Privacy-Preserving Adaptive Resilient Consensus for Multiagent Systems Under Cyberattacks

Chenduo Ying, Ning Zheng, Yiming Wu, Ming Xu, and Wen-An Zhang, *Member, IEEE*

***Abstract*—This article investigates the secure and privacy-preserving consensus problem of multiagent systems (MASs) with directed interaction topologies under multiple cyberattacks, which contain deception attacks and DoS attacks. First, a unified attack model is introduced to characterize such a multiple attack phenomenon. Besides, considering the existence of eavesdroppers who can intercept the data transmitted on the links, a fully distributed agent value reconstruction method based on the idea of state decomposition is designed to prevent the leakage of the agent's initial information. Then, a novel privacy-preserving adaptive resilient consensus algorithm (PPARCA) with certain graph robustness condition for MASs under the multiple cyberattacks is proposed. The algorithm adaptively takes different countermeasures in the face of different cyberattacks. PPARCA uses the reconstructed agents' states and combines with the modified secure acceptance and broadcast algorithm (SABA). Theoretical analysis shows that the proposed algorithm can effectively protect the privacy of the initial state of the agents, and reach resilient consensus in the face of cyberattacks. Finally, numerical simulations and Raspberry Pi MASs practical application experiments demonstrate the effectiveness of the proposed results.**

***Index Terms*—Cyberattacks, directed graph, multiagent systems (MASs), privacy preservation, resilient consensus.**

I. Introduction

RECENTLY, distributed average consensus algorithms in multiagent systems (MASs) have attracted much attention [1], where agents seek to cooperatively reach the average of their initial states only through information interaction between adjacent agents. Due to the inherent robustness, flexibility, and scalability, average consensus has been applied in a variety of areas, such as cooperative formation control, intelligent transport systems, sensor networks, smart grid, etc. [2], [3], [4], [5].

However, because of the simple structure of one single agent and real-time information exchange requirement among agents deployed in open environments, MASs are more susceptible to cyberattacks targeting both physical agents and communication links. These negative impact of cyberattacks can affect the existing average consensus performance dramatically, and hence the secure and resilient consensus control for MASs have attracted increasing interest [6], [7], [8], [9], [10], [11], [12]. There are two typical kinds of attacks, namely, denial-of-service (DoS) attacks and deception attacks, studied in the existing resilient consensus works. DoS attacks destroy the data availability of a multiagent network by inducing packet losses or other methods, and can be launched even without knowing the details of the system [6], [7], [10], [11]. In [6], the distributed event-triggered collaboration for linear MASs under unscheduled DoS attacks was investigated, and the authors proposed a secure event-based control scheme to achieve resilient consensus. The leader–follower consensus of MASs with DoS attacks was studied in [10], where the output consensus is guaranteed if the communication graph is connected and the maximal attack duration is bounded. Under the self-triggered mechanism design framework proposed in [11], the resilient output synchronization problem of heterogeneous MASs suffering from DoS attacks was solved. The quantitative relationship between the nonperiodic DoS attack attributes and synchronization was also analyzed.

Compared with DoS attacks, deception attacks are more difficult to detect and seriously affect the integrity of the data [9]. The resilient consensus protocol design algorithm under deception attacks was introduced in [8], where the trusted edge and graph robustness were considered simultaneously. In [9], the secure synchronization algorithm of MASs with deception attacks was studied, where a distributed impulsive controller using a pinning strategy was introduced. One implementation of the deception attacks is the presence of misbehaving agents. A class of general consensus algorithms for computing auxiliary point was proposed in [12], which implements multidimensional consensus in a networked system under this form of deception attacks. Recently, a fully distributed averaging consensus algorithm,

Manuscript received 4 January 2023; revised 9 April 2023; accepted 13 May 2023. Date of publication 26 May 2023; date of current version 19 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61803135, Grant 62073109, and Grant 62173305, and in part by the Zhejiang Provincial Public Welfare Research Project of China under Grant LGF21F020011. Paper no. TII-23-0027. *(Corresponding author: Yiming Wu.)*

Chenduo Ying, Ning Zheng, Yiming Wu, and Ming Xu are with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: cdying@hdu.edu.cn; nzheng@hdu.edu.cn; ymwu@hdu.edu.cn; mxu@hdu.edu.cn).

Wen-An Zhang is with the Department of Automation, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: wazhg@hotmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3280318.

Digital Object Identifier 10.1109/TII.2023.3280318

named secure acceptance and broadcasting algorithm (SABA), has been proposed for MASs under deception attacks in [13]. The core of SABA is the introduction of a storage vector for each agent to hold the initial state about other agents and itself. By ensuring certain network robustness, a majority decision mechanism and a resilient distributed retrieval process are used to finally achieve resilient consensus. In addition to some of the related work mentioned above, the reader can refer to the latest surveys [14] and [15] for more recent works on resilient consensus.

Beside cyberattacks, MASs are also subject to the privacy leakage of agent states in the network. Sometimes the participating agents in the network may not want their own states to be fully known by adjacent agents in the collaborative execution of tasks. Huang et al. [16] first introduced the notion of the differential privacy into the average consensus study of MASs, and proposed a privacy-preserving iterative consensus method. Then, relevant problems including distributed optimization [17] and distribution estimation [18] have been further studied in recent years. The basic idea of such methods is to mask the nodes' true state values with zero-sum stochastic noises during the exchange of messages, which ensures that not only the individual agents' states can be preserved but the consensus is reached. However, such methods will inevitably lead to the problem in inaccuracy of system convergence, i.e., the nodes in the network unavoidably converge to a common value approximation of the initial state average with an error.

To improve resilience to disclosure attacks, some researchers have proposed cryptography-based methods. The most known privacy-preserving consensus implemented in cryptography is homomorphic encryption [19] and secure multiparty computation [20], [21]. Unfortunately, such methods have high computational complexity and may not be applicable to the current distributed MASs with limited resources. Recently, the advent of the state decomposition method offers a new viewpoint on how information should be protected in MASs [22]. This method enables the system to achieve an accurate average consensus and protects the initial state information of each node from being leaked and, at the same time, unlike cryptography-based methods, no additional computational burden is required. However, this method cannot be implemented in a completely distributed manner and is only effective for undirected networks.

Up to now, MASs attack tolerance and privacy preservation are addressed as two separate problems in most of the existing works. Nevertheless, in the actual scenarios, cyberattack and privacy eavesdropping can often be carried out at the same time. The attackers can even launch more targeted attack strategies by eavesdropping the communications among agents. To the best of our knowledge, few articles have been reported on considering privacy-preserving and resilient consensus control simultaneously. Fiore and Russo [23] attempted to consider the problems of privacy protection and consensus control under attacks, but the type of attack is relatively simple, and the impact of attacks on system convergence accuracy cannot be eliminated.

In view of the aforementioned problems, this article focuses on the problems of resilient average consensus and codesign of agent initial value privacy, thus, improving and complementing the existing research results. The main contributions of this article can be summarized as follows.

1) A novel unified attack model is established for the characteristics of the multiple cyberattacks in large-scale distributed MASs.
2) Different from the original state decomposition method in [22], a simplified but fully distributed privacy-preserving mechanism is proposed, where the assumptions of the preset parameter ε and undirected graph in [22] are removed.
3) On the basis of the new attack model and the specific network robustness condition, an adaptive resilient average consensus algorithm based on SABA for MASs is designed. It makes the system tolerate a fraction of nodes to be corrupted by multiple attacks, and finally, achieve accurate average consensus while achieving node initial state privacy-preserving.

The rest of this article is organized as follows. In Section II, we present preliminaries on graph theory together with network robustness, state decomposition method, and establish cyberattacks model, then formulate the problem. Privacy-preserving mechanism and adaptive resilient average consensus algorithm under cyberattacks, as well as some theorems, are shown in Section III. We give some simulation results in Section IV to illustrate and verify the main results presented in this article. Finally, Section V concludes this article.

II. Preliminaries and Problem Formulation

A. Graph Theory and Network Robustness

A directed graph (or digraph) with N nodes can be represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathrm{A})$. The node set and edge set are represented as $\mathcal{V} = \{1, 2, \ldots, N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, respectively. The connectivity between two nodes is denoted by the weight matrix $\mathrm{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$. If $(j, i) \in \mathcal{E}$, it indicates that there is an information interaction from node j to node i and $a_{ij} > 0$, if not then $a_{ij} = 0$. Meanwhile, this article does not consider the node self-loop case, i.e., $a_{ii} = 0$. $\mathcal{N}_i^- = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$ and $\mathcal{N}_i^+ = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$ represent the set of incoming and outgoing neighbor nodes of node i, respectively. The number of incoming data of node i is denoted $d_i = |\mathcal{N}_i^-|$.

In addition to the above knowledge of graph theory, we will use several concepts collectively known as *network robustness*, which was introduced in [24].

Definition 1. [24] (p-fraction reachable set): For a digraph $\mathcal{G}$, a nonempty subset $\mathcal{S}$ of the node set $\mathcal{V}$ is said to be p-fraction reachable set if $\exists i \in \mathcal{S}$ such that $|\mathcal{N}_i^-| > 0$ and $|\mathcal{N}_i^- \setminus \mathcal{S}| \geq p|\mathcal{N}_i^-|$, where $0 \leq p \leq 1$.

Definition 2. [24] (p-fraction robust graph): A nonempty, nontrivial digraph $\mathcal{G}$ on N nodes ($N \geq 2$) is p-fraction robust, with $0 \leq p \leq 1$, if for every pair of nonempty, disjoint subsets of $\mathcal{V}$, at least one of the subsets satisfies p-fraction reachable.

Moreover, there is a variant of robust graphs called strongly r-robust graph which introduced and used in [13] and [25]. Similarly, we will first propose a variant of the p-fraction robust graph and use it in the following.

Definition 3. (strongly p-fraction robust graph): A digraph $\mathcal{G}$ is strongly p-fraction robust if for any nonempty subset $\mathcal{S} \subseteq \mathcal{V}$,

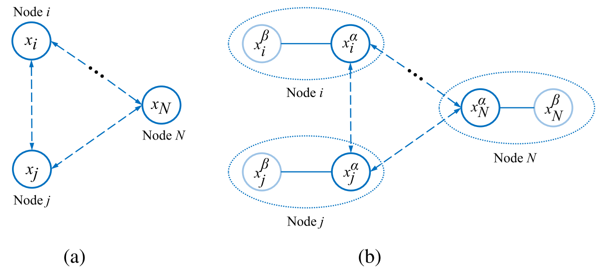

Fig. 1. Illustration of state decomposition method. (a) Before state decomposition. (b) After state decomposition.

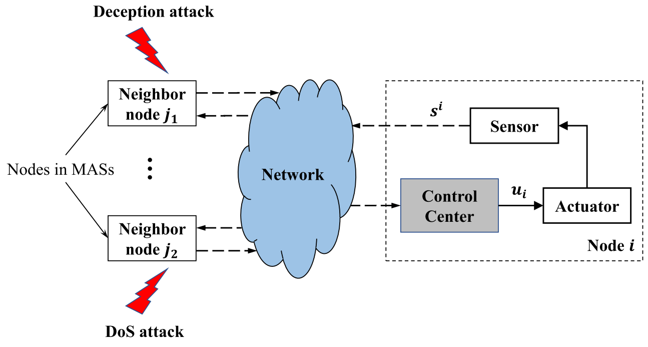

Fig. 2. Cyberattacks in MASs.

either $\mathcal{S}$ is p-fraction reachable set or $\exists i \in \mathcal{S}$ such that $|\mathcal{V} \setminus \mathcal{S}| \subseteq \mathcal{N}_i^-$.

B. State Decomposition Method

The state decomposition method was proposed in [22] as a noise-free privacy-preserving method, which has been widely used in the literature of privacy-preserving average consensus [26], [27]. The main idea of this method is to decompose the initial state x_i of each node into two random substates, which are denoted by x_i^{α} and x_i^{β}, respectively. Specifically, the substate values $x_i^{\alpha}[0]$ and $x_i^{\beta}[0]$ of the initial state of a node can be taken as any real number provided that $x_i^{\alpha}[0] + x_i^{\beta}[0] = 2x_i[0]$ is satisfied. The substate x_i^{α} play the role of the original node before decomposition to participate in the information interaction between neighboring nodes. While the other substate x_i^{β} is hidden, and it does not participate in the information interaction between neighbors, and only communicates with x_i^{α}.

To facilitate the understanding of the state decomposition method and the subsequent work, we take the abstract topological graph of N nodes as an example to illustrate the main idea of the state decomposition method. As illustrated by the change from Fig. 1(a) to (b), the node x_i is processed by the state decomposition method to generate two substates, namely, x_i^{α} and x_i^{β}. The first substate x_i^{α} serves the same purpose as the original node, i.e., it receives and broadcasts information about the state values of the neighboring nodes and the node itself, and x_i^{α} is uniquely visible to the neighboring nodes of node x_i. The second substate x_i^{β} serves to interact with the first substate x_i^{α}, but x_i^{β} is not visible to node x_i's neighbors. Noted that the coupling weights between the two substates x_i^{α} and x_i^{β} are symmetric and private, denoted as $a_{i,\alpha\beta}$.

C. Cyberattack Model

In this article, we consider two common forms of cyberattacks: DoS attack and deception attack. Note that they both attack directly against the agent rather than the communication link. The DoS attack is launched by an adversary outside the MASs and is achieved by, for example, a UDP flood attack to overwhelm the target agent. Then the target agent becomes compromised and no longer responds to legitimate requests to send and receive data. The impact of a DoS attack causes neighbors of the compromised agent to receive null data packets about this agent. Deception attack is also generated by adversary outside the MASs, who hijack traffic packets from the target node and tamper with the data in them before sending them out. Deception attacks can cause the agent to receive data with corrupt integrity. The adversary's attack on the target node can be either continuous or selective, depending on the strategy of the adversary. If no countermeasures are taken, then the desired consensus of the MASs will not be reached. Schematic of MASs under cyberattacks is described in Fig. 2, where s^i is the permanent storage vector maintained by each agent and will be discussed later. It is worth mentioning that for an individual agent in MASs we consider that at some time it may suffer from one of the DoS attacks or deception attacks, but not both. For MASs as a whole, some agents may suffer from DoS attacks and some agents may suffer from deception attacks at the same time. We divide the agents in the system into two parts, the set of normal agents is represented by $\mathcal{V}_n$ and the set of compromised agents is represented by $\mathcal{V}_c$.

It is straightforward to be able to realize that consensus is impossible to achieve when the vast majority of agents in MASs are subjected to cyberattacks. Consequently, it is usually assumed that attackers have limited capabilities. Specifically, by assuming an upper bound f on the total number of compromised nodes in MASs, or an upper bound f on the number of compromised nodes in the neighbor set for each node, they are modeled as f-total attack model [24] and f-local attack model [25], respectively.

Note that both the f-total attack model and the f-local attack model fix an upper limit f on the number, which is only applicable to small-scale systems. The connectivity of each node in a small-scale system is close. Unfortunately, this is not realistic in distributed MASs. Current MASs tend to be large-scale, and the number of agents in the system increases dramatically. Thus, the connectivity of each agent may range from very small to very large. Correspondingly, the number of normal agents affected by an attacked agent can range from small to large. It is imprecise and inappropriate to use a fixed number of compromised agents to represent the situation faced by normal agents, not to mention the whole system. To describe large-scale distributed MASs that match real-world scenarios, we established a new realistic model of cyberattacks which considers the proportion of compromised nodes in the neighborhood of each normal node. The model is defined as follows.

Definition 4. (f-fraction local cyberattacks model): An MAS is under f-fraction local cyberattacks if it contains at most a fraction f of agents subjected to cyberattacks in the neighbors of each normal agent.

Remark 1: Cyberattacks in realistic environments are usually multiple types of parallel attacks [14], [15]. Therefore, in our attack model, we unify dos attack and deception attack as cyberattacks to enhance the generality and realism of the model. Although different types of attacks are generalized in the model, DoS attack and deception attack are addressed adaptively and separately in the design of the resilient consensus algorithm below.

D. Problem Formulation

Consider a first-order discrete-time MAS consisting of N agents, where the agents in the system follow the dynamics as:

$$x_i[k] = \begin{cases} \theta_i[k] & k = 1 \\ \varepsilon_i x_i[k-1] + (1-\varepsilon_i)u_i[k] & k > 1 \end{cases} \tag{1}$$

where $x_i[k]$ denotes the state value of the agent i at time k, θ_i is the privacy protection state value to be designed below, ε_i is the control gain, and u_i is the control input also to be designed below.

In this article, our objective consists of three parts. First, to propose a simplified but fully distributed privacy-preserving mechanism applicable to directed graphs to address the problem of privacy leakage of the initial state value of nodes in MASs. The definition of node initial value privacy is given here, taken from [22].

Definition 5. [22] (privacy of node initial value): The privacy of the initial value $x_i[0]$ of node i is preserved if an eavesdropper cannot estimate the value of $x_i[0]$ with any guaranteed accuracy.

Second, to propose an adaptive resilient average consensus algorithm for MASs under f-fraction local cyberattack model. The nodes in the system can adaptively deal with the problems caused by different types of cyberattacks. The resilient average consensus is defined as follows.

Definition 6. (resilient average consensus): An MAS over the graph $\mathcal{G}$ under f-fraction local attack model is said to achieve resilient average consensus if for every initial value $x_i[0], i = 1, 2, \ldots, N$, it holds that $\lim_{k\to+\infty} x_j[k] = \sum_{i=1}^{N} x_i[0]/N, \forall j \in \mathcal{V}_n$.

Lastly, to merge the above two parts into a novel privacy-preserving adaptive resilient consensus algorithm.

III. Main Results

In this section, we present the privacy-preserving mechanism and the proposed algorithm of this article, along with some theorems. Before that, we give a restrictive but crucial assumption. Note that this assumption is a prerequisite for the main results derived in this article.

Assumption 1: The network environment of the system at $k = 1$ and $k = 2$ is secure, i.e., no cyberattack exists.

Remark 2: In real application scenarios, attackers often need a certain amount of preparation and deployment time to launch an attack on a target [15], [28], so it is reasonable to assume that the network is temporarily secure during the initial stage of the system.

A. Privacy-Preserving Mechanism

Here, we propose an improved state decomposition and reconstruction mechanism that can be used for digraphs.

Before giving our mechanism, let us recall the node state update rules used in [22], which are formulated as follows:

$$\begin{cases} x_i^{\alpha}[k+1] = x_i^{\alpha}[k] + \varepsilon \sum_{j\in\mathcal{N}_i^-} a_{ij}[k](x_j^{\alpha}[k] - x_i^{\alpha}[k]) \\ \qquad + \varepsilon a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k]) \\ x_i^{\beta}[k+1] = x_i^{\beta}[k] + \varepsilon a_{i,\alpha\beta}[k](x_i^{\alpha}[k] - x_i^{\beta}[k]). \end{cases} \tag{2}$$

We note that there is a carefully preset parameter ε in the update rules (2), which is restricted to $(0, \frac{1}{\Delta}]$ and $\Delta \triangleq \max_{i=1,2,\ldots,N} |\mathcal{N}_i|$. It requires global information about the network and is shared by all agents, so the update rules may not be fully distributed. Also, the above update rules are involved throughout the process of achieving consensus in the system. However, considering only the implementation of the privacy protections of the initial value of the nodes does not require the full process of the state decomposition method to be involved. Lastly, the original state decomposition method is only applicable to undirected graphs, which is one of its shortcomings. Therefore, we extract the core of the state decomposition idea from it and delete the parameter ε. Then propose a simplified but fully distributed update rules, which is applicable to balanced graphs and defined as follows:

$$\begin{cases} \theta_i^{\alpha}[1] = x_i^{\alpha}[0] + \sum_{j\in\mathcal{N}_i^-} a_{ij}(x_j^{\alpha}[0] - x_i^{\alpha}[0]) \\ \qquad + a_{i,\alpha\beta}(x_i^{\beta}[0] - x_i^{\alpha}[0]) \\ \theta_i^{\beta}[1] = x_i^{\beta}[0] + a_{i,\alpha\beta}(x_i^{\alpha}[0] - x_i^{\beta}[0]). \end{cases} \tag{3}$$

Theorem 1: Under Assumption 1, each node $i \in \mathcal{V}$ in the balanced graph $\mathcal{G}$ is iterated using the update rules (3). After the iteration each node state value is reconstructed, the system average is keep constant and the privacy of the initial state of nodes is preserved.

Proof: At the beginning, each node i in the network $\mathcal{G}$ decomposes its own initial value $x_i[0]$ into two substates $x_i^{\alpha}[0]$ and $x_i^{\beta}[0]$ as it usually does. Then, all nodes in the network perform one iteration according to the update rules (3) and recombine the two substates into one state value. Now, we have

$$\begin{aligned} x_i[1] &= \theta_i[1] \\ &= \frac{1}{2}(\theta_i^{\alpha}[1] + \theta_i^{\beta}[1]) \\ &= \frac{1}{2}\left(x_i^{\alpha}[0] + x_i^{\beta}[0] + \sum_{j\in\mathcal{N}_i^-} a_{ij}(x_j^{\alpha}[0] - x_i^{\alpha}[0])\right). \end{aligned} \tag{4}$$

Let

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N} x_i[1] &= \frac{1}{N}\sum_{i=1}^{N} \theta_i[1] \\ &= \frac{1}{2\,N}\sum_{i=1}^{N}(x_i^{\alpha}[0] + x_i^{\beta}[0]) \end{aligned}$$

$$+\frac{1}{2N}\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i^-}a_{ij}(x_j^\alpha[0]-x_i^\alpha[0])\right). \tag{5}$$

Recall the definition from [29], a graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathrm{A})$ is called balanced, which satisfied

$$\sum_{j\in\mathcal{N}_i^-}a_{ij}=\sum_{j\in\mathcal{N}_i^-}a_{ji}\qquad\forall i. \tag{6}$$

Since

$$\begin{aligned}
&\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i^-}a_{ij}(x_j^\alpha[0]-x_i^\alpha[0])\right)\\
&=\sum_{j\in\mathcal{N}_1^-}a_{1j}(x_j^\alpha[0]-x_1^\alpha[0])\\
&\quad+\sum_{j\in\mathcal{N}_2^-}a_{2j}(x_j^\alpha[0]-x_2^\alpha[0])\\
&\quad+\vdots\\
&\quad+\sum_{j\in\mathcal{N}_N^-}a_{Nj}(x_j^\alpha[0]-x_N^\alpha[0])\\
&=\sum_{j\in\mathcal{N}_1^-}a_{1j}x_1^\alpha[0]-\sum_{j\in\mathcal{N}_1^-}a_{j1}x_1^\alpha[0]\\
&\quad+\sum_{j\in\mathcal{N}_2^-}a_{2j}x_2^\alpha[0]-\sum_{j\in\mathcal{N}_2^-}a_{j2}x_2^\alpha[0]\\
&\quad+\vdots\\
&\quad+\sum_{j\in\mathcal{N}_N^-}a_{Nj}x_N^\alpha[0]-\sum_{j\in\mathcal{N}_N^-}a_{jN}x_N^\alpha[0]\\
&=\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i^-}a_{ij}x_i^\alpha[0]-\sum_{j\in\mathcal{N}_i^-}a_{ji}x_i^\alpha[0]\right)
\end{aligned} \tag{7}$$

substituting (6) into (7) and we have

$$\begin{aligned}
&\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i^-}a_{ij}(x_j^\alpha[0]-x_i^\alpha[0])\right)\\
&=\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i^-}a_{ij}x_i^\alpha[0]-\sum_{j\in\mathcal{N}_i^-}a_{ji}x_i^\alpha[0]\right)\\
&=0.
\end{aligned} \tag{8}$$

Thus, in (5)

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}x_i[1]&=\frac{1}{2N}\sum_{i=1}^{N}(x_i^\alpha[0]+x_i^\beta[0])\\
&\quad+\frac{1}{2N}\sum_{i=1}^{N}\left(\sum_{j\in\mathcal{N}_i^-}a_{ij}(x_j^\alpha[0]-x_i^\alpha[0])\right)\\
&=\frac{1}{N}\sum_{i=1}^{N}x_i[0]
\end{aligned} \tag{9}$$

which proves that the average of the system keeps constant.

Below, we proceed to demonstrate the privacy of the initial state of the node. Let us consider two popular types of eavesdroppers. One is a system internal eavesdropper known as a semihonest agent and the other is a system external eavesdropper known as an active adversary. Semihonest agents are nodes in the system that follow update rules to perform state updates but are curious about the state values of neighboring nodes. An active adversary is an eavesdropper that has access to the system topology and information transmitted between nodes.

Without losing generality, use $x_i[0]$ to denote the initial state value of node i that the eavesdropper tries to infer. According to the idea of state decomposition, the initial state value $x_i[0]$ of node i can be deduced from $x_i^\alpha[0]+x_i^\beta[0]=2x_i[0]$. Inevitably, $x_i^\alpha[0]$ is public. Therefore, inferring the value of the initial state of the node $x_i[0]$ is equivalent to inferring the value of $x_i^\beta[0]$. First, consider the existence of a semihonest node $m\in\mathcal{N}_i^-$. The set of interaction information that node m can obtain at $k=1$ is defined as

$$\mathcal{I}_m[1]=\{x_i^\alpha[0];x_m^\alpha[0],x_m^\beta[0],a_{m,\alpha\beta};a_{mp},x_p^\alpha[0]|_{p\in\mathcal{N}_m^-\setminus\{i\}}\}.$$

The value of $x_i^\beta[0]$ can be calculated by

$$x_i^\beta[0]=2x_i[1]-x_i^\alpha[0]-\sum_{j\in\mathcal{N}_i^-}a_{ij}(x_j^\alpha[0]-x_i^\alpha[0]). \tag{10}$$

It is obvious that for node m all other variables are unknown except $x_i^\alpha[0]$. It does not even know who is a neighbor of node i because our mechanism is fully distributed. Consequently, the privacy of the initial state of node i is perfectly preserved. Second, consider the existence of an external active adversary e. The active adversary has a greater threat compared to the internal semihonest node because it has access to the topology of the network and all the data transmitted between nodes. The interaction information set available to adversary e at $k=1$ is

$$\mathcal{I}_e[1]=\{x_i^\alpha[0];x_j^\alpha[0]|_{j\in\mathcal{N}_i^-}\}.$$

Looking back at (10), all variables are known at this point except for the $x_i[1]$ and a_{ij}, which are unknown. It is worth stating that for all nodes $i\in\mathcal{V}$, the state value $x_i[k]$ generated at time k by the dynamics will be stored first and then broadcasted only at time $k+1$. At the same time, the node will receive the state at time k sent by other nodes $j\in\mathcal{N}_i^-$. So, the active adversary cannot estimate $x_i[0]$ precisely and the initial state privacy of the node is protected. This finishes the proof. ■

B. Resilient Consensus Algorithm

We consider a network in which no communication delay exists. Each node $i\in\mathcal{V}$ sends its data at time k simultaneously with receiving the information sent by the neighboring nodes. Having protected the privacy of nodes by the above privacy-preserving mechanism, we now need to design algorithm that achieve resilient consensus under the proposed attack model. The final algorithm follows two objectives. The first one is to

Algorithm 1. Privacy-Preserving Adaptive Resilient Consensus Algorithm (PPARCA).

```
Input: Node initial state value $x_i[0]$
Output: Node updated state $x_i[k]$, storage vector
        $s^i[k] = [s_1^i[k], ..., s_{\hat{N}}^i[k]]$, $\hat{N} \geq N$.
Decompose node $i$ into two sub-states.
if $k = 1$ then
    for $j \in \mathcal{N}_i^-$ do
        Node $i$ receive $x_j^{\alpha}[0]$.
    Node $i$ updates its value using (3) and
      subsequently uses (4) to obtain $x_i[1]$.
    Node $i$ initializes $s^i[1] = [\ ]_{1\times\hat{N}}$ and $s_i^i[1] = x_i[1]$.
if $k = 2$ then
    Node $i$ broadcasts its $s^i[1]$ to $j \in \mathcal{N}_i^+$.
    for $j \in \mathcal{N}_i^-$ do
        Receiving $s^j[1]$ to update
          $s_j^i[2] = s_j^j[1], j \in \mathcal{N}_i^-$.
if $k > 2$ then
    Node $i$ broadcasts its $s^i[k-1]$ to all its neighbors.
    for $n \in \{1, 2, ..., \hat{N}\}$ do
        if $\sum_{j\in\mathcal{N}_i^-} |s^j[k-1]| \geq d_i$ then
            Node $i$ accepts the same values
              $s_n^j[k-1], j \in \mathcal{N}_i^-$ sent from $\lfloor f d_i \rfloor + 1$
              incoming neighbors and writes it to the
              vector $s_n^i[k]$.
        else
            Node $i$ accepts the same values
              $s_n^j[k-1], j \in \mathcal{N}_i^-$ sent from the majority
              of incoming neighbors and writes it to the
              vector $s_n^i[k]$.
Node $i$ derives the updated state $x_i[k]$ by dynamics (1).
```

integrate the privacy-preserving mechanisms in the algorithm and achieve a resilient and accurate consensus. The second one is that the algorithm can adaptively identify and take different countermeasures when subjected to different cyberattacks. To this end, we propose the privacy-preserving adaptive resilient consensus algorithm (PPARCA). The pseudocode of PPARCA is shown in Algorithm 1.

Concretely, each node $i \in \mathcal{V}$ in the system will be equipped with a long-term storage vector $s^i[k] = [s_1^i[k], \ldots, s_{\hat{N}}^i[k]]$, $\hat{N} \geq N$. The function of this storage vector is to record the data received by the node from its neighbor nodes $j \in \mathcal{N}_i^-$ and save the final accepted state values. The element $s_n^i[k], n \in \{1, 2, \ldots, \hat{N}\}$ in the storage vector denotes the recorded value of node n state in node i. It is worth mentioning that, from the point of view of facilitating practical applications in real-world environments, we default to nodes knowing the possible upper bound of the number of system's nodes $\hat{N} \geq N$ without limiting its specific value [13]. The algorithm starts executing if each node is ready for its initial state and continues performing throughout the consensus process until the end.

Remark 3: Note that the parameter $\hat{N}$ is in some way a global parameter, which may lead to a nonfully distributed problem of the algorithm. Existing maximum consensus algorithms can be used in advance while satisfying Assumption 1 so that each node forms a maximum consensus on parameter $\hat{N}$. Further, if Assumption 1 is not satisfied, cryptographic or resilient maximum consensus algorithms such as [30] can be used to achieve the same purpose.

At first, each node is decomposed into two substates according to the state decomposition idea and then waits for the first iteration. At the time $k = 1$, the nodes perform one iteration to update and recombine the resulted privacy protection values using the proposed privacy-preserving mechanism, and acquire $x_i[1]$. Meanwhile, the storage vector is initialized to $s^i[1] = [\ ]_{1\times\hat{N}}$ where $[\ ]$ represents an empty vector, afterward set $s_i^i[1] = x_i[1]$. At time $k = 2$ node i broadcasts $s^i[1]$ to all neighboring nodes $j \in \mathcal{N}_i^+$ and receives $s^j[1]$ to update its own storage vector $s_j^i[2] = s_j^j[1], j \in \mathcal{N}_i^-$.

To resist the cyberattacks that occur at time $k > 2$, an majority rule acceptance mechanism is introduced. This mechanism adaptively defends against deception attacks or DoS attacks by determining the number of received storage vectors $s^j[k], j \in \mathcal{N}_i^-$. If node i can receive the storage vector $s^j[k]$ from its neighbor nodes, then $|s^j[k]| = 1$, otherwise $|s^j[k]| = 0$. As we discussed before, deception attacks cause damage to the integrity of transmitted data, but no data loss. In this case, the number of storage vectors will be greater than or equal to d_i. Node i accepts the identical values $s_n^j[k-1], j \in \mathcal{N}_i^-$ sent from $\lfloor f d_i \rfloor + 1$ neighbors and saves it to the vector $s_n^i[k]$. DoS attacks lead to data loss, so the number of storage vectors will be less than d_i. Under this circumstance, node i accepts the identical values $s_n^j[k-1], j \in \mathcal{N}_i^-$ sent from the majority of neighbors and saves it to the vector $s_n^i[k]$. Eventually, node i is computed by the accepted storage vector $s_n^i[k]$ to gain u_i, which is then used by (1) to update the state $x_i[k]$ and asymptotically converge to the average consensus.

As time passes and the algorithm is continuously executed, node i receives and accepts the latest data of storage vector from its neighboring nodes, we define the control input of node i at time $k > 1$ as follows:

$$u_i[k] = \frac{\sum s_n^i[k]}{\lambda[k]}, \quad n \in \mathbb{S}^i[k] \tag{11}$$

where $\mathbb{S}^i[k]$ is the index set of the elements in the storage vector $s^i[k]$, and the cardinality of the index set is given by $\lambda[k] = |\mathbb{S}^i[k]|$. Accordingly, we rewrite the time $k > 1$ part of dynamics (1) as

$$x_i[k] = \varepsilon_i x_i[k-1] + (1 - \varepsilon_i)\frac{\sum s_n^i[k]}{\lambda[k]}, \quad n \in \mathbb{S}^i[k]. \tag{12}$$

The control gain ε_i takes values in the range $0 \leq \varepsilon_i < 1$. Control gain of 0 means the node's state $x_i[k] = u_i[k]$. Noted that the value of the control gain ε_i can be arbitrary within the limit, but its magnitude may determine the sensitivity of the node's dynamics to instantaneous varieties in the state values [13].

Next, we will substantiate how our proposed algorithm can induce digraph with certain network robustness condition to accurate resilient average consensus.

Theorem 2: Consider an MAS in the presence of f-fraction local cyberattacks, if its network topology satisfies the strongly

p-fraction robust graph, where $2f < p \leq 1$, then each normal node in the system can achieve accurate average consensus after $K \geq N-1$ iterations under the PPARCA.

Proof: The nodes in the system do not know when to stop running the algorithm, so they embed the time K to indicate the minimum number of times to run PPARCA. We assume that there exists a finite time $K \geq N-1$, by which time all normal nodes will have received all the initial states that have not been attacked. The exact procedure for introducing this lower bound parameter K is described in [13] and is omitted here for brevity. Just note that in terms of time consumption compared to the SABA proposed in [13], our algorithm has an extra processing step to protect the privacy of the initial state, and thus requires an extra iteration. Consider node $y \in \mathcal{V}$, each of whose outgoing neighbors $i \in \mathcal{N}_y^+$ receives the initial state $x_y[0]$ straight away. That all other normal nodes can receive the correct $x_y[0]$ will be proved by using contradiction. Let the set of all nodes for which $x_y[0]$ is not available be represented by $\mathcal{U}$. The problem is to be discussed in two cases as follows.

1) *Deception attacks:* According to definition 3, we can learn that at least $\lceil pd_i \rceil$ of the incoming neighbors are outside the set $\mathcal{U}$ for some node $i \in \mathcal{U}$. At most $\lfloor fd_i \rfloor$ of them send over $x_y[0]$ that have suffered deception attacks resulting in untrustworthiness. All other nodes that have received $x_y[0]$ will broadcast it again at certain time $k \leq K$. Because of $p \geq 2f$, node i can get at least $\lceil pd_i \rceil - \lfloor fd_i \rfloor \geq \lceil 2fd_i \rceil - \lfloor fd_i \rfloor \geq \lfloor fd_i \rfloor + 1$ correct $x_y[0]$ to use, which then yields the storage vector. This contradicts the assumption that node i has no access to the initial state of node y and same for other nodes. As a result, all nodes correctly acquire $x_y[0]$;
2) *DoS attacks:* Similarly, according to definition 3, we can learn that at least $\lceil pd_i \rceil$ of the incoming neighbors are outside the set $\mathcal{U}$ for some node $i \in \mathcal{U}$. At most $\lfloor fd_i \rfloor$ of them loss $x_y[0]$ that have suffered DoS attacks. All other nodes that have received $x_y[0]$ will broadcast it again at a certain time $k \leq K$. Because of $p \geq 2f$, node i can get at least $\lfloor fd_i \rfloor + 1$ of $x_y[0]$ to use, which account for the majority of all data and then generate the storage vector. This also contradicts the assumption that node i has no access to the initial state of node y and same for other nodes. As a result, all nodes acquire $x_y[0]$.

Now, after executing the PPARCA for K times, all nodes $i \in \mathcal{V}_n$ obtain the right and sufficient amount of $x_i[0], i \in \mathcal{V}$. Looking back at (11), $u_i[k]$ is a linear combination of the initial states $x_i[0]$ received and final accepted at each moment k. Each node i will eventually receive the initial states of other nodes at $t = K$ and converge asymptotically to x_a, i.e.,

$$\lim_{k \to +\infty} u_i[k] = x_a, \quad \forall i \in \{1, 2, \ldots, N\}. \tag{13}$$

Here, adding and subtracting x_a to the left side of (12) and adding and subtracting $\varepsilon_i x_a$ to the right side, we get

$$\begin{aligned}(x_i[k] - x_a) &- (u_i[k] - x_a) \\ &= \varepsilon_i(x_i[k-1] - x_a) - \varepsilon_i(u_i[k] - x_a).\end{aligned} \tag{14}$$

As $k \to +\infty$, according to (13), we have

$$x_i[k] - x_a = \varepsilon_i(x_i[k-1] - x_a). \tag{15}$$

Hence, the system comes with Schur stable when $0 \leq \varepsilon_i < 1$, and the theorem is proved.

The implication of Theorem 2 is that when an MAS suffers from the cyberattacks model proposed in this article, strongly p-fraction robust graph condition guarantees the network robustness of the system. In this case, the disruption of the state vector of the compromised nodes by the cyberattacks can be eliminated by performing PPARCA. If the network is not robust enough, then cyberattacks causes most nodes in the system to be compromised. For a normal node in the system, its received state vector of neighboring nodes that are not compromised by cyberattacks is less than what the algorithm requires. This makes the normal nodes uncertain about the state of each iteration, which eventually leads to the failure of the system's consensus.

Theorem 2 gives a sufficient but not necessary condition. We also give a necessary condition for achieving resilient average consensus utilizing PPARCA which is effective for f-fraction local cyberattacks model.

Theorem 3: If an MAS achieves resilient average consensus by performing PPARCA under f-fraction local cyberattacks, its fundamental topology graph is p-fraction robust, where $p > 2f$.

