C.-C. Lim, and L. Yu, “Distributed h_∞ estimation in sensor networks with two-channel stochastic attacks,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 465–475, Feb. 2020.

[38] Y. Shi and Y. Wang, “Online secure state estimation of multiagent systems using average consensus,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 5, pp. 3174–3186, May 2022.

[39] J. G. Lee, J. Kim, and H. Shim, “Fully distributed resilient state estimation based on distributed median solver,” *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3935–3942, Sept. 2020.

[40] M. Fauser and P. Zhang, “Resilient homomorphic encryption scheme for cyber-physical systems,” in *Proc. 60th IEEE Conf. Decis. and Control (CDC)*, 2021, pp. 5634–5639.

[41] Y. Ni, J. Wu, L. Li, and L. Shi, “Multi-party dynamic state estimation that preserves data and model privacy,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2288–2299, Jan. 2021.

[42] J. Zhou, W. Ding, and W. Yang, “A secure encoding mechanism against deception attacks on multi-sensor remote state estimation,” *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1959–1969, 2022.

[43] M. Fauser and P. Zhang, “Resilience of cyber-physical systems to covert attacks by exploiting an improved encryption scheme,” in *Proc. 59th IEEE Conf. Decis. and Control (CDC)*, 2020, pp. 5489–5494.

[44] H. Lin, Z. T. Kalbarczyk, and R. K. Iyer, “Raincoat: Randomization of network communication in power grid cyber infrastructure to mislead attackers,” *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4893–4906, Sept. 2019.

[45] A. Mitra and S. Sundaram, “Byzantine-resilient distributed observers for lti systems,” *Elsevier Automatica*, vol. 108, p. 108487, Oct. 2019.

[46] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos, “DETOX: A redundancy-based framework for faster and more robust gradient aggregation,” in *Proc. NIPS*, vol. 32, 2019, p. 10320–10330.

[47] P. Krishnamurthy and F. Khorrami, “Resilient redundancy-based control of cyber–physical systems through adaptive randomized switching,” *Systems & Control Letters*, vol. 158, p. 105066, Dec. 2021.

[48] A. Mitra, F. Ghawash, S. Sundaram, and W. Abbas, “On the impacts of redundancy, diversity, and trust in resilient distributed state estimation,” *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 713–724, June 2021.

[49] K. Ryu and J. Back, “Distributed Kalman-filtering: Distributed optimization viewpoint,” in *Proc. 58th IEEE Conf. Decis. and Control*, 2019, pp. 2640–2645.

[50] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, “Robust distributed consensus using total variation,” *IEEE Trans. Autom. Control*, vol. 61, no. 6, pp. 1550–1564, June 2016.

[51] J. Peng, W. Li, and Q. Ling, “Byzantine-robust decentralized stochastic optimization over static and time-varying networks,” *Elsevier Signal Process.*, vol. 183, p. 108020, June 2021.

[52] J. Peng, W. . Li, and Q. Ling, “Variance reduction-boosted Byzantine robustness in decentralized stochastic optimization,” in *Proc. 47th IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2022, pp. 4283–4287.

[53] Y. Chen, S. Kar, and J. M. Moura, “Optimal attack strategies subject to detection constraints against cyber-physical systems,” *IEEE Control Netw. Syst.*, vol. 5, no. 3, pp. 1157–1168, Sept. 2018.

[54] C.-Z. Bai, V. Gupta, and F. Pasqualetti, “On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds,” *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6641–6648, Dec. 2017.

Ashkan Moradi received the M.Sc. degree in Telecommunication Networks from University of Tehran, Iran, in 2016. He is currently pursuing a Ph.D. degree at the Department of Electronic Systems at the Norwegian University of Science and Technology (NTNU). His expertise and research interests include distributed learning and estimation algorithms in resource-constrained networks, with an emphasis on agent privacy and data security. From June 2022 to Aug. 2022, he was visiting researcher at the Technical University of Munich in Germany.

Naveen K. D. Venkategowda (S’12–M’17) received the B.E. degree in electronics and communication engineering from Bangalore University, Bengaluru, India, in 2008, and the Ph.D. degree in electrical engineering from Indian Institute of Technology, Kanpur, India, in 2016. He is currently an Universitetslektor at the Department of Science and Technology, Linköping University, Sweden. From Oct. 2017 to Feb. 2021, he was postdoctoral researcher at the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway. He was a Research Professor at the School of Electrical Engineering, Korea University, South Korea from Aug. 2016 to Sep. 2017. He was a recipient of the TCS Research Fellowship (2011-15) from TCS for graduate studies in computing sciences and the ERCIM Alain Bensoussan Fellowship in 2017.

Stefan Werner (F’23) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998, and the D.Sc. degree (Hons.) in electrical engineering from the Signal Processing Laboratory, Helsinki University of Technology, Espoo, Finland, in 2002. He is currently a Professor at the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Director of IoT@NTNU, and Adjunct Professor at Aalto University in Finland. He was a visiting Melchor Professor with the University of Notre Dame during the summer of 2019 and an Adjunct Senior Research Fellow with the Institute for Telecommunications Research, University of South Australia, from 2014 to 2020. He held an Academy Research Fellowship, funded by the Academy of Finland, from 2009 to 2014. His research interests include adaptive and statistical signal processing, wireless communications, and security and privacy in cyber-physical systems. He is a member of the editorial boards for the EURASIP Journal of Signal Processing and the IEEE Transactions on Signal and Information Processing over Networks. Dr. Werner is a Fellow of the IEEE.

ISBN 978-82-326-6586-0 (printed ver.)
ISBN 978-82-326-5501-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology