IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 8, 2021, accepted August 21, 2021, date of publication August 30, 2021, date of current version September 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3108639

Event-Triggered Resilient Average Consensus With Adversary Detection in the Presence of Byzantine Agents

PENG ZHANG[1], CHANGQING HU[2,3], SENTANG WU[1], RUIYAN GONG[2], AND ZIMING LUO[4]

[1]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2]Beijing Institute of Aerospace Control Devices, Beijing 100854, China
[3]Pilot National Laboratory for Marine Science and Technology (Qingdao), Qingdao 266237, China
[4]The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China

Corresponding author: Peng Zhang (zhang-peng@buaa.edu.cn)

ABSTRACT This paper addresses the problem of resilient average consensus in the presence of Byzantine agents in multi-agent networks. An event-triggered secure acceptance and broadcasting algorithm is proposed in which full knowledge of the network and high computational capabilities of each regular node are not required. The computational expense and communication times are also reduced for the event-triggered mechanism. We analyze the conditions for such a fully distributed algorithm to succeed in the f-local adversarial model. A new definition called an f-propagation graph, which is extended from r-robustness, turns out to be more accurate in describing the required topology conditions. Based on the proposed algorithm and topology conditions, we provide another algorithm to detect the adversarial nodes according to their abnormal behavior. When the network topology is an f-propagation graph, regular nodes that are equipped with the proposed algorithms update state values synchronously and eventually converge asymptotically to resilient average consensus. Simulation results are provided to verify the effectiveness of our proposed algorithms and the network topology conditions.

INDEX TERMS Multi-agent networks, resilient consensus, adversary detection, event-triggered, Byzantine agents.

I. INTRODUCTION

The consensus problem is widely recognized as one of the most fundamental problems in distributed multi-agent networks. It has attracted significant attention from diverse contexts due to its widespread application, with examples such as distributed computing [1], smart grids [2], sensor networks [3], and robotics teams [4]. In general, the consensus objective requires that the nodes in the network collectively reach agreement on global quantities of interest, such as a certain state variable or set of state variables. Often, agreement on the average, mean, or some other function of the state variables is desired.

One of the major challenges in consensus in large-scale multi-agent networks is their limited communication capability [5]. Nodes obtain information by sensor measurements, calculations, or communication only with neighbors in the network, due to limited communication channels, energy, or computational capacity. Another major concern in networked systems is security since large-scale distributed systems have many potential vulnerable points for failures or attacks. Cyberattacks [6] on multi-agent networks can be roughly classified into two types, denial-of-service attacks and deception attacks, depending on the adversary's security goals of the data exchanged through communication networks. Denial-of-service attacks [7] aim to disrupt data availability and exchange ability by maliciously consuming communication or computational resources. Deception attacks [8] intend to compromise data integrity and trustworthiness by manipulating packets over communication networks. To defend against these attacks, one common approach is to increase the barriers to entry in security, such as cryptographic techniques [9]. Another approach is to improve the resilience of the application layer protocols, such as by designing consensus algorithms so that even if a subset of the nodes are compromised, the remaining regular nodes are still

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

able to achieve their objective (possibly a relaxed version of the objective).

In this paper, our focus is on security issues in consensus dynamics and especially on averaging. While resilient consensus has been studied for a long time in the literature, a number of papers devoted to designing consensus algorithms in various scenarios have appeared during the past decade (e.g., [10]–[12]). Average consensus means that all nodes in the network reach the average of the initial states by exchanging local information among nodes. In [13], the authors first proposed a so-called weighted gossip algorithm for solving the distributed averaging problem by using weights calculated from the topological characteristics of the communication graph. Yu *et al.* [14] proposed a periodic gossiping algorithm, where each pair of agents executes repeatedly following a prespecified periodic schedule. Related work on privacy preservation of initial states can be seen in [15], where a privacy-preserving average consensus algorithm was proposed to guarantee the privacy of the initial state and asymptotic consensus on the exact average of the initial values by adding and subtracting random noise to the consensus process. However, most of these consensus schemes are easily disrupted in an uncertain environment where faults or even adversarial attacks can be present. In particular, we deal with so-called Byzantine attacks, where some of the nodes are hijacked and do not follow the given algorithms or may even attempt to keep the regular nodes from reaching consensus by sending inconsistent or misguiding information to their neighbors.

In previous work, connectivity has been considered a fundamental metric in analyzing resilience to adversaries [16]. Under the classical point-to-point communication model, f Byzantine nodes among n nodes can be overcome if and only if two conditions are satisfied: $n \geq 3f + 1$ and vertex connectivity at least $2f + 1$ [17], [18]. Under the wireless broadcast communication model, it is also sufficient for regular nodes to achieve consensus in the presence of f malicious nodes if the network connectivity is at least $2f + 1$ [19], [20]. However, these proposed methods either require that regular nodes have at least some full knowledge of the network topology or assume that the network is complete.

Recent remarkable advances made in resilient consensus include a novel definition of network robustness, termed r-robustness, which facilitates purely local interaction rules against adversarial nodes [21]. This property provides a comprehensive characterization of network topologies for algorithms such as Weighted Mean-Subsequence-Reduced (W-MSR) to succeed despite the presence of broad class adversaries. Vaidya *et al.* [22] provided a necessary and sufficient condition for the algorithm to succeed under the Byzantine model, which used different proof techniques but had similar main results. In [23], this notion of robustness in common random graph models for complex networks was studied, and it was shown that the properties of robustness and connectivity share the same values or threshold function in particular graphs. Since a computationally efficient method to check whether this property holds for an arbitrary graph is not available, [24] provided algorithms to build an r-robust graph, which starts with an r-robust graph and continually adds new nodes with incoming edges from at least r nodes in the existing graph. In [25], different algorithms to create r-robust graphs with the minimum necessary number of nodes were given, which are called F-elemental graphs. Larger r-robust graphs can be built by appending these elemental graphs sequentially to other r-robust graphs. The results for r-robustness were later generalized to different cases. Dibaji and Ishii [26] and [27] investigated resilient consensus of second-order sampled-data multi-agent systems and derived topological conditions in terms of graph robustness. The resilient consensus problem for switched multi-agent systems composed of continuous-time and discrete-time subsystems was considered in [28], where a switched filtering strategy for regular nodes based upon available local information was proposed. Wang and Ishii [29] developed event-triggered update rules that could mitigate the influence of the malicious agents and, at the same time, reduce the communication. Sundaram and Gharesifard [30] proposed a secure distributed optimization algorithm that guarantees that the regular nodes converge to the convex hull of the minimizers of their local functions under certain conditions on the graph topology. Dibaji *et al.* [31] proposed a fully distributed secure acceptance and broadcasting algorithm (SABA) in the presence of adversarial Byzantine agents, which relies on the strong robustness of the graph and is easy to implement in practice. Robust graphs have been proved useful in the context of resilient consensus problems. However, in these works, r-robustness is only applicable to a specific class of algorithms. When the update rules of the algorithm change, the r-robustness may not be suitable for the required topological properties.

Recently, there have been a growing number of research results on event-triggered control, the aim of which is to reduce the computational and communication burden while ensuring satisfactory system performance. Wu *et al.* [32] developed a distributed algorithm derived from the event-triggered strategy for achieving resilient consensus multi-agent networks under deception attacks. Rahnama and Antsaklis [33] proposed an event-triggered control design that guarantees synchronization for output passive agents. Learning-based control methods for mitigating adversarial effects are used, which are based on the passivity of agents and the statistical distribution of agents' output. Zegers *et al.* [34] investigated the approximate leader-follower consensus problem in the presence of Byzantine adversaries using event- and self-triggered controllers. A Lyapunov-based detection strategy is used to identify Byzantine agents in the neighborhood. The stability of the developed event- and self-triggered control strategies is proved through a non-smooth Lyapunov analysis. Zegers *et al.* [35] investigated formation control and leader tracking with robustness to Byzantine adversaries using event-triggered controllers. A reputation-based strategy was

developed for each agent to detect Byzantine agent behaviors within their neighbor set. Zhao *et al.* [36] studied edge-based event-triggered strategies for the average consensus problem in multi-agent systems. A distributed event-triggered algorithm was presented based on edge information rather than neighbor information. However, in these works, high computational capabilities of nodes are needed for state updating, and extra resources are needed for computation of the event triggering functions or for monitoring node states to detect when the states reach the thresholds for triggering events.

Motivated by these observations, in this paper, we approach the resilient average consensus problem in the presence of Byzantine nodes through a retrieval procedure and an adversary detection procedure. First, an algorithm is developed for resilient distributed retrieval, in which global knowledge of network topology and high computational capabilities of each regular node are not required. One reason for this drastic change is that, unless the method of [19] is used, the retrieval process does not involve the node dynamics or global graph knowledge. Furthermore, computational expense and broadcasting times are reduced, benefitting from the event-triggered mechanism. However, the cost of such fully distributed algorithms appears in more restrictive topology. Second, we analyze the topological properties as part of the convergence conditions, which shows that f-propagation graphs are more accurate than r-robust graphs [31] in describing the topological properties needed for the algorithms to succeed. Third, an adversary detection algorithm is developed. This adversary detection strategy relaxes the limitations of the adversarial model in [31], where it is supposed that adversarial nodes do not send faulty initial states. Finally, state update functions are given for regular nodes that use the accepted initial state values and converge to resilient average consensus asymptotically. Simulation results are provided to verify the effectiveness of the proposed algorithms, and comparisons are made with the Linear Consensus Protocol (LCP) [10] and the W-MSR algorithm. The main contributions are summarized as follows:

1) An event-triggered secure acceptance and broadcasting algorithm (E-SABA) is developed for resilient distributed retrieval of initial states.
2) We analyze the topological properties in terms of f-propagation graphs and calculate the lower-bounded number of time steps for nodes to run the algorithms as part of the convergence conditions.
3) An adversary detection algorithm (ADA) is developed to detect the adversarial nodes that deviate from the prescribed rules of E-SABA and then broadcast through the network.

The rest of the paper is organized as follows: In Section II, we introduce the preliminaries of graph theory and the adversary model, and we give a problem statement. In Section III, we introduce the system model and present E-SABA, the network topology conditions, and the time steps for E-SABA to succeed. The ADA and state update functions are provided in Section IV and Section V. The simulation results are given in Section VI. Finally, the conclusion and future work are provided in section VII.

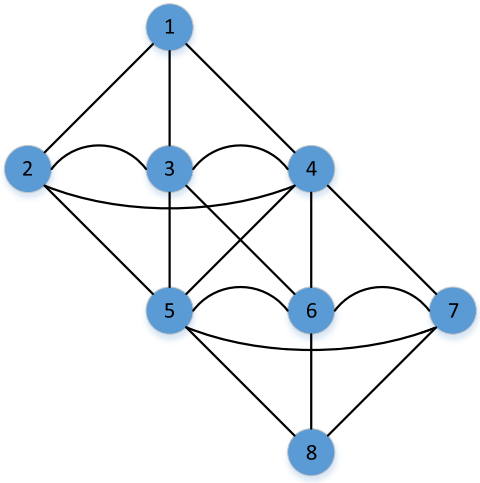

FIGURE 1. An example of network topology that is not a strongly 3-robust graph but a 1-propagation graph.

II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, first, some basic concepts in graph theory and the adversary model that will be used throughout the paper are reviewed; then, the adversary model is introduced, and the problem to be considered is formulated.

A. NOTATION AND GRAPH THEORY

Throughout this paper, the symbols $\mathbb{Z}$, $\mathbb{Z}^+$, $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}^+$ denote the set of integers, positive integers, natural numbers, real numbers, and positive real numbers, respectively. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. Given two sets $\mathcal{S}_1$ and $\mathcal{S}_2$, the union and intersection of the sets are denoted by $\mathcal{S}_1 \cup \mathcal{S}_2$ and $\mathcal{S}_1 \cap \mathcal{S}_2$, respectively. The reduction of $\mathcal{S}_1$ by $\mathcal{S}_2$ is denoted by $\mathcal{S}_1 \backslash \mathcal{S}_2 = \{x \in \mathcal{S}_1 : x \notin \mathcal{S}_2\}$.

A network modeled by an undirected graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge $(j, i) \in \mathcal{E}$ indicates that node i and node j can share information with each other. The neighbor set of node i is denoted as the set $\mathcal{N}_i = \{j, |(j, i) \in \mathcal{E}\}$, and the degree of node i is denoted by $d_i = |\mathcal{N}_i|$. The index of each node is used as a unique ID.

In this paper, the main result is closely related to the network topology. We introduce a graph property known as network robustness, which has been widely used in the literature on resilient distributed computation over networks.

Definition 1 (r-Reachable Set [24]): For a given graph $\mathcal{G}$, a nonempty subset $\mathcal{S}$ of nodes of $\mathcal{G}$ is said to be r-reachable if $\exists i \in \mathcal{S}$ such that $|\mathcal{N}_i \backslash \mathcal{S}| \geq r$, where $r \in \mathbb{Z}^+$.

Definition 2 (Strongly r-Robust Graph [24]): A graph is strongly r-robust if for any nonempty subset $\mathcal{S} \subseteq \mathcal{V}$, either $\mathcal{S}$ is r-reachable or $\exists i \in \mathcal{S}$ such that $\mathcal{V} \backslash \mathcal{S} \subseteq \mathcal{N}_i$, $r \in \mathbb{Z}^+$.

For a better description of the conditions for the network topology, we introduce two new concepts extended from the above definitions. An example can be seen in Fig. 1.

Definition 3 (f-Propagation Graph): A graph is an f-propagation graph if for any nonempty subset $\mathcal{S} \subseteq \mathcal{V}$, $\mathcal{S}$ satisfies at least one of the following conditions:
(i) $\exists i \in \mathcal{S}, |\mathcal{N}_i \backslash \mathcal{S}| \geq 2f + 1$.
(ii) $\forall i \notin \mathcal{S}, |\mathcal{N}_i \cap \mathcal{S}| \geq f + 1$.

Definition 4 (Regular f-Propagation Graph): A graph is a regular f-propagation graph if for any nonempty subset $\mathcal{S} \subseteq \mathcal{V}$, $\mathcal{S}$ satisfies at least one of the following conditions:
(i) $\exists i \in \mathcal{S}, |\mathcal{N}_i \backslash \mathcal{S}| \geq f + 1$.
(ii) $\forall i \notin \mathcal{S}, |\mathcal{N}_i \cap \mathcal{S}| \geq 1$.

B. ADVERSARY MODEL

We consider a consensus problem with Byzantine nodes in the network. Byzantine nodes can obtain complete knowledge of the network topology and all the communications between nodes in the network at every time step. They can deviate from any predefined algorithm rules, such as by updating their state values arbitrarily or transmitting different values to different neighbors, perhaps colliding with other Byzantine nodes.

It is quite clear that no distributed consensus among nodes can be achieved without constraining the adversary if there are too many adversarial nodes. We partition the set of nodes $\mathcal{V}$ into two subsets: the subset $\mathcal{R}$ of regular nodes and the subset $\mathcal{A} = \mathcal{V} \backslash \mathcal{R}$ of adversarial nodes. Thus, it is necessary to restrict the number of adversarial nodes in the network or the neighborhood of regular nodes. For dealing with distributed fault-tolerant algorithms, there are two adversarial node distribution models, the f-total model and the f-local model, that have been widely used in the literature. Under the f-total model, the total number of adversarial nodes in the network is upper bounded by the number of $f \in \mathbb{N}$. However, to allow for a large number of adversarial nodes in large-scale networks, we consider the locally bounded fault model taken from [37], [38], defined as follows:

Definition 5 (f-Local Adversarial Model): Each regular node i in the network has at most f adversarial nodes in the set of neighbors; i.e., $|\mathcal{N}_i \cap \mathcal{A}| \leq f, \forall i \in \mathcal{R}$.

In this paper, we consider a time-invariant network modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Once a node is categorized as a Byzantine node, it cannot be reincorporated into the regular node set even if it becomes cooperative. This strategy seems strict but ensures the accuracy of information sharing while considering the information from a node that was once Byzantine to be suspicious.

C. PROBLEM STATEMENT

Suppose that each node i has an initial state value given by $x_i[0]$ at time step $k = 0$. The goal is for each regular node i to gather sufficient information from its neighbors and to calculate some functions that depend on (some of) these initial values and then update its state value. To achieve this, nodes can exchange values with neighbors, securely identify and accept the values of other regular nodes and broadcast them through the network based on a strategy that adheres to the constraints imposed by the network topology. The average consensus problem involves designing distributed algorithms where each node updates its states using only the local information from neighbors so that all nodes $x_i[k]$ eventually converge to the initial state average $\bar{x} = \sum_{i=1}^{n} x_i[0]/n$, defined as follows:

Definition 6 (Average Consensus [31]): A network of n nodes over graph $\mathcal{G}$ is said to achieve average consensus if $\forall i \in \mathcal{V}, \lim_{k\to\infty} x_i[k] = \bar{x}$, where $\bar{x} = \sum_{i=1}^{n} x_i[0]/n, k \in \mathbb{Z}^+$.

The average consensus problem in the presence of Byzantine adversarial nodes is defined as follows:

Definition 7 (Resilient Average Consensus [31]): A network of n nodes over graph $\mathcal{G}$ under Byzantine adversarial attacks is said to achieve resilient average consensus if $\forall i \in \mathcal{R}, \lim_{k\to\infty} x_i[k] = \bar{x}$, where $\bar{x} = \sum_{i\in\mathcal{R}} x_i[0]/|\mathcal{R}|$, $k \in \mathbb{Z}^+$.

Unlike average consensus, the value $\bar{x}$ to be calculated is only the average of the initial state values of regular nodes which is despite the engagement of adversarial nodes in the process. In this paper, we aim to present a solution for the resilient average consensus problem. We describe the solution in three parts: the resilient distributed retrieval of the initial state values, the adversary detection process, and the state update rule each node has to execute. First, Algorithm 1 is for initial state information diffusion between nodes, which is described as resilient distributed retrieval. Second, Algorithm 2 is for regular nodes to detect Byzantine nodes in the set of neighbors, which deviate from the prescribed rules of Algorithm 1, and share the detection results. Third, regular nodes update states using the average of the initial state values received by nodes up to time instant k, converging to average consensus asymptotically. The formal statement of resilient distributed retrieval is defined as follows:

Definition 8 (Resilient Distributed Retrieval [31]) A network of N nodes over graph $\mathcal{G}$ under Byzantine adversarial attacks is said to achieve resilient distributed retrieval if $\forall i \in \mathcal{R}$, i can retrieve the initial state values of all the other regular nodes, i.e., $x_j[0], j \in \mathcal{R} \backslash \{i\}$.

III. RESILIENT DISTRIBUTED RETRIEVAL

To reach the average value of the network, regular nodes need to obtain the initial state values of the other nodes. In this section, we first introduce a novel event-triggered distributed algorithm for information retrieval in the network in the presence of f-local adversarial nodes. With this algorithm, regular nodes can securely identify and accept the true initial state values of the other nodes and broadcast them through the network. Then, we analyze the required constraints on the network topology that guarantee that all regular nodes in the network will achieve resilient distributed retrieval using the proposed algorithm.

A. SYSTEM MODEL

Each node i holds a state value $x_i[k]$, a state vector $\Phi_i[k]$, and a state matrix $X_i[k]$ at time step k. $x_i[k]$ is the state of node i, the elements of $\Phi_i[k]$ are the other nodes' initial states that node i accepted, and the elements of $X_i[k]$ are

the other nodes' initial states that node i received from its neighbors at time step k. Each regular node only has access to its neighbors' state vector information $\Phi_j\,[k], j \in \mathcal{N}_i$, and does not know anything about the rest of the network except the assumption mentioned below.

Assumption 1: Regular nodes know an upper bound m for the number of nodes (including both regular and adversarial) in the neighborhood and an upper bound n for the number of nodes (including both regular and adversarial) in the network.

Note that these upper bounds m and n are defined for preferred implementations where fixed-size static memory vectors are used instead of variable-sized memories. That means this assumption is not restricted, and the exact value m or n is not needed for regular nodes to calculate functions or update states.

State vector $\Phi_j\,[k-1]$, which node i gets from its neighbor node j at time step k, is expressed as

$$\Phi_j\,[k-1] = [x_{j,1}\,[k-1], x_{j,2}\,[k-1], \ldots, x_{j,l} \times [k-1], \ldots, x_{j,n}[k-1]] \quad (1)$$

where $x_{j,l}\,[k-1]$ is the initial state value of node l that node j holds at time step $k-1, l \in \{1, 2, \ldots, n\}, j \in \mathcal{N}_i, k \geq 1$. The state vector $\Phi_j\,[k-1]$ will be stored in state matrix $X_i\,[k]$, which is represented as

$$X_i\,[k] = \begin{bmatrix} \Phi_{j_1}\,[k-1] \\ \Phi_{j_2}\,[k-1] \\ \cdots \\ \Phi_{j_m}\,[k-1] \end{bmatrix} \quad (2)$$

for $j_1, j_2, \ldots, j_m \in \mathcal{N}_i$. Substituting $\Phi_j\,[k-1]$ into this expression, $X_i\,[k]$ can be expressed as

$$X_i\,[k] = \begin{bmatrix} x_{j_1,1}\,[k-1] & x_{j_1,2}\,[k-1] & \ldots & x_{j_1,n}\,[k-1] \\ x_{j_2,1}\,[k-1] & x_{j_2,2}\,[k-1] & \ldots & x_{j_2,n}\,[k-1] \\ \cdots & \cdots & \cdots & \cdots \\ x_{j_m,1}\,[k-1] & x_{j_m,2}\,[k-1] & \ldots & x_{j_m,n}\,[k-1] \end{bmatrix} \quad (3)$$

$X_i\,[k]$ contains all state information that node i received from all neighbors at time step k. For an easy description of the state value acceptance function Θ, the expression of $X_i\,[k]$ can be written in a column vector form as

$$X_i\,[k] = [\Psi_1\,[k], \Psi_2\,[k], \ldots, \Psi_l\,[k], \ldots, \Psi_n\,[k]] \quad (4)$$

where $l \in \{1, 2, \ldots, n\}$ and

$$\Psi_l\,[k] = [\,x_{j_1,l}\,[k-1] \;\; x_{j_2,l}\,[k-1] \;\; \ldots \;\; x_{j_m,l}\,[k-1]\,]^T. \quad (5)$$

B. ALGORITHM DESIGN

We assume that there are no communication delays in the network. At each time step k, each regular node i receives the data packets $\Phi_j\,[k-1]$ related to time instant $k-1$ from neighbors and stores them in $X_i\,[k]$. The data packets are marked with the identity of the nodes and the time steps. In this paper, we do not consider the situation in which adversarial nodes use a fake identity to send values, since this can be solved by cryptographic techniques.

Adversarial nodes may send different false values to their neighbors. To prevent this, strategies are considered and a majority vote is employed. At time step $k = 1$, regular nodes accept only the initial states of neighbors. At time step $k > 1$, each regular node i accepts only the values that are sent by at least $f + 1$ neighbors and saves these values with the corresponding identity tag l in $x_{i,l}[k]$. We use Θ to represent the majority vote. As the algorithm is running, each regular node i fills its state vector $\Phi_i\,[k]$ with more initial state values tagged with new identities. The state value accepting rule is defined as

$$x_{i,l}\,[k] = \Theta(\Psi_l\,[k]) \quad (6)$$

where $l \in \{1, 2, \ldots, n\}$. $x_{i,l}\,[k]$ is the initial state value of node l that is accepted and stored in $\Phi_i\,[k]$ by node i. We obtain

$$\Phi_i\,[k] = [\Theta(\Psi_1\,[k]), \Theta(\Psi_2\,[k]), \ldots, \times \Theta(\Psi_l[k]), \ldots, \Theta(\Psi_n\,[k])] \quad (7)$$

Now, $\Phi_i\,[k]$ can be calculated from $X_i\,[k]$. (7) can be written as

$$\Phi_i\,[k] = \Theta(X_i\,[k]) \quad (8)$$

The values in state vector $\Phi_i\,[k]$ update only when new state values have been accepted by node i according to the accepting rule. Each regular node will act as a source node and a transmitter. The information propagates in the network somewhat similarly to a ''flood''. The nodes in the center of the network will achieve resilient distributed retrieval earlier than the edge nodes. The time steps for different nodes to achieve resilient distributed retrieval are different. Therefore, it is meaningless to make a broadcast if the values in $\Phi_i\,[k]$ have not been updated, since there is nothing new to share with neighbors. Event-triggered broadcasting is employed to solve this problem. We define the trigger condition as

$$\Phi_i\,[k] \neq \Phi_i\,[k-1] \quad (9)$$

where $k > 1$. At time step k, each node i compares the values in $\Phi_i\,[k]$ and $\Phi_i\,[k-1]$, then broadcasts $\Phi_i[k]$ to neighbors only when condition (9) is satisfied. This checking process relies on a global clock embedded in each node, and each regular node checks only once at each time step k. Thus, Zeno behavior is excluded.

Now, we will give a detailed description of E-SABA. Inspired by SABA, E-SABA is designed as follows:

At time step $k = 0$, each node i has an initial state value $x_i[0]$, a state vector $\Phi_i\,[0]$, and a state matrix $X_i[0]$. We set

$$\Phi_i\,[0] = [null, null, \ldots, x_{i,i}\,[0], \ldots, null]_{1*n} \quad (10)$$

where $x_{i,i}\,[0] = x_i\,[0]$, $x_{i,i}\,[0]$ denotes the state value of node i that node i holds at time step $k = 0$, and n is an upper bound

for the number of nodes in the network. We set

$$X_i[0] = \begin{bmatrix} null \\ null \\ \cdots \\ null \end{bmatrix}_{m*1} \tag{11}$$

Then, each node i broadcasts $\Phi_i[0]$ to its neighbors.

At time step $k = 1$, each node i receives $\Phi_j[0]$ from its neighbors, updates $X_i[1]$, accepts $x_{j,j}[0]$ from $\Phi_j[0]$, and updates the state value in $\Phi_i[1]$ with $x_{i,j}[1] = x_{j,j}[0]$. Then, each node i broadcasts $\Phi_i[1]$ to its neighbors.

At time step $k > 1$, each node i does the following:

1) Copy: make a copy of $\Phi_i[k-1]$ and $X_i[k-1]$, and rename them as $\Phi_i[k]$ and $X_i[k]$, respectively.

2) Receive: receive $\Phi_j[k-1]$ from neighbors, $j \in \mathcal{N}_i$, and store them in $X_i[k]$.

3) Accept: calculate $\Phi_i[k] = \Theta(X_i[k])$, and store the new accepted values in $\Phi_i[k]$.

4) Broadcast: broadcast $\Phi_i[k]$ to all neighbors if $\Phi_i[k] \neq \Phi_i[k-1]$.

Regular nodes never stop executing this algorithm until a lower-bounded number of time steps denoted by $K_{\max}$ is reached. The calculation method of $K_{\max}$ is given in the next section.

With this algorithm, each regular node will act as a source node and a transmitter. The information sets that are broadcasted between nodes contain not only their initial state values but also the initial state values of the others that they have accepted. The pseudocode of E-SABA can be seen in Algorithm 1.

Remark 1: Unlike SABA, in E-SABA, first, the node broadcasts its state vectors only at a certain time event when the state vector is updated. At each time step k, each node checks its state vector $\Phi_i[k]$ to determine whether it has been updated. The node broadcasts its state vector to its neighbors if this condition is satisfied and otherwise does not. With the help of this event-triggered update mechanism, we find that there are fewer nodes that broadcast state vectors during the information exchange period. Second, after receiving state vectors from neighbors, the node stores them in the state matrix $X_i[k]$. Before the state value accepting process, node i checks the elements of the state vector $\Phi_i[k]$ to determine whether they are null. If the element $x_{i,j}[k]$ is not null, this means that the state value of node j has been accepted by node i. If $x_{i,j}[k]$ is null, then the algorithm goes to the state value acceptance process. This mechanism reduces the expense of computation. Third, the state value acceptance function Θ is also optimized. When dealing with a certain state value received from neighbors, node i can accept an identical value and end the loop once it has counted $f+1$ copies from different neighbors, and it continues to deal with the next state value from neighbors that it may accept. Since there are many more than $f+1$ neighbors of one node, this mechanism also reduces the expense of computation.

Remark 2: The goal of E-SABA is for nodes to gather enough initial values of other nodes and then calculate certain functions that depend on these initial values. In E-SABA, each regular node acts as a source node and a transmitter. Consider a regular node s that holds the initial value $x_s[0]$ at time step $k = 0$. According to the update rule of E-SABA, it will take at most $K_{\max}$ time steps for any regular node in the network to accept $x_s[0]$. For a single node s, $K_{\max}$ denotes the upper bound of time steps. However, while each node acts as a source node, E-SABA needs to execute at least $K_{\max}$ time steps for each regular node in order to ensure that all regular nodes have accepted the initial values of the others. This shows that $K_{\max}$ is a lower bound of the number of time steps for E-SABA.

Algorithm 1 : Event-Triggered Secure Acceptance and Broadcasting Algorithm (E-SABA)

Initialization: Each regular node i holds an initial state value $x_i[0]$, a state vector $\Phi_i[0]$, and a state matrix $X_i[0]$ at time step $k = 0$, then broadcasts $\Phi_i[0]$ to all its neighbors.
If $k = 1$ **then**
C: make a copy of $\Phi_i[0]$ and $X_i[0]$, and rename them as $\Phi_i[1]$ and $X_i[1]$, respectively.
R: receive $\Phi_j[0]$ from neighbors, store them in $X_i[1]$.
A: accept $x_{j,j}[0]$ in $\Phi_j[0]$ and update $\Phi_i[1]$, where $x_{i,j}[1] = x_{j,j}[0]$, $j \in \mathcal{N}_i$.
B: broadcast $\Phi_i[1]$ to all neighbors.
end if
If $k > 1$ **then**
C: make a copy of $\Phi_i[k-1]$ and $X_i[k-1]$, and rename them as $\Phi_i[k]$ and $X_i[k]$, respectively.
R: receive $\Phi_j[k-1]$ from neighbors, store them in $X_i[k]$.
For $x_{i,l}[k-1]$ is *null* in $\Phi_i[k-1]$ **do**
If state value acceptance $\Theta(\Psi_l[k]) \neq null$ **then**
A: accept the result value of $\Theta(\Psi_l[k])$ and save it in memory $\Phi_i[k]$.
end if
end for
If $\Phi_i[k] \neq \Phi_i[k-1]$ **then**
B: broadcast $\Phi_i[k]$ to all neighbors.
end if
end if
Result: $\Phi_i[k] = [x_{i,1}[k], x_{i,2}[k], \ldots, x_{i,n}[k]]$

C. NETWORK TOPOLOGY ANALYSIS FOR E-SABA

For SABA, the following result from [31] provides a sufficient condition for each regular node in the network to eventually accept the initial states of all the other regular nodes.

Theorem 1 ([31]): Each node $i \in R$ in the network $\mathcal{G}$, by executing SABA for $\bar{K} \geq N-2$ steps, will retrieve $x_l[0]$, $l \in \mathcal{R}\backslash\{i\}$, under the f-local adversarial model if $\mathcal{G}$ is strongly $(2f+1)$-robust.

This is only a sufficient condition. We will now provide a different sufficient condition and a necessary condition for

SABA and E-SABA to succeed in terms of network topology. We will first introduce a sufficient condition with the concept of an f-propagation graph.

Theorem 2: Consider a time-invariant network modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node updates information by executing E-SABA with parameter f. Under the f-local adversarial model, resilient distributed retrieval is achieved if the network topology is an f-propagation graph.

Proof: Consider a partition S_1, S_2, F_1, F_2 of $\mathcal{V}$ such that S_1 and S_2 are nonempty and composed of regular nodes and the nodes in F_1 and F_2 behave as if they are adversarial. $S_L = S_1 \cup F_1$, $S_R = S_2 \cup F_2$, $F = F_1 \cup F_2$. Assume that some node $s \in S_1$ holds an initial state value x_s [0]. According to the rules in E-SABA, there are two ways for any regular node $i \in S_2$ to accept x_s [0]. One is that node i receives the identical value x_s [0] from at least $f + 1$ distinct neighbors and then accepts x_s[0]. The other is that node i is a neighbor of s and it accepts x_s [0] at time step $k = 1$.

We use contradiction to prove that all other regular nodes will receive and accept x_s [0]. Suppose that E-SABA fails in the given network so that some regular nodes fail to accept x_s [0]. Let S_1 denote the regular nodes that accept x_s [0], and let S_2 denote the regular nodes that fail to accept x_s [0].

According to condition (i) of the f-propagation graph, we know that some regular node i in S_R must have at least $2f + 1$ neighbors in S_L. Under the f-local adversarial model, at most f of these nodes that are in F_2 can be adversarial. Thus, all the other nodes are regular nodes in S_1 that have accepted x_s[0] and rebroadcast it to node i; then, node i will receive and accept x_s [0]. Fig. 2(a) illustrates the sets used in this proof.

If condition (i) of the f-propagation graph is not satisfied, according to condition (ii) of the f-propagation graph, we know that any regular node s must have at least $f + 1$ neighbors in S_R. Under the f-local adversarial model, at most f of these nodes that are in F_2 can be adversarial. Node s must have a regular neighbor node i in S_2; then, regular node i will receive and accept x_s [0]. Fig. 2(b) illustrates the sets used in this proof.

In either case, this contradicts the assumption that regular nodes in S_2 fail to receive and accept x_s [0]. ■

Note that if the condition of either Theorem 1 or Theorem 2 is satisfied, E-SABA and SABA will also succeed under the f-local adversarial model. Finally, the proposition below shows that E-SABA and SABA succeed in certain networks that do not satisfy the condition proposed in Theorem 1.

Proposition 1: For some f, there exist graphs that are not strongly $(2f + 1)$-robust but are f-propagation graphs.

Proof: For $f = 1$, construct an undirected graph $\mathcal{G}$ as follows: Start with a complete graph of four nodes, denoted as 2, 3, 4, 5. Add node 1 connected to 2, 3, 4. Add node 6 connected to 3, 4, 5. Add node 7 connected to 4, 5, 6. Add node 8 connected to 5, 6, 7. This example can be seen in Fig. 1. Take nodes 1, 8 as subset S_1, and take the other nodes as subset S_2. Assuming that the nodes in S_1 and S_2 are all regular, it is easy to check that the initial state of the nodes in S_1 can be accepted by the nodes in S_2. S_2 does not satisfy the conditions of strongly $(2f+1)$-robust graph, but graph $\mathcal{G}$ is an f-propagation graph. ■

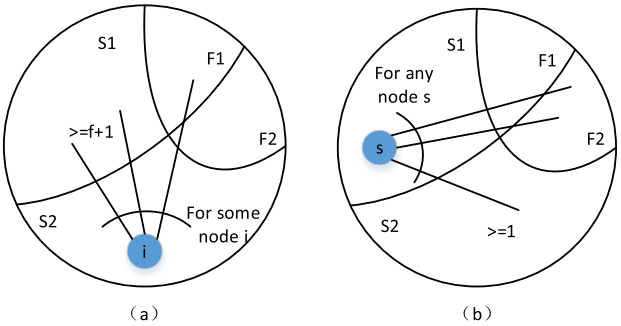

FIGURE 2. Illustration for the proof of theorem 2.

Although the condition in Theorem 2 is sufficient, it is not necessary since the f-local adversarial model gives only an upper bound f for adversaries in each regular node's neighborhood. Next, we give a necessary condition.

Theorem 3: Consider a time-invariant network modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which each node updates information by executing E-SABA with parameter f. Under the f-local adversarial model, the network topology is a regular f-propagation graph if resilient distributed retrieval is achieved.

Proof: Consider S_L and S_R, which are disjoint subsets of $\mathcal{V}$ such that S_L and S_R are nonempty and composed of both regular and adversary nodes. Assume that regular node $s \in S_L$ holds the initial value x_s [0].

By assumption, E-SABA is correct in a given network under the f-local adversarial model. Thus, all the regular nodes will accept x_s [0] eventually. Let $r(r > 0)$ be the earliest time step at which at least one of the regular nodes in S_R accepts x_s [0]. Let node i be one of the regular nodes in S_R that accepts x_s [0] at time step r. Such a regular node i must exist since S_R is nonempty and composed of both regular and adversary nodes. For regular node i to accept x_s [0], according to the update rule of E-SABA, either node i must be the neighbor of s that directly accepts x_s [0] at time step $k = 1$ or node i must have $f + 1$ distinct neighbors that have already accepted x_s [0]. For the former situation, node s must have a neighbor in S_R; this corresponds to condition (ii) of a regular f-propagation graph. For the latter situation, these $f+1$ nodes that have already accepted x_s [0] before node i must be outside S_R by time step r; this corresponds to condition (i) of a regular f-propagation graph. ■

Remark 3: From Theorem 2 and Theorem 3, we know that E-SABA is correct in a given network if the network topology satisfies certain conditions. The sufficient condition and the necessary condition are not the same since the f-local adversarial model gives only an upper bound f for adversaries in each regular node's neighborhood. These conditions are

closely related to the actual number of adversaries in each regular node's neighborhood. In some special cases, the sufficient and necessary conditions may coincide. For example, the conditions for the network topology in Theorem 2 are sufficient and necessary for E-SABA to succeed when each regular node has f adversarial nodes in its neighborhood. The case is similar for Theorem 3 when each regular node has no adversarial nodes in its neighborhood.

Remark 4: Since we do not know which nodes in the network may be adversarial, the concept of the f-propagation graph describes what the network topology of all nodes should be for E-SABA to succeed. These conditions are conducive to actual implementation, such as building the network topology by adding new nodes to a known f-propagation graph.

Next, we investigate the time step $K_{\max}$ for regular nodes to stop executing the algorithm in E-SABA when the regular nodes are programmed to run the algorithm for a lower-bounded number of steps denoted by $K_{\max}$.

D. CALCULATING $K_{\max}$

First, we investigate some properties of the nodes and their neighbors in the f-propagation graph under the f-local adversarial model.

Theorem 4: In a given network $\mathcal{G}$ of N nodes, $N > 2f+1$, $N \in \mathbb{Z}^+$. Under the f-local adversarial model, each node (including both regular and adversarial nodes) has at least $2f+1$ neighbors (including both regular and adversarial) if $\mathcal{G}$ is an f-propagation graph.

Proof: Consider any node $s \in \mathcal{V}$; we obtain a subset $S_1 = \{s\}$ such that $S_1 \subseteq \mathcal{V}$. Since $\mathcal{G}$ is an f-propagation graph, subset S_1 must satisfy at least one of the two conditions of the f-propagation graph. For condition (i), the only node s in S_1 must have at least $2f+1$ neighbors outside. For condition (ii), any nodes outside S_1 must have neighbors in S_1; since $N > 2f+1$, $N \in \mathbb{Z}^+$, this means node s has $(N-1) \geq 2f+1$ neighbors. In either case, node s must have at least $2f+1$ neighbors. ■

Proposition 2: In a given network $\mathcal{G}$ of N nodes, $N > 2f+1$, $N \in \mathbb{Z}^+$. Under the f-local adversarial model, each regular node has at least $f+1$ regular nodes as neighbors if $\mathcal{G}$ is an f-propagation graph.

Proof: From Theorem 4, we know that any node has at least $2f+1$ neighbors; this also means that any regular node has $2f+1$ neighbors. Under the f-local adversarial model, we know that each regular node has at most f adversarial nodes among its neighbors. So, each regular node has at least $f+1$ regular nodes as neighbors if $\mathcal{G}$ is an f-propagation graph. ■

Proposition 3: In a given network $\mathcal{G}$ of N nodes, $N > 2f+1$, $N \in \mathbb{Z}^+$. Under the f-local adversarial model, each adversarial node has at least $f+1$ regular nodes as neighbors if $\mathcal{G}$ is an f-propagation graph.

Proof: Consider S_1 and S_2, which are disjoint subsets of $\mathcal{V}$ such that S_1 and S_2 are nonempty, S_1 is composed of regular nodes and S_2 is composed of adversarial nodes. Now, consider that the nodes in S_1 and S_2 must satisfy at least one of the two conditions in the f-propagation graph. However, under the f-local adversarial model, any regular node $s \in S_1$ has no more than f neighbors in S_2 since S_2 is composed of adversarial nodes; this means S_1 cannot satisfy condition (i). Any node outside S_1 must have at least $f+1$ neighbors in S_1 when condition (ii) is satisfied. We conclude that any adversarial nodes in S_2 must have at least $f+1$ neighbors in S_1. ■

These theorems and propositions concern the adjacency relationship among nodes of different natures (regular or adversary nodes) in the f-propagation graph under the f-local adversarial model. Theorem 4 is used in calculating $K_{\max}$. Proposition 2 and Proposition 3 are used in adversary detection and explain why adversary nodes can be detected and broadcasted through the network. Based on these properties, $K_{\max}$ is given as follows:

Theorem 5: In a network of N nodes that is an f-propagation graph, $N > 2f+1$, $N \in \mathbb{Z}^+$. Under the f-local adversarial model, each regular node will accept the initial states of the other regular nodes if E-SABA is executed for $K_{\max} \geq (N-2f-1)$ time steps.

Proof: Consider a regular node s that holds the initial value $x_s[0]$ at time step $k=0$. According to the update rule of E-SABA, regular node j will receive and accept $x_s[0]$ at $k=1$, where $j \in \mathcal{N}_s$. Now, consider a regular node $i \in \mathcal{N}_j$, where $j \in \mathcal{N}_s$; i.e., i is a neighbor of a neighbor of node s or a so-called 2-hop neighbor of node s. At time step $k=2$, node i receives $x_s[0]$ from node j for the first time. However, node i cannot decide whether to accept $x_s[0]$ sent by node j until it receives $x_s[0]$ from at least f other nodes. In the worst case, $x_s[0]$ has to be passed through all the other nodes that have not accepted $x_s[0]$ at time step $k=1$, which are all the nodes in $\mathcal{V}\backslash\{s, \mathcal{N}_s\}$. From Theorem 4, we know that $|\mathcal{N}_s| \geq 2f+1$, so it will take $N-(2f+1)$ steps at most for node i to accept $x_s[0]$. Note that each node i does not know exactly when to stop executing E-SABA, and thus keeps executing it until $K_{\max}$ to ensure that each regular node has accepted the initial states of the other regular nodes. ■

IV. ADVERSARY DETECTION

The value $\bar{x}$ to be calculated in resilient average consensus is only the average of the initial state values of regular nodes, which is despite the engagement of adversarial nodes in the process. To reach a resilient average consensus for the network, each regular node not only needs to obtain the initial state values of the other nodes but also needs to know the adversarial node set. Thus, regular nodes remove the initial state values of adversarial nodes in the function calculation. In E-SABA, the adversarial nodes are assumed to broadcast their initial state values to neighbors at time step $k=0$, and these initial values will broadcast through the network and be accepted by regular nodes. However, the initial values of the adversarial nodes are also suspicious or not reliable, as the adversarial nodes behave abnormally in the network. Now, we provide a method to detect the adversarial nodes

according to their behavior and broadcast the adversarial node set through the network.

Note that through the procedure described in E-SABA, each node i receives state information $\Phi_j\ [k-1]$ from its neighbors and stores them in $X_i\ [k]$ at time step k. All nodes are programmed to calculate the state set $\Phi_i\ [k]$ that they will accept from $X_i[k]$. Consider the following scenarios:

Detection (1): An adversarial node s may broadcast a false constant state value $x_{s,l}\ [k] = a, l \in \{1, 2, \ldots, n\}$, to node i at time step $t = k > 0$, and the value $x_{s,l}\ [k]$ will be stored in $X_i[k+1]$ by node i. In E-SABA, all regular nodes are programmed to send the accepted constant value to neighbors, and $x_{s,l}\ [k]$ is the initial state value of node l that node s holds at time step k. From Theorem 2 and Theorem 5, we know that all regular nodes can retrieve the initial state value of regular node l earlier than or at $K_{\max}$. From Proposition 2, we also know that each regular node has at least $f+1$ regular nodes as neighbors. Therefore, regular node i will accept the correct value $x_l\ [0] \neq a$ from neighbor l at time step $t = 1$ or $x_{j,l}\ [k] \neq a, j \in \mathcal{N}_i$, from at least $f+1$ regular neighbors at some time step $1 < t \leq K_{\max}$. Then, it checks the state value in $X_i[k+1]$ and detects node s as an adversarial node.

Detection (2): An adversarial node s may broadcast a value $x_{s,l}[k] = a, l \in \{1, 2, \ldots, n\}$, to node i at time step $t_1 = k$ and change it at some time step $t_2 > k$. According to E-SABA, all regular nodes are programmed to send the accepted constant values to neighbors until time step $K_{\max}$. Thus, node i will detect node s as an adversarial node when checking the state values in $X_i\ [t_2]$ and $X_i\ [t_2-1]$ at time step t_2.

Note that regular node i will be able to detect adversarial nodes among neighbors and mark them as adversarial nodes in E-SABA, with extra memory buffers $X_i\ [k]$ and $X_i\ [k-1]$ to track the adversarial behaviors. Now we propose an ADA, and the pseudocode of the ADA can be seen in Algorithm 2.

Theorem 6: Each regular node $i \in \mathcal{R}$ in the network $\mathcal{G}$, by executing the ADA for $2*K_{\max}$ time steps, will detect any adversarial node s in the E-SABA process, $s \in \mathcal{A}$, under the f-local adversarial model if $\mathcal{G}$ is an f-propagation graph.

Proof: From Theorem 2 and Theorem 4, we can easily obtain that the initial state value $x_s\ [0]$ of node s will be retrieved by all the other nodes at time step $K_{\max}$ if at least $f+1$ neighbors of node s accept $x_s\ [0]$ and broadcast in the E-SABA process. The information broadcasting and accepting rules in the ADA are the same as in E-SABA.

Consider an adversarial node s that broadcasts false state values to neighbors. From Proposition 3, we know that each adversarial node has at least $f+1$ regular node neighbors. If at least $f+1$ regular neighbors of node s detect it as an adversary and broadcast according to the rules, all regular nodes in the network will accept node s as an adversary by executing the ADA. If at most f regular neighbors of node s detect it as an adversary and broadcast according to the rules, the other regular nodes in the network will not accept node s as an adversary according to the rule in the ADA. In this case, the neighbors of node s will be corrected by removing the adversary mark of node s at time step $k = 2*K_{\max}$. Thus, all regular nodes in the network are consistent in accepting node s as an adversary or not an adversary in the ADA.

Algorithm 2 : Adversary Detection Algorithm (ADA)

Initialization: Each regular node i has an adversary node set $\Omega_i\ [0] = [null]_{1*n}$
If $1 < k < 2*K_{\max}$ **then**
  R: receive $\Omega_j\ [k-1]$ from neighbors, $j \in \mathcal{N}_i$.
  If receive adversary marks of node l in $\Omega_j\ [k-1]$ from $f+1$ neighbors, **then**
    A: accept node l as an adversary in $\Omega_i\ [k]$.
  end if
  If $1 < k \leq K_{\max}$ **then**
    If Detection (1) is true **or** Detection (2) is true **then**
      D: mark node s as an adversary in $\Omega_i\ [k]$.
    end if
  end if
  If $\Omega_i\ [k]$ is updated **then**
    B: broadcast $\Omega_i\ [k]$ to all neighbors.
end if
end if
  If $k = 2*K_{\max}$ **then**
    If not receive marks of node s as an adversary from $f+1$ neighbors in $\Omega_j\ [k-1]$, **then**
      Unmark node s as an adversary in $\Omega_i\ [k]$.
end if
Result: $\Omega_i\ [k]$

In the ADA, we only consider detecting the adversary nodes during the E-SABA execution time, which is $1 < k \leq K_{\max}$. However, it will take another $K_{\max}$ time steps at most for all regular nodes to accept the result. Thus, it will take $2*K_{\max}$ time steps at most for all regular nodes to detect any adversarial node s in the network. ■

Remark 5: We describe the sets $\Phi_i\ [k]$ and $\Omega_i\ [k]$ separately for ease of understanding in this paper. In fact, the sets $\Phi_i[k]$ and $\Omega_i\ [k]$ can be merged during implementation by using a strategy such as the following:

$$x_{i,l}\ [k] = \begin{cases} null, & initial \\ value, & acceptedvalue \\ FF, & adversary \end{cases} \tag{12}$$

where FF is a specified value or flag for marking the node as adversarial in the set. Thus, the memory expense is reduced. We can observe that node i uses only sets from neighbors and calculates according to the detection rules while executing the ADA. The computational complexity is clearly $O(n)$ for each node i. However, the algorithm needs to be executed for more time steps (at most $K_{\max}$); thus, the result can be broadcasted and accepted by regular nodes in the network.

V. RESILIENT AVERAGE CONSENSUS

In this subsection, we suppose that the nodes in the network update their states synchronously. Note that the resilient distributed retrieval process does not involve the node dynamics

or global graph knowledge. We propose a state update function for regular nodes using the accepted initial state values at each time step, and eventually, all regular nodes converge to a consensus asymptotically. We consider two scenarios.

One is that all regular nodes are equipped with E-SABA and achieve average consensus. At each time step k, regular node i, by running E-SABA, receives $\Phi_j\ [k-1]$ from its neighbors and then updates $\Phi_i\ [k]$ when new values are accepted. Node i updates its state at each time step k using the average of the initial state values $x_{i,l}\ [k]$ in $\Phi_i\ [k]$, which are accepted by node i up to time step k. The state update function is defined as

$$x_i\ [k] = \frac{\sum x_{i,l}\ [k]}{|\mathcal{F}_i\ [k]|}, \quad l \in \mathcal{F}_i\ [k] \tag{13}$$

where $\mathcal{F}_i\ [k]$ is the set of indices of the elements in $\Phi_i[k]$ that are nonempty and its cardinality is given by $|\mathcal{F}_i[k]|$.

The other scenario is that all regular nodes are equipped with E-SABA and the ADA and achieve resilient average consensus. At every time step k, regular node i, by running E-SABA and the ADA, receives $\Phi_j\ [k-1]$ and $\Omega_j\ [k-1]$ from its neighbors and then updates $\Phi_i\ [k]$ and $\Omega_i\ [k]$ when new values are accepted. Node i updates its state at each time step k using the average of the initial state values in $\Phi_i\ [k]$ (excluding the initial state values of nodes that are marked in $\Omega_i\ [k]$) that are accepted by node i up to time step k. The state update function is defined as

$$x_i\ [k] = \frac{\sum x_{i,l}\ [k]}{|\mathcal{F}_i\ [k]| - |\mathcal{W}_i\ [k]|}, \quad l \in \mathcal{F}_i\ [k] \& l \notin \mathcal{W}_i\ [k] \tag{14}$$

where $\mathcal{F}_i\ [k]$ is the set of indices of the elements in $\Phi_i[k]$ that are nonempty and its cardinality is given by $|\mathcal{F}_i[k]|$; $\mathcal{W}_i\ [k]$ is the set of indices of the elements in $\Omega_i\ [k]$ that are nonempty, and its cardinality is given by $|\mathcal{W}_i\ [k]|$.

Remark 6: Note that the resilient distributed retrieval process does not involve the node dynamics or global graph knowledge. The information shared between nodes while the algorithms are executing is the state value vector $\Phi_i\ [k]$ and adversary node set $\Omega_i\ [k]$. Each regular node updates the state values $x_i\ [k]$ according to (14) but does not share the updated state values $x_i\ [k]$ with neighbors. The state update function (14) can also be expressed in another format as a controlled system. However, both formats have the same meaning while solving the average consensus problem in this paper.

Remark 7: The solution for the resilient average consensus problem is described in three parts: the resilient distributed retrieval of the initial state values, the adversary detection process, and the state update rule each node has to execute. First, E-SABA is for initial state information diffusion between nodes, which is described as resilient distributed retrieval. Second, the ADA is for regular nodes to detect Byzantine nodes in the set of neighbors, which deviate from the prescribed rules of E-SABA, and share the detection results. Third, regular nodes update their states according to (14) using the average of the initial state values accepted by nodes up to time instant k, converging to an average consensus asymptotically.

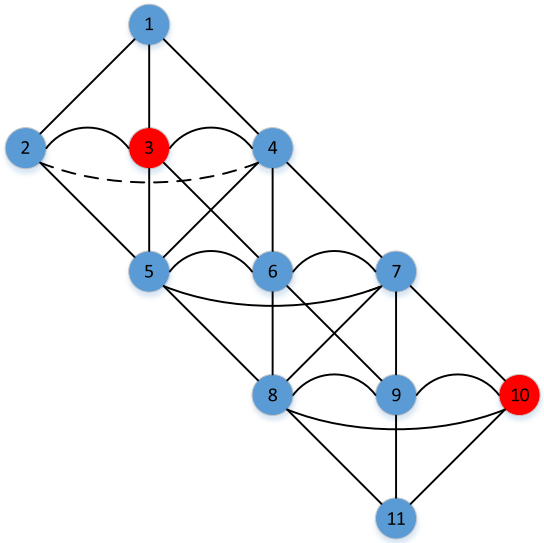

FIGURE 3. A 1-propagation graph with 11 nodes. Nodes 3 and 10 are adversarial nodes. If we remove the edge between 2 and 4, the network fails to reach average consensus since it is no longer an f-propagation graph.

Here, we connect our results together to show how our proposed strategy leads a network with specific topology constraints to resilient average consensus.

Proposition 4: Each regular node $i \in \mathcal{R}$ achieves resilient average consensus by executing E-SABA for $K_{\max} \geq (N - 2f - 1)$ time steps and the ADA for $2K_{\max} \geq 2(N - 2f - 1)$ time steps, performing update rule (14) under the f-local adversarial model if the network topology is an f-propagation graph.

Proof: Nodes are considered to work in an f-propagation graph under the f-local adversarial model. Each regular node updates its state value $x_i\ [k]$ at each time step k according to update rule (14) using the information accepted in $\Phi_i\ [k]$ and $\Omega_i\ [k]$ up to time step k. According to update rule (14), $x_i\ [k]$ is a linear combination of the accepted initial values $x_{i,l}\ [k]$ in $\Phi_i[k]$, excluding the values of nodes in $\Omega_i\ [k]$. As E-SABA (or the ADA) is executing, more values may be accepted in $\Phi_i\ [k]$ (or $\Omega_i\ [k]$). All regular nodes will converge to $\bar{x}$ asymptotically if all regular nodes ultimately have the same values in $\Phi_i\ [k]$ and $\Omega_i\ [k]$. Referring to Theorem 2 and Theorem 5, each regular node will securely retrieve the initial state values of the others (stored in $\Phi_i\ [k]$) if it executes E-SABA for $K_{\max}$ steps. Referring to Theorem 6, each regular node will detect or retrieve any adversarial nodes (stored in $\Omega_i\ [k]$) if it executes the ADA for $2K_{\max}$ steps. Thus, resilient average consensus is achieved. ■

VI. EXAMPLE

In this section, we present a simulation example to illustrate the effectiveness of our proposed algorithm in the presence of adversarial Byzantine attacks. The LCP and W-MSR

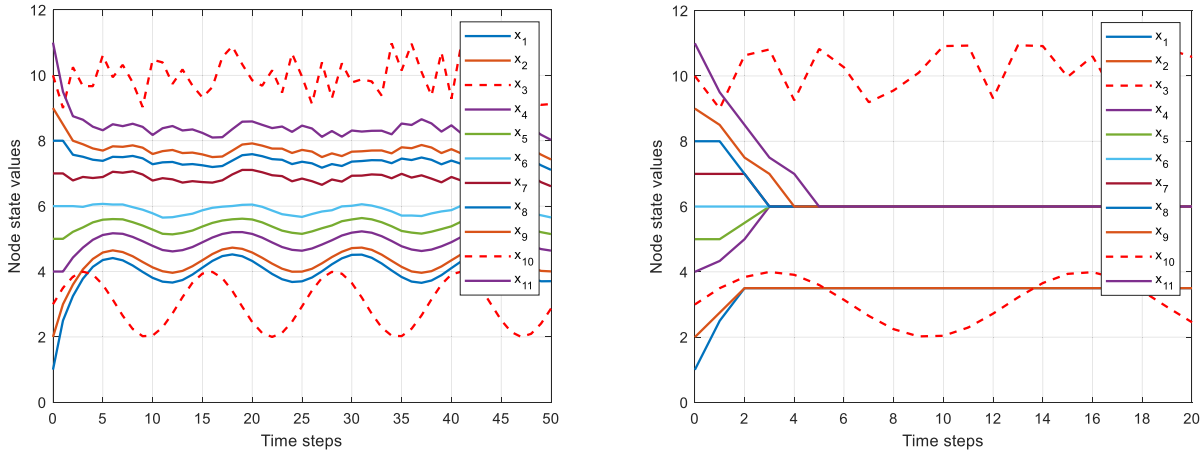

FIGURE 4. The LCP algorithm fails to achieve consensus because of adversarial attacks. Average consensus is not achieved using our proposed algorithm in a network that does not satisfy the f-propagation graph requirements.

algorithm are considered in the comparison of our proposed algorithms.

Consider the network shown in Fig. 3, which is an f-propagation graph with $f = 1$. We assign the initial values of the nodes as $x_i\,[0] = i$, $i \in \{1, 2, \ldots 11\}$, which yields the average value $\bar{x} = 6$. Nodes 3 and 10 are chosen as the adversarial nodes, and the others are regular nodes. Node 3 begins and continues broadcasting the false information set

$$\Phi_3\,[k] = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

to neighbors, and it updates its state following

$$x_3\,[k] = \sin(\frac{k}{2}) + 3$$

when $k \geq 2$. Node 10 begins and continues broadcasting the false information set

$$\Phi_{10}\,[k] = [10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10]$$

to neighbors, and it updates its state to a random value between 9 and 11 when $k \geq 2$. The other regular nodes are assumed to broadcast information sets and update states following the predefined algorithm.

Without any security strategy, the LCP algorithm is very vulnerable to adversarial attacks. This can be seen in Fig 4(a), where the regular nodes update their states with a standard LCP. We can see from the figure that the regular nodes constantly oscillate and cannot reach agreement because of the influence of adversarial nodes 3 and 10.

The network in Fig. 3 is not an f-propagation graph if we remove the edge between nodes 2 and 4. Nodes 1 and 2 cannot accept the state values of regular nodes other than nodes 4 and 5 in E-SABA. This leads to the failure of average consensus even though the regular nodes are equipped with E-SABA and the regular nodes update their states according to function (13) at every time step. The simulated state trajectories are plotted in Fig. 4(b).

Next, we connect nodes 2 and 4. Thus, the network of Fig. 3 is an f-propagation graph. According to Theorem 2, it will achieve resilient distributed retrieval with E-SABA. While regular nodes update their states according to function (13) at every time step, average consensus is achieved. The simulated state trajectories are illustrated in Fig. 5(a), where the regular nodes achieve a consensus of $\bar{x}_i\,[K_{\max}] = \bar{x}$, $i \in \{1, 2, 4, 5, 6, 7, 8, 9, 11\}$. Furthermore, the time steps at which a node broadcasts during E-SABA execution are shown by colored markers • in Fig. 5(b). It is notable that some nodes stop broadcasting much earlier than time step $K_{\max} = 8$. Since some nodes have accepted all the initial states of other regular nodes before $K_{\max}$, $\Phi_i\,[k]$ will not update anymore; thus, broadcasting stops. Nodes 3 and 10 are assumed to broadcast at every time step since their goal is to make more nodes accept their false values and affect the system.

When the regular nodes in the network of Fig. 3 are equipped with both E-SABA and the ADA, they can not only accept the values of regular nodes $\Phi_j\,[k]$ but also detect the adversarial nodes and broadcast the results $\Omega_i\,[k]$ through the network. Regular nodes update their states according to function (14) at every time step. The simulation results are presented in Fig. 6(a) and Fig. 6(b). While the time step $k \leq 2$, the situation in Fig. 6 is similar to that in Fig. 5, where the regular nodes behave according to E-SABA and no nodes have been detected as an adversary. When the time step $k = 2$, adversarial nodes 3 and 10 broadcast false information sets to their neighbors. The neighbors of nodes 3 and 10 will detect nodes 3 and 10 as adversarial at time step $k = 3$ and broadcast this result $\Omega_i\,[k]$ to their neighbors according to the ADA. When the regular nodes accept node 3or 10 as adversarial, the state value of node 3 or 10 is removed from the calculation of state updates. This explains the fluctuation in Fig. 6(a) and the extra broadcast times in Fig. 6(b) compared with those in Fig. 5. Since the state values of adversarial nodes 3 and 10

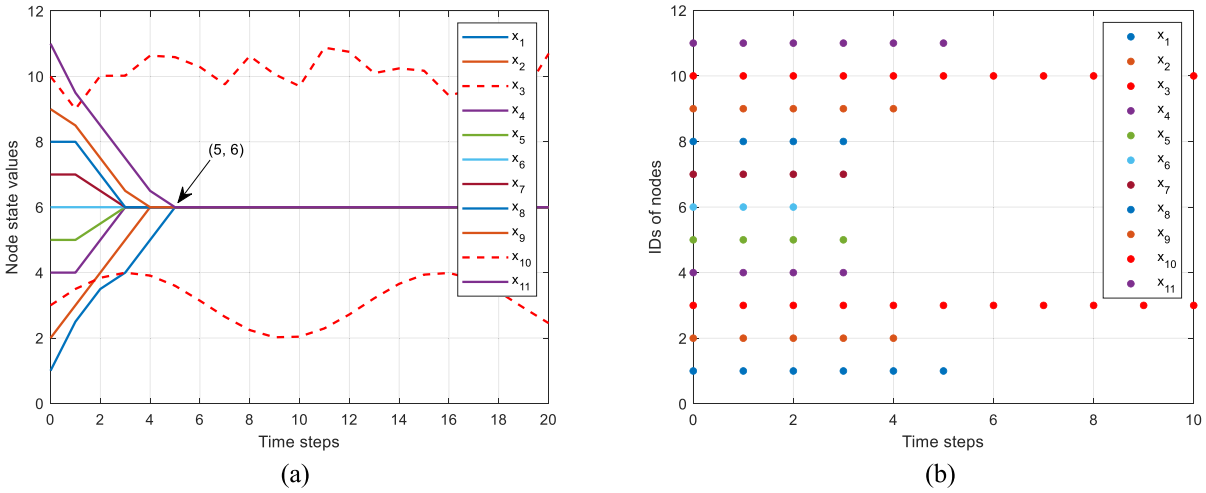

FIGURE 5. Average consensus is achieved using E-SABA in the presence of adversarial nodes 3 and 10 since the network is an f-propagation graph. The time steps for broadcasting are reduced, which is a benefit of the event-triggered mechanism.

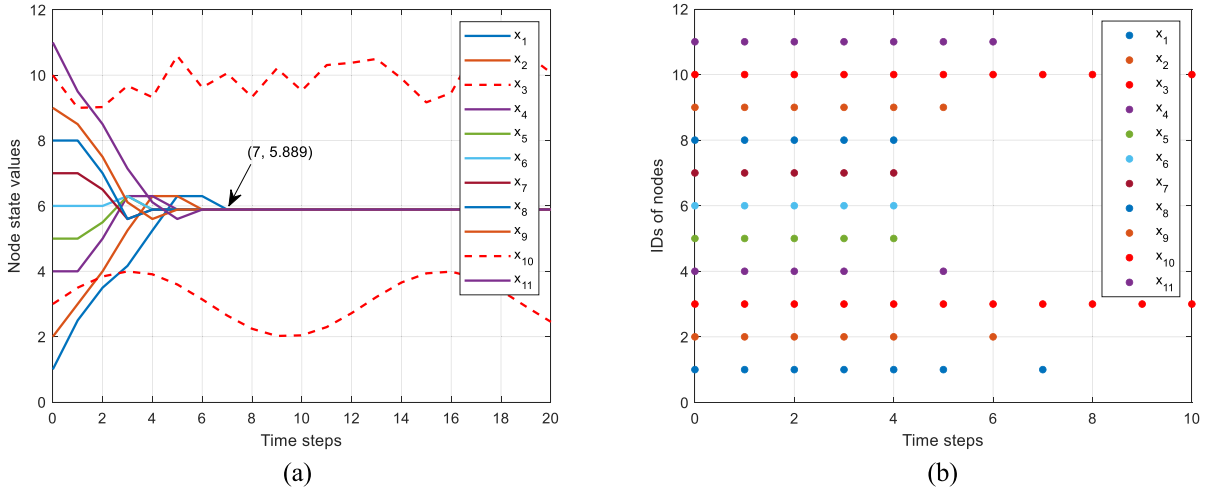

FIGURE 6. Resilient average consensus is achieved using E-SABA and the ADA in the presence of adversarial nodes 3 and 10 since the network is an f-propagation graph. Extra time steps are needed for adversary detection and information broadcasting.

are removed by the regular nodes, the regular nodes finally achieve resilient average consensus.

The performance of the W-MSR algorithm in the network of Fig. 3 can be seen in Fig. 7(a). The W-MSR algorithm achieves an asymptotic consensus despite the adversarial behavior of nodes 3 and 10. In contrast to our proposed algorithm, whose results are shown in Fig. 5(a) and Fig. 6(a), the asymptotic consensus of the W-MSR algorithm is different from the average consensus or resilient average consensus; the influence of the adversarial state value cannot be completely eliminated once it appears, and it finally achieves a consensus of approximately 5.58. Furthermore, the number of time steps that the W-MSR algorithm needs for reaching consensus is much larger than that of our proposed algorithm. The comparison is shown in Fig. 7(b), where nodes 3 and 10 operate as regular nodes in the network. Fig. 6(a) and Fig. 7(a) provide a comparison of our proposed algorithm and W-MSR when nodes 3 and 10 operate as adversarial nodes in the network. In both cases, our proposed algorithms converge to consensus faster with many fewer broadcast times. Comparing Fig. 7(a) and Fig. 7(b), we can see that many more time steps are needed for the W-MSR algorithm, especially when adversarial nodes occur.

More time steps for consensus means more calculation and greater communication expenses since the regular nodes in the W-MSR algorithm broadcast at every time step. Our proposed algorithm only updates states and broadcasts at special time events, as illustrated in Fig. 5(b) and Fig. 6(b). The calculation and communication expenses are thereby reduced.

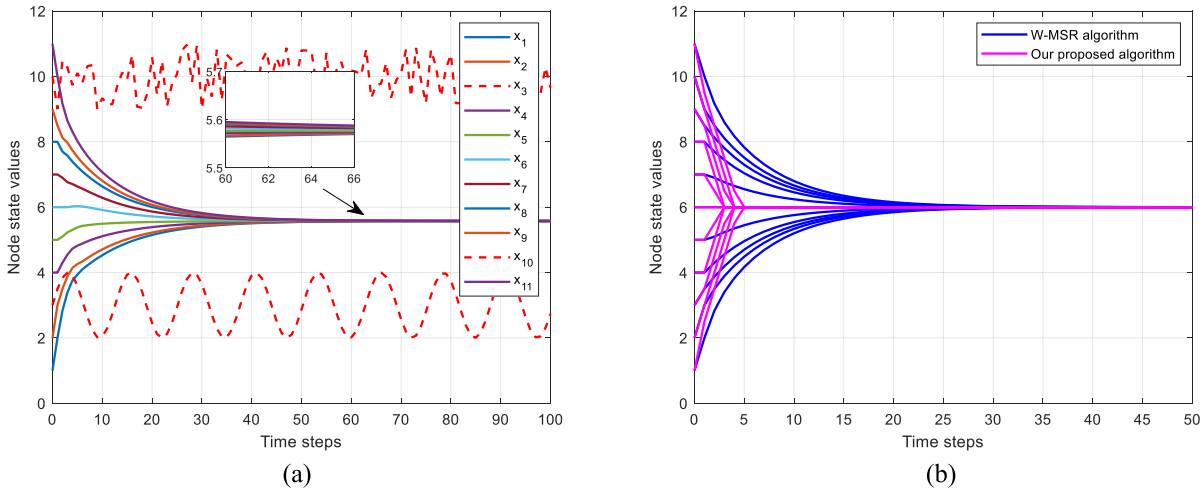

FIGURE 7. Compared with the W-MSR algorithm, our proposed algorithm achieves an exact consensus with fewer time steps, either with or without the presence of adversarial nodes, when the network topology is an f-propagation graph.

VII. CONCLUSION

In this paper, an event-triggered secure acceptance and broadcasting algorithm and an adversary detection algorithm are developed for the resilient average consensus problem in the presence of Byzantine agents. Global knowledge of network topology and high computational capabilities of each regular node are not required in the proposed scheme. Furthermore, computational expense and broadcasting times are reduced, benefitting from the event-triggered mechanism. The sufficient and necessary conditions are provided for the algorithm to succeed under the f-local adversarial model. It is shown that f-propagation graphs are more accurate than r-robust graphs in describing the required topological properties for the algorithms to succeed. Simulation results are provided to verify the effectiveness of the proposed algorithms, and comparisons are made with the LCP and the W-MSR algorithm.

In the future, we will extend the research on the network topology property called the f-propagation graph. Currently, we do not have a computational method to check whether this property holds in arbitrary graphs. We find that the preferential attachment model for complex networks may be helpful in constructing an f-propagation graph. Furthermore, although the objective of this paper is resilient average consensus, the method proposed in this paper can be utilized for more general functional calculation problems without modification, allowing any node to calculate any arbitrary function of the node values in a finite number of time steps. Utilizing the proposed protocols in formation control of unmanned aerial vehicles, distributed filtering, and wireless sensor networks is a topic for future work. There may be delays in communications over the network and nodes may update asynchronously. Additional constraints may be considered when extending the results to fit real applications.

REFERENCES

[1] N. M. Freris and P. Patrinos, "Distributed computing over encrypted data," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 1116–1122.

[2] 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.

[3] Y. Wu and X. He, "Finite-time consensus-based clock synchronization under deception attacks," *IEEE Access*, vol. 8, pp. 110748–110758, 2020.

[4] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1039–1046, Apr. 2017.

[5] H. LeBlanc and X. Koutsoukos, "Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multiagent systems," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1219–1231, Sep. 2018.

[6] D. Ding, Q.-L. Han, X. Ge, and J. Wang, "Secure state estimation and control of cyber-physical systems: A survey," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 176–190, Jan. 2021.

[7] X.-M. Zhang, Q.-L. Han, X. Ge, and L. Ding, "Resilient control design based on a sampled-data model for a class of networked control systems under denial-of-service attacks," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3616–3626, Aug. 2020.

[8] D. Ding, Z. Wang, Q.-L. Han, and G. Wei, "Security control for discrete-time stochastic nonlinear systems subject to deception attacks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 779–789, May 2018.

[9] B. A. Forouzan, *Cryptography & Network Security*. New York, NY, USA: McGraw-Hill, 2007.

[10] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[11] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[12] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 575–600, Jan. 2008.

[13] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 1753–1757.

[14] C. Yu, B. D. O. Anderson, S. Mou, J. Liu, F. He, and A. S. Morse, "Distributed averaging using periodic gossiping," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4282–4289, Aug. 2017.

[15] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[16] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA, USA: Morgan Kaufmann, 1996.

[17] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[18] M. S. Khan, S. S. Naqvi, and N. H. Vaidya, "Exact Byzantine consensus on undirected graphs under local broadcast model," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2019, pp. 327–336.

[19] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.
[20] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.
[21] 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.
[22] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate byzantine consensus in arbitrary directed graphs," in *Proc. ACM Symp. Princ. Distrib. Comput. (PODC)*, 2012, pp. 365–374.
[23] H. Zhang, E. Fata, and S. Sundaram, "A notion of robustness in complex networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 3, pp. 310–320, Sep. 2015.
[24] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 5855–5861.
[25] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 841–848, Apr. 2017.
[26] S. M. Dibaji and H. Ishii, "Consensus of second-order multi-agent systems in the presence of locally bounded faults," *Syst. Control Lett.*, vol. 79, pp. 23–29, May 2015.
[27] S. M. Dibaji and H. Ishii, "Resilient consensus of second-order agent networks: Asynchronous update rules with delays," *Automatica*, vol. 81, pp. 123–132, Jul. 2017.
[28] Y. Shang, "Resilient consensus of switched multi-agent systems," *Syst. Control Lett.*, vol. 122, pp. 12–18, Dec. 2018.
[29] Y. Wang and H. Ishii, "Resilient consensus through event-based communication," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 471–482, Mar. 2020.
[30] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.
[31] S. M. Dibaji, M. Safi, and H. Ishii, "Resilient distributed averaging," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 96–101.
[32] Y. Wu, M. Xu, N. Zheng, and X. He, "Event-triggered resilient consensus for multi-agent networks under deception attacks," *IEEE Access*, vol. 8, pp. 78121–78129, 2020.
[33] A. Rahnama and P. J. Antsaklis, "Learning-based event-triggered control for synchronization of passive multiagent systems under attack," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4170–4185, Oct. 2020.
[34] F. Zegers, P. Deptula, J. M. Shea, and W. E. Dixon, "Event/self-triggered approximate leader-follower consensus with resilience to byzantine adversaries," *IEEE Trans. Autom. Control*, early access, Apr. 21, 2021, doi: 10.1109/TAC.2021.3074849.
[35] F. Zegers, M. Hale, J. M. Shea, and W. E. Dixon, "Event-triggered formation control and leader tracking with resilience to Byzantine adversaries: A reputation-based approach," *IEEE Trans. Control Netw. Syst.*, early access, Mar. 23, 2021, doi: 10.1109/TCNS.2021.3068348.
[36] H. Zhao, X. Meng, and S. Wu, "Distributed edge-based event-triggered coordination control for multi-agent systems," *Automatica*, vol. 132, Oct. 2021, Art. no. 109797.
[37] C.-Y. Koo, "Broadcast in radio networks tolerating Byzantine adversarial behavior," in *Proc. 23rd Annu. ACM Symp. Princ. Distrib. Comput. (PODC)*, 2004, pp. 275–282.
[38] A. Pelc and D. Peleg, "Broadcasting with locally bounded Byzantine faults," *Inf. Process. Lett.*, vol. 93, no. 3, pp. 109–115, Feb. 2005.

PENG ZHANG received the B.S. and M.S. degrees in computer science and technology from Harbin Engineering University, Harbin, China, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with Beihang University, Beijing, China. His research interests include multi-agent networks, resilient consensus control, and fault detection.

CHANGQING HU received the Ph.D. degree in control engineering from China University of Geosciences, Beijing, in 2019. He is currently a Professor of engineering with Beijing Institute of Aerospace Control Devices and the Pilot National Laboratory for Marine Science and Technology, Qingdao. His research interests include the general design and application of unmanned surface vehicles.

SENTANG WU received the Ph.D. degree in dynamics, ballistics, and aircraft motion control systems from the National Aviation University, Ukraine, in 1992. He is currently a Professor of automation science and electrical engineering and a Ph.D. Tutor at Beihang University, Beijing, China. He is also a Navy Missile Expert with the National Defense Basic Research Institute and a member of the academic committee. His research interests include the theory and application of nonlinear stochastic systems, computer information processing and control, aircraft cooperative control, precision, and guidance.

RUIYAN GONG received the M.Eng. degree in design and construction of naval architecture and ocean structure from Harbin Engineering University, China, in 2018. He is currently an Assistant Engineer with Beijing Institute of Aerospace Control Devices. He mainly studies the general design and application of unmanned surface vehicles.

ZIMING LUO received the B.S. degree from Harbin Engineering University, in 2017, and the M.S. degree from Shijiazhuang Communication Observation and Control Technology Institute, in 2020. She is currently an Assistant Engineer with CETC 54. Her research interests include network management and network security.