Guidance, Navigation and Control

Self-triggering Secure Consensus Against Adversarial Attacks

Zirui Liao
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
Shenyuan Honors College, Beihang University, Beijing 100191, China
by2003110@buaa.edu.cn

Jian Shi
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
shijian@buaa.edu.cn

Shaoping Wang*
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
Tianmushan Laboratory, Hangzhou 310023, China
Beihang Ningbo Research Institute, Ningbo 315800, China
shaopingwang@buaa.edu.cn

Yuwei Zhang
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
zhangyuwei@buaa.edu.cn

Rui Mu
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
ruimu@buaa.edu.cn

Zhiyong Sun
College of Engineering, Peking University, Beijing 100091, China
zhiyong.sun@pku.edu.cn

Received 13 October 2024
Revised 17 November 2024
Accepted 21 November 2024
Published Day Month Year

*Corresponding author.

The problem of secure consensus for multi-agent systems (MASs) is tackled in this study. The self-triggering strategy is designed to enable each healthy agent to estimate its next triggering step at the current triggering step. Thus, each healthy agent only needs to sense and broadcast at its triggering steps, and to monitor the latest broadcast states of their neighbors at their triggering steps. The frequent monitoring is thereby mitigated. Subsequently, a self-triggering secure consensus algorithm is developed to guarantee that the state variables of healthy agents reach consensus despite the influence of faulty agents in the network. The convergence analysis of the proposed method is conducted with graph tools and Lyapunov theory. Numerical examples are given to illustrate the superior performance of the proposed self-triggering secure consensus algorithm compared with the existing methods based on the static and dynamic event-triggering mechanisms.

Keywords: Secure consensus; self-triggering mechanism; multi-agent system.

1. Introduction

Over the past decade, the study of consensus-seeking problems for multi-agent systems (MASs) has garnered significant attention due to its wide range of applications in fields such as robotics, multi-UAV systems, and sensor networks.[1–5] In such problems, multiple agents possess individual dynamics, share limited local information, and aim to achieve agreement on certain state variables. However, in practical scenarios, the security problem is critical in the MAS due to its distributed property and simple hardware, as well as being in open environments.[6,7] Specifically, adversarial attacks may be injected into one or several vulnerable nodes and spread over the network. As stated in Ref. 8, even the compromise of a single vulnerable node may ultimately lead to system crash. Therefore, development of algorithms secure to adversarial attack has become increasingly essential.

In the context of secure control for MASs, a widely-used attack model is that the number of faulty agents in the neighbor set of each healthy agent is upper bounded. One attack-tolerant solution to this type of attack is the weighted mean-subsequence-reduced (W-MSR) algorithm.[9] The core idea is that a healthy agent with sufficient neighbors' state variables can perform the state update normally by eliminating the potential information. This seminal work was subsequently extended to miscellaneous system dynamics, network structures, and application scenarios, e.g, higher-order MASs, time-varying networks, and distributed optimization.[10–12] Variants of this kind of algorithm have also been developed from different perspectives, e.g., multidimensional-bipartite-absolute-MSR (MBA-MSR), trusted-edge MSR (TE-MSR), and second-order MSR (S-MSR) algorithms.[10,13,14]

Although the W-MSR algorithm can efficiently eliminate the for healthy agents, it renders the MAS bear heavy communication burden. Specifically, for the purpose of threat elimination and state update, each healthy agent needs to communicate with its neighbors at each time step to obtain their state variables. This operation consumes massive communication and is unnecessary when the state variables of healthy agents approach consensus. To avoid the frequent communication between agents, an event-based MSR (E-MSR) algorithm was proposed in Ref. 15. The introduction of (static) event-triggering mechanism renders the healthy agents

communicate with their neighbors only when the designed triggering function is activated. Subsequently, a dynamic event-triggering MSR (DE-MSR) algorithm was further developed in Ref. 16, where the triggering function is associated with system state, i.e., it can dynamically change as the system state evolves. It has been validated that both of these two mechanisms can save communication resources and reduce communication overheads for MASs.

Despite the effectiveness of the existing event-based secure consensus algorithms in mitigating the communication burden, a common characteristic of them is that they need to frequently monitor the latest broadcast states of neighbors. This is because the healthy agents do not know when their neighbors will trigger. To address these issues, the self-triggering mechanism has emerged as an effective strategy for improving both the security and efficiency of consensus in MAS. Unlike traditional periodic control approaches, which require continuous or frequent communication, self-triggering control allows agents to determine the next triggering step at the current triggering step, thus reducing unnecessary data transmissions and saving system resources. This is particularly useful for systems where energy or bandwidth is limited. The frequent monitoring of neighboring agents' states can also be avoided with the self-triggering mechanism. In Ref. 17 and Ref. 18, the self-triggering strategy is designed to address the average consensus problem in the continuous-time and discrete-time domains, respectively. In hostile environments, Matsume *et al.*[19,20] developed a self-triggering secure consensus algorithm based on the ternary control. However, these two studies require additional clock variables to facilitate the self-triggering strategy and merely achieve approximate secure consensus, i.e., the state variables of healthy agents converge to an error range instead of the consensus value.

Inspired by the aforementioned observations, a self-triggering secure consensus algorithm for MASs is developed in this paper. The algorithm is comprised of the self-triggering part and the attack-tolerant part. Therein, the self-triggering part is designed to render each agent determine its next triggering step at the current triggering step. The attack-tolerant part helps the healthy agents eliminate the potential malicious information received from their neighbors. The convergence analysis of the proposed control protocol is conducted with Lyapunov theory.

The main contributions of this paper are summarized as follows.

(i) The secure consensus problem for MASs with the self-triggering mechanism is addressed in this paper. Compared with Refs. 17, 18 that adopt the self-triggering mechanism to achieve average consensus, we guarantee that the state variables of healthy agents reach consensus within a safety interval, despite the misbehavior of faulty agents in the network. Verifiable sufficient conditions to ensure secure consensus are further derived through a rigorous Lyapunov-function-based approach.
(ii) A self-triggering secure consensus algorithm is designed, which enables the agents in the MAS to calculate the next triggering step at the current triggering step,

thereby avoiding the frequent monitoring to the latest broadcast states of neighbors. Compared with the studies in Refs. 15, 16 that address the secure consensus problem based on the static and dynamic event-triggering mechanisms, the proposed self-triggering strategy can save more communication resources, which is particularly useful in limited energy or bandwidth scenarios.

(iii) Different from the self-triggering secure consensus schemes in Refs. 19, 20 that rely on ternary control to achieve self-triggering control and merely pursue approximate secure consensus, this paper integrates the self-triggering mechanism into controller design, which is more lightweight to implement. Furthermore, the proposed self-triggering secure consensus algorithm achieves exact secure consensus, where the state variables of healthy agents converge to the identical reference value instead of an error range.

The rest of this paper is organized as follows. In Section 2, the self-triggering mechanism is introduced and the secure consensus problem is formulated. In Section 3, the theoretical analysis is conducted to guarantee secure consensus with the proposed self-triggering attack-tolerant algorithm. Then, two comparative numerical results are illustrated and discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2. Problem Formulation

In this section, we present the dynamic model of the MAS with the self-triggering mechanism, the model of adversarial attack, and the formulation of secure consensus problem.

2.1. *Graph theoretical preliminaries*

Consider a multi-agent network modeled by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the directed edge set. The edge $(j, i) \in \mathcal{E}$ represents that agent j can send messages to agent i. Let $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$ be the neighbor set of agent i.

We consider the scenario that the multi-agent network is subject to adversarial attacks, and our strategy is to exploit the data redundancy of the network to counter adversarial attacks. The following definitions quantify the data redundancy with respect to (w.r.t.) sets and graphs, respectively.

Definition 1 (r-reachable set).[9] Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a nonempty subset $\mathcal{S} \subset \mathcal{V}$. The set $\mathcal{S}$ 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 (r-robust graph).[9] A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be r-robust if for each pair of nonempty and disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of them is r-reachable, where $r \in \mathbb{Z}_+$.

2.2. *System model with self-triggering mechanism*

Consider a multi-agent network modeled by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each agent $i \in \mathcal{V}$ has a discrete-time state variable $x_i[t] \in \mathbb{R}$, which adheres to the following update rule:

$$x_i[t+1] = x_i[t] + u_i[t]. \tag{1}$$

The corresponding control input $u_i[t] \in \mathbb{R}$ is designed as

$$u_i[t] = \varepsilon \sum_{j \in \mathcal{N}_i} \theta_{ij}[t](\hat{x}_j[t] - \hat{x}_i[t]), \tag{2}$$

where $\varepsilon \in (0, 1)$ is a control gain and $\hat{x}_j[t]$ denotes the latest broadcast state of agent j at time step t, which is mathematically expressed as

$$\hat{x}_j[t] = x_i[t_h^j], \quad k \in [t_h^j, t_{h+1}^j), \tag{3}$$

where $\{t_0^j, t_1^j, \ldots \in \mathbb{Z}_{>0}\}$ denotes the sequence of triggering steps of agent j. By combining Eqs. (1), (2) and (3), the state variable of agent i at time steps $t \in [t_h^i, t_{h+1}^i)$ can be rewritten as

$$x_i[t] = x_i[t_h^i] + (t - t_h^i)\varepsilon \sum_{j \in \mathcal{N}_i} \theta_{ij}[t](\hat{x}_j[t] - \hat{x}_i[t]). \tag{4}$$

Assumption 1. For all time steps $t \in \mathbb{Z}_{\geq 0}$ and for each healthy agent $i \in \mathcal{H}$, the weight $\theta_{ij}[t]$ satisfies the following conditions.

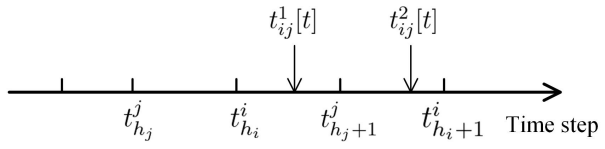
(1) $\theta_{ij}[t] \geq \omega, \ \forall j \in \mathcal{N}_i$, where $\omega \in (0, 1)$;
(2) $\theta_{ij}[t] = 0$ if $j \notin \mathcal{N}_i$;
(3) $\sum_{j=1}^{n} \theta_{ij}[t] = 1$, where $n = |\mathcal{V}|$.

Remark 1. Note that choosing an appropriate control gain ε for Eq. (2) is important to ensure both the convergence rate and the steady performance. When the control gain is too small, the system may experience a slow convergence rate. When the control gain is too large, the state variables of healthy agents may exhibit oscillations.

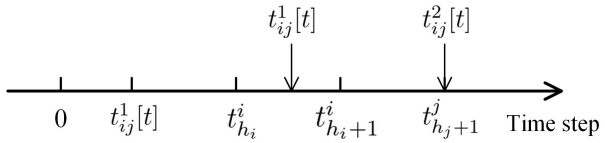
Subsequently, one defines $e_i[t] = \hat{x}_i[t] - x_i[t]$ as the error term and $d_{ij}[t] = \hat{x}_i[t] - \hat{x}_j[t]$ as the difference between the latest broadcast states of agent i and agent j. By invoking Eqs. (1) and (2), the error term $e_i[t]$ can be mathematically expressed as

$$e_i[t] = (t - t_h^i)\varepsilon \sum_{j \in \mathcal{N}_i} \theta_{ij}[t] d_{ij}[t]. \tag{5}$$

Note that $d_{ij}[t]$ may vary for all $t \in [t_h^i, t_{h+1}^i)$, since agent i's neighbors may trigger at any step during this interval. Thus it is difficult to calculate the exact value of $e_i[t]$. However, the upper bound of $e_i[t]$ can be estimated if one finds an upper bound for $d_{ij}[t]$. Then, the next triggering step t_{h+1}^i could be estimated at the



(a) Scenario of $t^j_{h_j+1} < t^i_{h_i+1}$.



