# 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 $\chi^2$ 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 $i \in \mathcal{N}$

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

6: $\mathbf{M}^{-1}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \mathbf{\Gamma}_{j,n}\}$

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

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

9: $\hat{\mathbf{x}}_{i,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\forall j \in \mathcal{N}_i : \boldsymbol{r}_{j,n}\}$

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

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

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 $\mathbf{\Gamma}_{i,n}$, 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:

- 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.

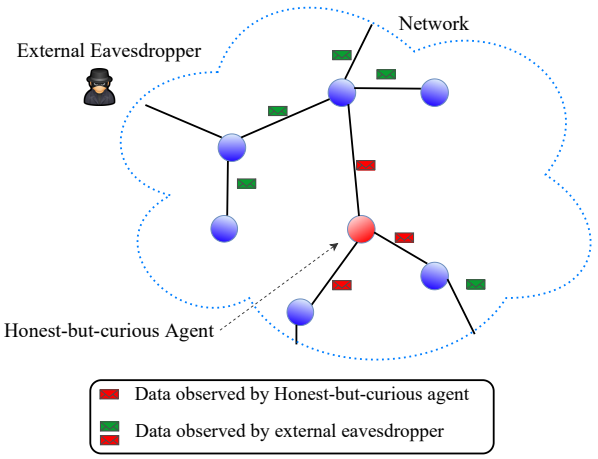


Fig. 1. Illustration of information accessible to external eavesdroppers and HBC agents.

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

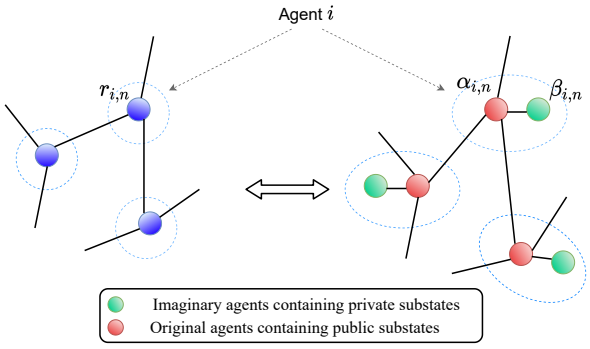


Fig. 2. State decomposition representation of $\boldsymbol{r}_{i,n}$ to public substate $\boldsymbol{\alpha}_{i,n}$ and private substate $\boldsymbol{\beta}_{i,n}$.

information of agent $i$, at time instant $n$, denoted by $\boldsymbol{\Gamma}_{i,n}$, 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 $i$th 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:

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

where $\varepsilon$ 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.,

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

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

**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: $\boldsymbol{\Gamma}_{i,n}=\mathbf{M}_{i,n|n-1}^{-1}+N\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}\mathbf{H}_i$
6: $\mathbf{M}_{i,n|n}^{-1}\leftarrow\boxed{\text{ACF}}\leftarrow\{\forall j\in\mathcal{N}_i:\boldsymbol{\Gamma}_{j,n}\}$
7: $\mathbf{G}_{i,n}=N\mathbf{M}_{i,n|n}\mathbf{H}_i^{\mathrm{T}}\mathbf{C}_{\mathbf{w}_{i,n}}^{-1}$
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: $\hat{\mathbf{x}}_{i,n|n}=\boldsymbol{\alpha}_{i,n}(K)$

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

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:

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

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

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

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

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

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

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

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} = \mathbf{G}^k \boldsymbol{\mathcal{E}}_n + \phi^{k-1} \boldsymbol{\mathcal{B}} \boldsymbol{\nu}(k-1) \\ + \sum_{s=2}^{k} \phi^{k-s} \left( \mathbf{G}^{s-1} - \mathbf{G}^{s-2} \right) \boldsymbol{\mathcal{B}} \boldsymbol{\nu}(k-s)\end{aligned} \tag{17}$$

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

$$\begin{aligned}\boldsymbol{\mathcal{P}}_k &= \mathbf{G}^k \mathsf{Blockdiag}(\{\mathbf{P}_i \mathbf{A}\}_{i=1}^{2N}) \\ \boldsymbol{\mathcal{Q}}_k &= \mathbf{G}^k \mathsf{Blockdiag}(\{\mathbf{P}_i\}_{i=1}^{2N}) \\ \boldsymbol{\Omega}_{n,k} &= \mathbf{G}^k \mathsf{Blockdiag}(\{\mathbf{Q}_i\}_{i=1}^{2N}) [\mathbf{w}_{1,n}^{\mathrm{T}}, \cdots, \mathbf{w}_{2N,n}^{\mathrm{T}}]^{\mathrm{T}}\end{aligned}$$

with $\mathbf{P}_i = \mathbf{I}_m - N \mathbf{M}_i \mathbf{H}_i^{\mathrm{T}} \mathbf{C}_{\mathbf{w}_i}^{-1} \mathbf{H}_i$ 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

$$\mathcal{E}_{j,n}(k) \triangleq \mathrm{tr}\left( \mathbb{E}\{ (\boldsymbol{r}_{j,n} - \hat{\boldsymbol{r}}_{j,n}(k)) (\boldsymbol{r}_{j,n} - \hat{\boldsymbol{r}}_{j,n}(k))^{\mathrm{T}} \} \right). \tag{22}$$

### 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*

$$\mathcal{E}_j = \lim_{n\to\infty}\lim_{k\to\infty}\mathcal{E}_{j,n}(k) = \mathit{tr}\Big((\mathbf{e}_j^T\otimes\mathbf{I}_m)\,\tilde{\boldsymbol{\mathcal{L}}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{\mathcal{L}}}^T(\mathbf{e}_j\otimes\mathbf{I}_m)\Big) \tag{24}$$

*where $\mathbf{e}_j \in \mathbb{R}^N$ is a vector with 1 in the $j$th 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 $j$th 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 $k$th 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

$$\mathbf{C}_\alpha = \Big[\mathbf{e}_{j_1},\mathbf{e}_{j_2},\cdots,\mathbf{e}_{j_{N_N}},\mathbf{e}_N\Big]^{\mathrm{T}}\otimes\mathbf{I}_m\in\mathbb{R}^{(N_N+1)m\times Nm},$$

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 $j$th 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

$$\sum_{t=0}^{k}\mathbf{y}_n(t) =\mathbf{C}(\mathbf{I}_{2Nm}-\mathbf{G})^{k+1}(\mathbf{I}_{2Nm}-\mathbf{G})^{-1}\mathbf{z}_n(0) + \mathbf{C}_\alpha\left(\sum_{t=0}^{k-1}\phi^t\boldsymbol{\mathcal{C}}_{k-1-t}\mathbf{B}\boldsymbol{\nu}(t)+\phi^k\boldsymbol{\nu}(k)\right).$$

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

$$\hat{\mathbf{F}}_k =\frac{\varepsilon}{\phi^{k+1}}\mathbf{C}_\alpha\left(\boldsymbol{\Theta}_{1:Nm}\bar{\boldsymbol{\Lambda}}^k\boldsymbol{\Theta}_{1:Nm}^{\mathrm{T}}\right)(\mathbf{W}\otimes\mathbf{I}_m) + \frac{\varepsilon}{2N\phi^{k+1}}\left(\mathbf{1}_{(N_N+1)}[d_1,d_2,\ldots,d_N]\right)\otimes\mathbf{I}_m \tag{33}$$

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,*

$$\bar{\mathbf{P}}_n(k)=\mathbb{E}_{\mathbf{U}}\{\varepsilon^2\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\tilde{\boldsymbol{\Sigma}}_n\boldsymbol{\Delta}_{\mathbf{H}}^T(k)(\mathbf{H}^\dagger(k))^T + \sigma^2(\mathbf{I}-\varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))\mathbf{H}^\dagger(k)\mathbf{F}(k)\mathbf{F}^T(k)(\mathbf{H}^\dagger(k))^T (\mathbf{I}-\varepsilon\mathbf{H}^\dagger(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))^T\} \tag{35}$$

*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 $\lim_{n\to\infty}\mathsf{tr}\left(\mathbb{E}\{\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\}\right)=\infty$. Therefore, a lower bound of the privacy leakage at agent $j$ after $k$ consensus iterations is given by

$$\bar{\mathcal{E}}_j(k)=\mathsf{tr}\left((\mathbf{e}_j^{\mathrm{T}}\otimes\mathbf{I}_m)\mathbf{P}(k)(\mathbf{e}_j\otimes\mathbf{I}_m)\right) \tag{36}$$

where $\mathbf{e}_j\in\mathbb{R}^N$ is a vector with 1 in the $j$th 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

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

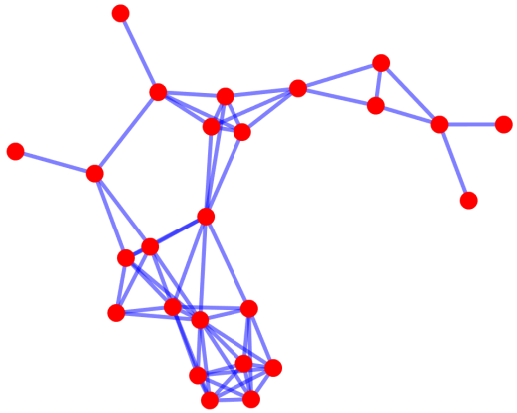

Fig. 3. Network topology with $N = 25$ agents.

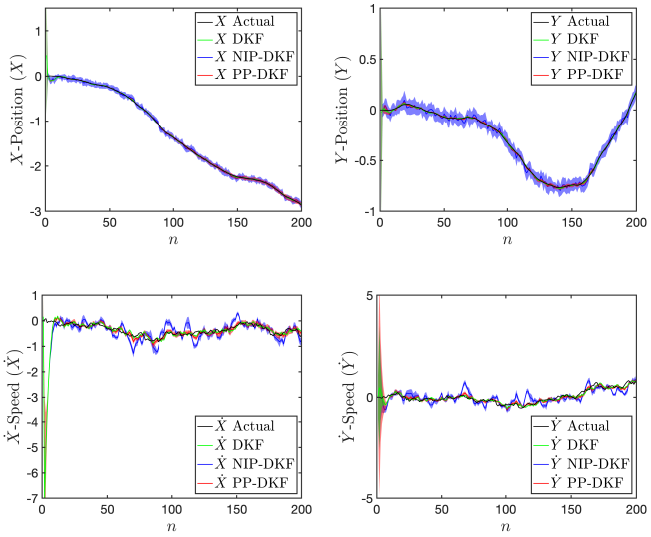


Fig. 4. Tracking performance of distributed Kalman filtering settings for each $N = 25$ agents (shaded color) and their average as a solid line with $K = 30$ consensus iterations and noise variance $\sigma^2 = 4$.

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 $\eta = 0.4$.

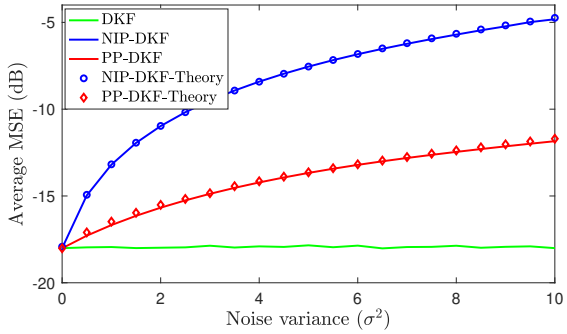


Fig. 5. Average MSE of the filtering process versus noise variance $\sigma^2$ for both theory and simulation with $K = 30$ consensus iterations.

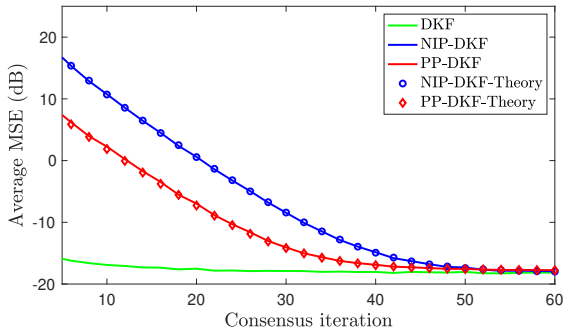


Fig. 6. The overall filtering average MSE versus the number of consensus iteration with noise variance $\sigma^2 = 4$.

### 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 $\sigma^2$. 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.

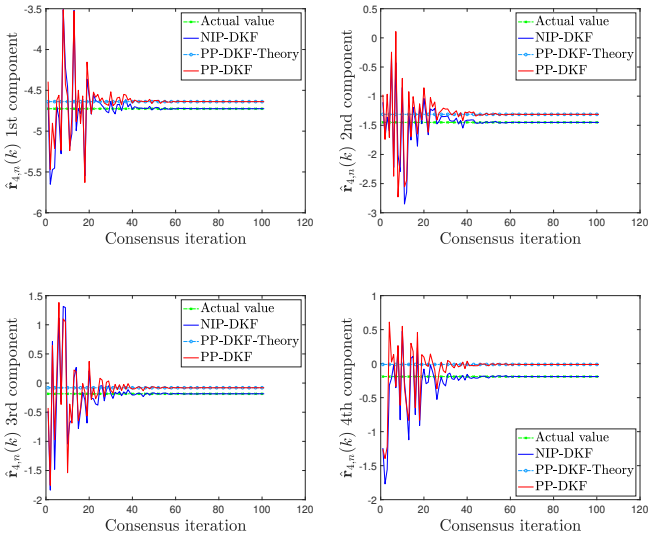


Fig. 7. The observer of the external eavesdropper to estimate all components of the initial state $\boldsymbol{r}_{4,n}(0)$, i.e., $\hat{\boldsymbol{r}}_{4,n}(k)$, given the noise variance $\sigma^2 = 4$.

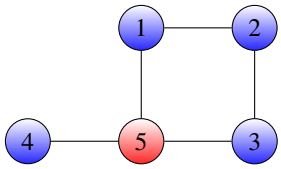


Fig. 8. Network topology with $N = 5$ agents.

### 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))

$$\hat{\boldsymbol{r}}_n(k+1) = \hat{\boldsymbol{r}}_n(k) + \tilde{\boldsymbol{r}}_n(k+1) - (\mathbf{Q} \otimes \mathbf{I}_m)\, \tilde{\boldsymbol{r}}_n(k) \quad (39)$$

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

$$\hat{\boldsymbol{r}}_n(k+1) = \boldsymbol{r}_n(0) + \phi^{k+1}\boldsymbol{\nu}(k+1) \quad (40)$$

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.

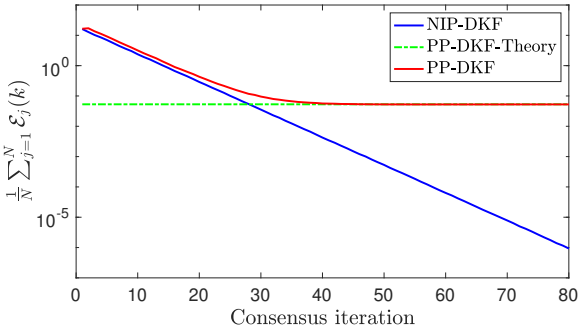


Fig. 9. Average privacy $\frac{1}{N}\sum_{j=1}^{N}\mathcal{E}_j(k)$ versus the number of consensus iterations in the presence of the external eavesdropper.

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 $\eta$. 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 $\sigma^2$. 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 $\sigma^2$. 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

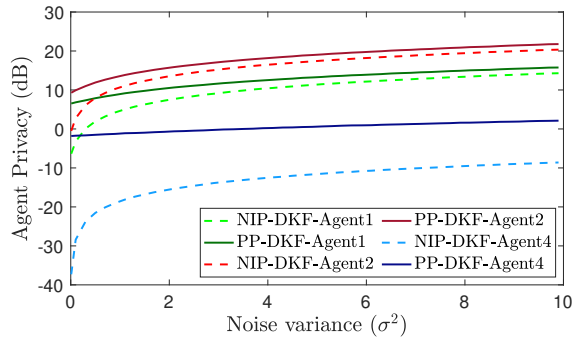


Fig. 10. Agent privacy versus noise variance ($\sigma^2$), given $K = 30$ consensus iterations. Due to the symmetric topology, agents 1 and 3 achieve same privacy level and only the result of the 1st agent is shown in the figure.

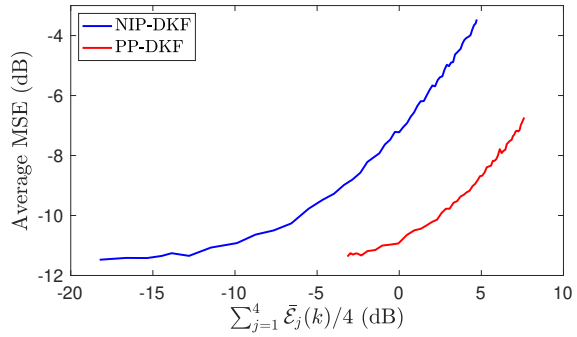


Fig. 11. The trade-off between Kalman filtering accuracy and average privacy $\sum_{j=1}^{4} \bar{\mathcal{E}}_j(k)/4$ for different values of the injected noise variance $\sigma^2$.

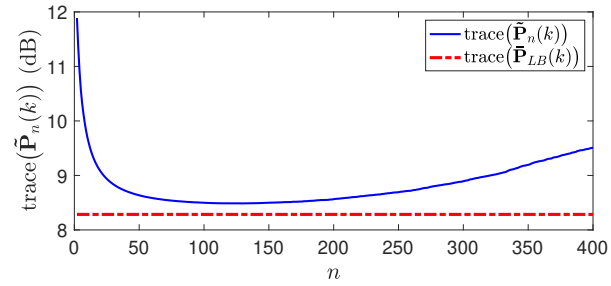


Fig. 12. The mean squared estimation error at the HBC agent after $K = 30$ consensus iterations versus filtering time instant $n$.

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., $\tilde{\mathbf{P}}_{LB}(k)$, 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

$$\mathbf{z}(k+1) = \mathbf{G}^{k+1}\mathbf{z}(0). \tag{44}$$

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 $j$

$$\begin{aligned}\hat{\boldsymbol{r}}_{j,n}(k+1) = \hat{\boldsymbol{r}}_{j,n}(k) + \tilde{\boldsymbol{\alpha}}_{j,n}(k+1) \\ - \left(\tilde{\boldsymbol{\alpha}}_{j,n}(k) + \varepsilon \sum_{l\in\mathcal{N}_j} w_{jl}\left(\tilde{\boldsymbol{\alpha}}_{l,n}(k) - \tilde{\boldsymbol{\alpha}}_{j,n}(k)\right)\right)\end{aligned} \tag{45}$$

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

$$\begin{aligned}\hat{\boldsymbol{r}}_n(k+1) = \hat{\boldsymbol{r}}_n(k) + \boldsymbol{\alpha}_n(k+1) + \boldsymbol{\omega}(k+1) \\ - \left((\mathbf{I}_N - \varepsilon(\mathbf{D}-\mathbf{W})) \otimes \mathbf{I}_m\right)\left(\boldsymbol{\alpha}_n(k) + \boldsymbol{\omega}(k)\right).\end{aligned} \tag{47}$$

Considering the network-wide substate update equations in (10), i.e.,

$$\boldsymbol{\alpha}_n(k+1) = \mathbf{M}\boldsymbol{\alpha}_n(k) + \varepsilon\mathbf{U}\boldsymbol{\beta}_n(k) + \varepsilon(\mathbf{W}\otimes\mathbf{I}_m)\boldsymbol{\omega}(k) \tag{48}$$

$$\boldsymbol{\beta}_n(k+1) = \varepsilon\mathbf{U}\boldsymbol{\alpha}_n(k) + \left(\mathbf{I}_{Nm} - \varepsilon\mathbf{U}\right)\boldsymbol{\beta}_n(k) \tag{49}$$

where $\mathbf{M} = \left(\mathbf{I}_N - \varepsilon(\mathbf{D}-\mathbf{W})\right) \otimes \mathbf{I}_m - \varepsilon\mathbf{U}$, 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

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

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

$$\begin{aligned}\bar{\mathbf{P}}_{\mathrm{n}}(k) = \mathbb{E}_{\mathbf{U}}\{&\varepsilon^2 \mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k)\tilde{\boldsymbol{\Sigma}}_n \boldsymbol{\Delta}_{\mathbf{H}}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}} \\ &+ \sigma^2(\mathbf{I} - \varepsilon\mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))\mathbf{H}^{\dagger}(k)\mathbf{F}(k)\mathbf{F}^{\mathrm{T}}(k)(\mathbf{H}^{\dagger}(k))^{\mathrm{T}} \\ &(\mathbf{I} - \varepsilon\mathbf{H}^{\dagger}(k)\boldsymbol{\Delta}_{\mathbf{H}}(k))^{\mathrm{T}}\}.\end{aligned} \tag{70}$$

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 $j$th entry and zeros elsewhere. The mean squared estimation error $\tilde{\mathcal{E}}_j(k)$ for $j = 1, 2, \cdots, 2N$ is lower-bounded as

$$\tilde{\mathcal{E}}_j(k) = \mathsf{tr}\left((\tilde{\mathbf{e}}_j \otimes \mathbf{I}_m)(\tilde{\mathbf{e}}_j^{\mathrm{T}} \otimes \mathbf{I}_m)\tilde{\mathbf{P}}(k)\right) > \lambda_{\min} m$$

where $\lambda_{\min}$ 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 $\lambda_{\min} > 0$. 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.