Proof: We use contradiction to prove it. Suppose a network $\mathcal{G}$ that achieves average consensus under the f-fraction local cyberattacks by performing PPARCA has a p-fraction robust fundamental topology graph, where $p \leq 2f$. Under the assumption that there exists node i belonging to set $\mathcal{S}$, it has at most $\lceil pd_i \rceil$ incoming data from outside set $\mathcal{S}$ and uses them with (12) to compute state values. In the worst case, all $\lfloor fd_i \rfloor$ of these incoming data are corrupted or lost by the cyberattacks, leaving at most $\lfloor fd_i \rfloor$ of data from other nodes. At this point, the corrupted data occupies the majority or at least half of all state data, so node i cannot select the correct initial state of other nodes received from its neighbors by the adaptive majority rule acceptance mechanism in the PPARCA. Hence, node i cannot achieve average consensus, which contradicts the hypothesis. ■

IV. Simulation and Practical Application Experiments

In this section, we first verify the privacy-preserving and resilient average consensus capabilities of PPARCA through simulation experiments. Then, we build an MAS based on Raspberry Pi development board by referring to the communication topology in the simulation experiments. PPARCA is deployed to this MAS to verify its practical application effectiveness.

A. Simulation Experiments

Consider an MAS consisting of six nodes whose fundamental topology is shown in Fig. 3. The network suffers from f-fraction local cyberattacks and possesses strongly p-fraction robust, where $f = 0.2, p = 0.5$. The initial state of agents are selected as $x_i[0] = i, i = 1, 2, \ldots, 6$, which makes the average value $x_a = 3.5$. Agent 2 is postulated to be the target of cyberattacks. Each agent in the network performs PPARCA and sets control gain $\varepsilon_i = 0$ for simplicity. To ensure maximum protection of

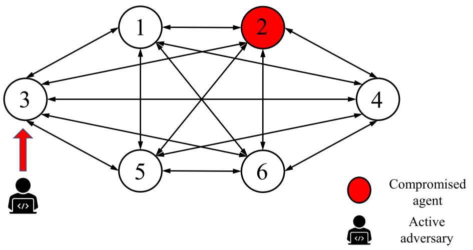

Fig. 3. Fundamental topology graph of six nodes MASs.

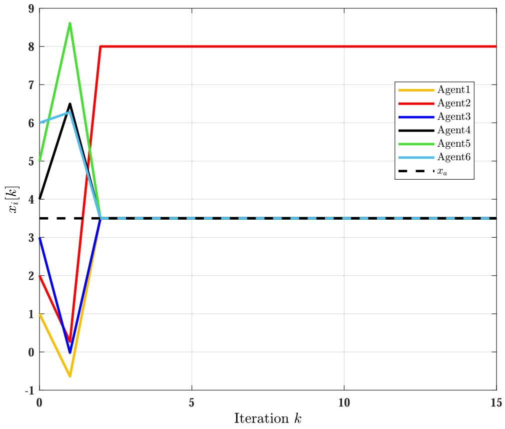

Fig. 4. Simulation result of MASs under deception attacks.

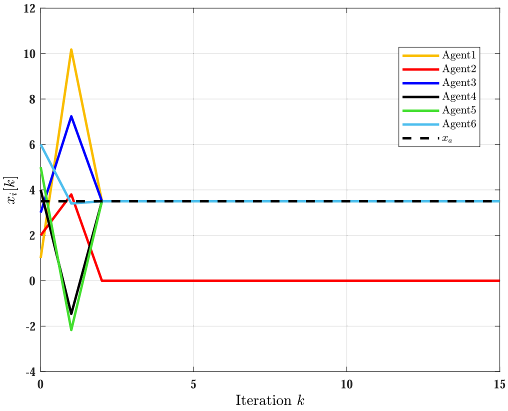

Fig. 5. Simulation result of MASs under DoS attacks.

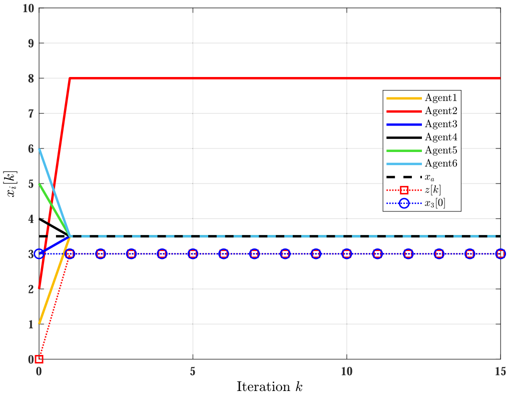

Fig. 6. Observed result of active adversary under PPARCA without privacy-preserving mechanism.

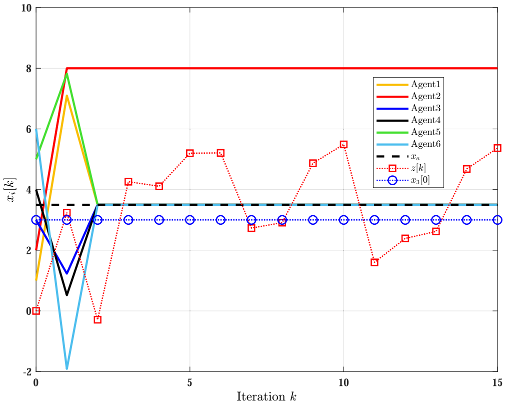

Fig. 7. Observed result of active adversary under complete PPARCA.

initial state privacy, $a_{i,\alpha\beta}$ is chosen randomly within the $\mathbb{R}$, and a_{ij} is chosen randomly within the $\mathbb{R}^+$ but to satisfy the requirements of the balanced digraph.

To begin with, we observe the adaptive resilience of PPARCA to different types of cyberattacks under the new functional cyberattacks model. Note that, for simplicity reasons, our simulations examine the adaptive capabilities of an individual agent against different types of cyberattacks separately. Our algorithm is still applicable in the context of large-scale distributed MASs that are subject to deception attacks and DoS attacks simultaneously. The deception attack disrupts agent 2 at $k > 2$ by starting to continuously broadcast the wrong data $s_n^2[k] = 8, n \in \{1, 2, \ldots, \hat{N}\}$ to its neighbors. Fig. 4 shows the trajectory of the state value change for each agent, and it can be seen that each agent that is not subjected to deception attacks has reached a consensus on the average initial state, where $x_i[K] = x_a, i = 1, 3, 4, 5, 6$. The DoS attack interferes with agent 2 at $k > 2$ so that it cannot send and receive data, which for the whole system behaves as $x_2[k] = 0$. Fig. 5 shows the trajectory of the state value change for each agent, and it can be seen that each agent that is not subjected to DoS attacks has reached a consensus on the average initial state, where $x_i[K] = x_a, i = 1, 3, 4, 5, 6$.

Next, we observe the privacy of node initial value of PPARCA under the deception attacks. The network topology of the system is likewise depicted by Fig. 3. We consider the situation where an active adversary tries to obtain the initial state $x_3[0]$ and its observed value is represented by $z[k]$, since the active adversary has more information than the semihonest agent and is more threatening. We first delete the privacy-preserving mechanism in PPARCA. The results in Fig. 6 demonstrate that the execution of PPARCA without privacy-preserving mechanism by agent leads to easy access to its initial state for an eavesdropper. Instead, as shown in Fig. 7, the use of complete PPARCA prevents the active adversary from obtaining the exact initial state, and the active adversary can only infer by randomly selecting some of the data spread through the network.

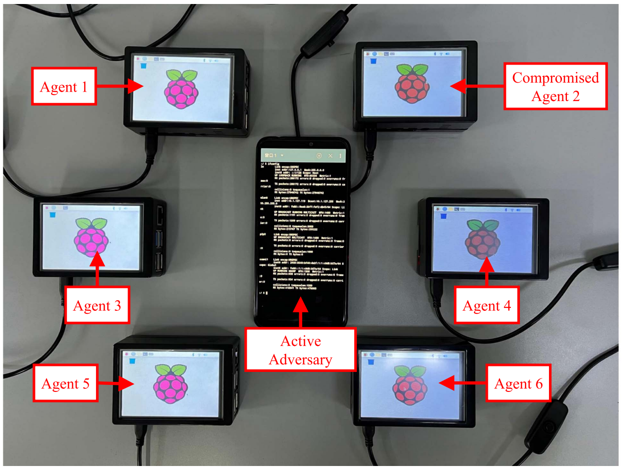

Fig. 8. Experimental setup of the Raspberry Pi agents and adversary.

B. Practical Application Experiments

In order to verify the above simulation results and illustrate the practical application performance of PPARCA, we built an MAS based on Raspberry Pi development board. Raspberry Pi is widely used in industrial applications due to its flexibility and light-weight [31], [32]. Therefore, we chose to build an MAS based on Raspberry Pi to demonstrate the effectiveness of PPARCA in industrial applications.

Each piece of Raspberry Pi 4 Model B is equipped with Cortex-A72 64-bit 1.5 GHz SoC, 4 GB RAM, sensor module, and communication module. These components make any Raspberry Pi development board has a certain perception, interaction and computing power. Therefore, a Raspberry Pi can be regarded as an intelligent agent. Referring to the MAS communication topology of six nodes considered in the simulation experiments, as shown in Fig. 3, we built an practical MAS composed of 6 Raspberry Pi agents. Fig. 8 shows the setup of the Raspberry Pi agents and the adversary. This Raspberry Pi MAS suffers from 0.2-fraction local cyberattacks and is strongly 0.5-fraction robust, which is the same as the setting in the simulation experiments. Each agent in the system is connected to the one local area network under IEEE 802.11 standards via open channel. The communication between agents and the implementation of PPARCA are conducted by Python code. In addition, an Android phone connected to the same local area network acts as the active adversary. The expectation of this MAS is that the flashing frequency of the LEDs on each Raspberry Pi board achieves average consensus, to ensure that the sensors or controllers indicated by the LEDs are in the same state. The dynamics that are embedded in each Raspberry Pi agent to control the flashing frequency of the LEDs are

$$f_i[k] = \begin{cases} \theta_i[k] & k = 1 \\ \varepsilon_i f_i[k-1] + (1-\varepsilon_i) u_i[k] & k > 1 \end{cases} \tag{16}$$

where $f_i[k]$ denotes the frequency of the agent i at time k, θ_i is the privacy-preserving frequency value, ε_i is the control gain and sets to 0, and u_i is the control input determined by other agents.

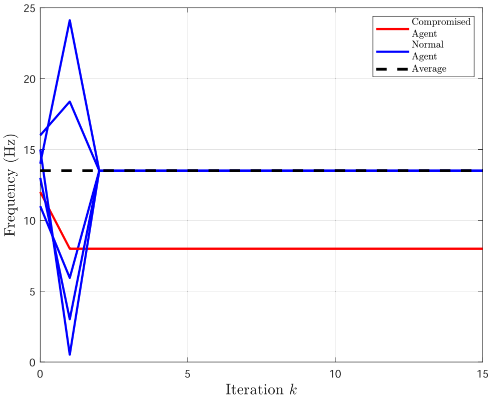

Fig. 9. Frequency traces of Raspberry Pi MAS under deception attack.

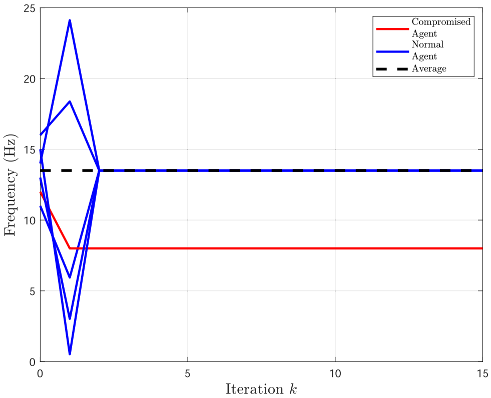

Fig. 10. Frequency traces of Raspberry Pi MAS under DoS attack.

In order to reflect and strengthen the link between simulation experiment and practical application experiment, we implemented practical application experiment corresponding to simulation experiment one by one. First, observe the performance of Raspberry Pi MAS under different types of cyberattacks. The active adversary implemented deception attacks and DoS attacks through tampering with data packets and UDP flooding, respectively. The frequency traces of Raspberry Pi agents in MAS under deception attacks is shown in Fig. 9, and the frequency traces under DoS attacks is shown in Fig. 10. The blue solid lines indicate the frequency traces of normal agents in MAS, and the red solid line indicates the frequency trace of compromised agent. The results indicate that the Raspberry Pi MAS under PPARCA control achieved average consensus, whether under deception attacks or DoS attacks.

Next, observe the privacy performance of Raspberry Pi MAS while suffering from deception attacks. The active adversary eavesdrops on agent 3's interaction data through the open channel and conducts deception attacks against agent 2. We consider the situation where the active adversary tries to obtain the initial state $x_3[0]$ and its observed value is represented by $z[k]$. The frequency traces of Raspberry Pi MAS without privacy-preserving mechanism and with complete PPARCA are shown in Figs. 11 and 12, respectively, along with the observed value trajectory of the adversary. The blue dashed line with

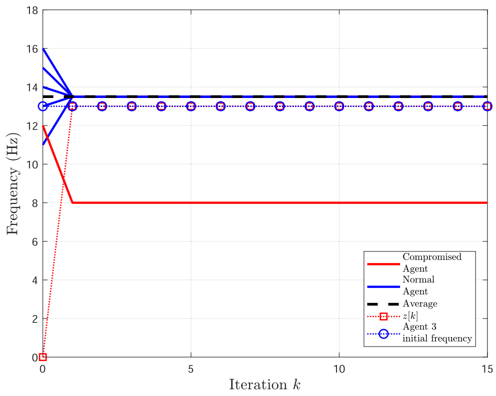

Fig. 11. Frequency traces of Raspberry Pi MAS without privacy-preserving mechanism and observed result of active adversary.

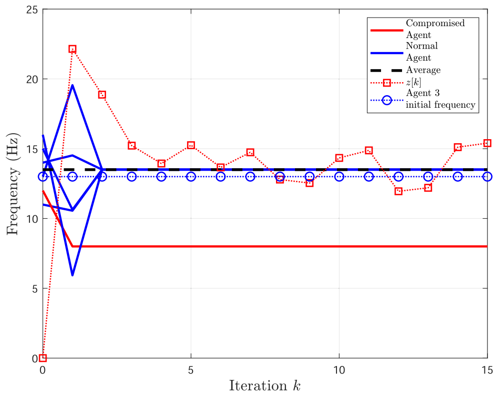

Fig. 12. Frequency traces of Raspberry Pi MAS under complete PPARCA and observed result of active adversary.

circles indicates the initial frequency of agent 3, and the red dashed line with boxes indicates the observed value of the active adversary's eavesdropping on the initial frequency of agent 3. The results demonstrate that the initial frequency privacy of agent 3 is successfully protected. Moreover, the normal agents in MAS achieved accurate average consensus under the attacked environment.

The MAS built based on Raspberry Pi not only reflects the correctness of the simulation experiments, but also verifies the effectiveness of PPARCA in practical industrial applications.

V. Conclusion

In this article, we had presented a novel privacy-preserving adaptive resilient average consensus algorithm for MASs. We had improved the original state decomposition method to make it more simplified but fully distributed for realistic application. Further, based on the SABA, we had also modified it to confront the f-fraction local cyberattack model under strongly p-fraction robust network condition for rendering it more practically oriented. Finally, the features of the proposed algorithm had been validated by some simulation examples.

Although we had studied the problem of privacy preservation and resilient consensus, the algorithm still had some deficiencies. How to make the algorithm applicable to general directed graphs and how to solve the problem that the algorithm is immediately subjected to cyberattacks in the initial stage are among the directions worthy of future research.

Acknowledgment

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the anonymous reviewers for their valuable comments and suggestions that have significantly improved the quality of this article.

References

[1] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Syst. Mag.*, vol. 39, no. 3, pp. 40–72, Jun. 2019.

[2] Z. Zuo, Q.-L. Han, B. Ning, X. Ge, and X.-M. Zhang, "An overview of recent advances in fixed-time cooperative control of multiagent systems," *IEEE Trans. Ind. Inform.*, vol. 14, no. 6, pp. 2322–2334, Jun. 2018.

[3] L. E. Beaver and A. A. M. records, "Constraint-driven optimal control of multiagent systems: A highway platooning case study," *IEEE Control Syst. Lett.*, vol. 6, pp. 1754–1759, 2022.

[4] C. Zhao, J. Chen, J. He, and P. Cheng, "Privacy-preserving consensus-based energy management in smart grids," *IEEE Trans. Signal Process.*, vol. 66, no. 23, pp. 6162–6176, Dec. 2018.

[5] B. Xu, F. Guo, W.-A. Zhang, W. Wang, C. Wen, and Z. Li, "Distributed successive convex approximation for nonconvex economic dispatch in smart grid," *IEEE Trans. Ind. Inform.*, vol. 17, no. 12, pp. 8288–8298, Dec. 2021.

[6] Z. Feng and G. Hu, "Secure cooperative event-triggered control of linear multiagent systems under DoS attacks," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 741–752, May 2020.

[7] W. Xu, G. Hu, D. W. Ho, and Z. Feng, "Distributed secure cooperative control under denial-of-service attacks from multiple adversaries," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3458–3467, Aug. 2020.

[8] W. Fu, J. Qin, Y. Shi, W. X. Zheng, and Y. Kang, "Resilient consensus of discrete-time complex cyber-physical networks under deception attacks," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4868–4877, Jul. 2020.

[9] W. He, Z. Mo, Q.-L. Han, and F. Qian, "Secure impulsive synchronization in lipschitz-typemulti-agent systems subject to deception attacks," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 5, pp. 1326–1334, Sep. 2020.

[10] D. Zhang and G. Feng, "A new switched system approach to leader–follower consensus of heterogeneous linear multiagent systems with DoS attack," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 2, pp. 1258–1266, Feb. 2021.

[11] S. Du, W. Xu, J. Qiao, and D. W. Ho, "Resilient output synchronization of heterogeneous multiagent systems with DoS attacks under distributed event-/self-triggered control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1169–1178, Mar. 2023.

[12] J. Yan, X. Li, Y. Mo, and C. Wen, "Resilient multi-dimensional consensus in adversarial environment," *Automatica*, vol. 145, 2022, Art. no. 110530.

[13] S. M. Dibaji, M. Safi, and H. Ishii, "Resilient distributed averaging," in *Proc. IEEE Amer. Control Conf.*, 2019, pp. 96–101.

[14] W. He, W. Xu, X. Ge, Q. L. Han, W. Du, and F. Qian, "Secure control of multiagent systems against malicious attacks: A brief survey," *IEEE Trans. Ind. Inform.*, vol. 18, no. 6, pp. 3595–3608, Jun. 2022.

[15] H. Ishii, Y. Wang, and S. Feng, "An overview on multi-agent consensus under adversarial attacks," *Annu. Rev. Control*, vol. 53, pp. 252–272, 2022.

[16] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proc. ACM Workshop Privacy Electron. Soc.*, 2012, pp. 81–90.

[17] T. Ding, S. Zhu, J. He, C. Chen, and X. Guan, "Differentially private distributed optimization via state and direction perturbation in multiagent systems," *IEEE Trans. Autom. Control*, vol. 67, no. 2, pp. 722–737, Feb. 2022.

[18] S. Wang et al., "Local differential private data aggregation for discrete distribution estimation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2046–2059, Sep. 2019.

[19] D. Wang, N. Zheng, M. Xu, Y. Wu, Q. Hu, and G. Wang, “Resilient privacy-preserving average consensus for multi-agent systems under attacks,” in *Proc. IEEE 16th Int. Conf. Control, Autom., Robot. Vis.*, 2020, pp. 1399–1405.

[20] Q. Li, I. Cascudo, and M. G. Christensen, “Privacy-preserving distributed average consensus based on additive secret sharing,” in *Proc. IEEE 27th Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[21] S. Zhang, T. Ohlson Timoudas, and M. A. Dahleh, “Consensus with preserved privacy against neighbor collusion,” *Control Theory Technol.*, vol. 18, no. 4, pp. 409–418, 2020.

[22] Y. Wang, “Privacy-preserving average consensus via state decomposition,” *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, Nov. 2019.

[23] D. Fiore and G. Russo, “Resilient consensus for multi-agent systems subject to differential privacy requirements,” *Automatica*, vol. 106, pp. 18–26, 2019.

[24] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[25] H. Zhang and S. Sundaram, “Robustness of information diffusion algorithms to locally bounded adversaries,” in *Proc. IEEE Amer. Control Conf.*, 2012, pp. 5855–5861.

[26] M. Calis, R. Heusdens, and R. C. Hendriks, “A privacy-preserving asynchronous averaging algorithm based on state decomposition,” in *Proc. IEEE 28th Eur. Signal Process. Conf.*, 2021, pp. 2115–2119.

[27] Y. Wang, J. Lu, W. X. Zheng, and K. Shi, “Privacy-preserving consensus for multi-agent systems via node decomposition strategy,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 68, no. 8, pp. 3474–3484, Aug. 2021.

[28] T. T. Huong et al., “Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach,” *Comput. Ind.*, vol. 132, 2021, Art. no. 103509.

[29] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[30] 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.

[31] G. Huang, X. Wu, F. Guo, L. Yu, and W.-A. Zhang, “DEID-based control of networked rapid control prototyping system: Design and applications,” *IEEE Trans. Ind. Electron.*, vol. 70, no. 1, pp. 1047–1056, Jan. 2023.

[32] S. E. Mathe, A. C. Pamarthy, H. K. Kondaveeti, and S. Vappangi, “A review on raspberry pi and its robotic applications,” in *Proc. IEEE 2nd Int. Conf. Artif. Intell. Signal Process.*, 2022, pp. 1–6.

Chenduo Ying received the B.E. degree in software engineering from NingboTech University, Ningbo, China, in 2020. He is currently working toward the M.S. degree in cyberspace security with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China.

His main research interests include resilient consensus control, privacy preservation, and distributed system security.

Ning Zheng received the M.S. degree in computer application from Zhejiang University, Hangzhou, China, in 1990.

He is currently a Full Professor with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China. His current research interests include multiagent security, information management systems, and privacy preservation.

Yiming Wu received the B.Eng. degree in automation and the Ph.D. degree in control science and engineering from Zhejiang University of Technology, Hangzhou, China, in 2010 and 2016, respectively.

He held a visiting position from 2012 to 2014 with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Since July 2016, he has been with the Hangzhou Dianzi University, Hangzhou, China, where he is currently an Associate Professor with the School of Cyberspace. His main research interests include multiagent systems, security and privacy theory, iterative learning control, and applications in intelligent transportation systems and sensor networks.

Ming Xu received the M.S. degree in computer software and Theory and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2000 and 2004, respectively.

He is currently a Full Professor with Hangzhou Dianzi University, Hangzhou, China. His main research interests include distributed system security and digital forensics.

Wen-An Zhang (Member, IEEE) received the B.Eng. degree in automation and the Ph.D. degree in control theory and control engineering from the Zhejiang University of Technology, Hangzhou, China, in 2004 and 2010, respectively.

Since 2010, he has been with the Zhejiang University of Technology, where he is currently a Professor with the Department of Automation. From 2010 to 2011, he was a Senior Research Associate with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong. His current research interests include networked control systems, multisensor information fusion estimation, and robotics.

Dr. Zhang was the recipient of the Alexander von Humboldt Fellowship in 2011–2012. Since September 2016, he has been a Subject Editor for *Optimal Control Applications and Methods*.