(b) Scenario of $t^j_{h_j+1} \geq t^i_{h_i+1}$.

Fig. 1. Relation between $t^i_{h_i}$, $t^j_{h_j}$, $t^i_{h_i+1}$, $t^j_{h_j+1}$, $t^1_{ij}[t]$, and $t^2_{ij}[t]$.

current triggering step t^i_h. To this end, one will eventually guarantee that $e_i[t]$ decreases exponentially as

$$|e_i[t]| \leq \alpha \mathrm{e}^{-\beta t}, \tag{6}$$

where $\alpha, \beta \in \mathbb{R}_{>0}$. Notice that Eq. (6) is also an essential condition to guarantee secure consensus based on the static and dynamic event-triggering mechanisms.[15,16]

Next, one develops the self-triggering control protocol. To start with, let

$$A = 1 - \frac{\omega}{2}, \quad B = 2(1-\omega)\varepsilon\alpha. \tag{7}$$

The reason for such definition can be found in Eq. (31), where the time-invariant parameters $1-\omega/2$ and $2(1-\omega)$ are denoted as A and B, respectively, for the convenience of subsequent derivation.

At agent i's triggering step $t^i_{h_i}$, agent i can obtain agent j's latest triggering step $t^j_{h_j}$, which is before $t^i_{h_i}$, and its next triggering step $t^j_{h_j+1}$, which is after $t^i_{h_i}$, where $j \in \mathcal{N}_i$. Then, the difference $d_{ij}[t]$ is constant for $t \in [t^i_{h_i}, t^j_{h_j+1}]$. For the convenience of expression, we define

$$t^1_{ij}[t] = \min\left\{t, t^j_{h_j+1}\right\}, \quad t^2_{ij}[t] = \max\left\{t, t^j_{h_j+1}\right\}, \quad t \in \left[t^i_{h_i}, t^i_{h_i+1}\right). \tag{8}$$

To show the relation between these time steps more intuitively, Fig. 1 discusses the two scenarios regarding $t^i_{h_i+1}$ and $t^j_{h_j+1}$. According to the definitions of $t^1_{ij}[t]$ and $d_{ij}[t]$, one knows that $d_{ij}[t]$ is constant for $t \in \left[t^i_{h_i}, t^1_{ij}[t]\right]$. For $t > t^1_{ij}[t]$, the difference $d_{ij}[t]$ has an upper bound given in Eq. (35). By invoking Eq. (5), one designs the following function w.r.t. agent i:

$$g_i[t] = \varepsilon \left| \sum_{j\in\mathcal{N}_i} \theta_{ij}[t](t^1_{ij} - t^i_h) d_{ij}[t^i_h] \right| + \varepsilon \sum_{j\in\mathcal{N}_i} \theta_{ij}[t] \sum_{m=t^j_{h+1}}^{t^2_{ij}-1} \left(2\alpha \mathrm{e}^{-\beta t} + f[m]\right), \tag{9}$$

where

$$f[m] = A^t L_{\max} + \frac{A^t - \mathrm{e}^{-\beta t}}{A - \mathrm{e}^{-\beta}} B. \tag{10}$$

The parameter $L_{\max}$ satisfies $L_{\max} \geq L[0]$, where $L[t]$ is a Lyapunov function candidate, which will be defined in Eq. (26).

It should be noted that the motivation of designing Eq. (9) is to find an upper bound for $e_i[t]$, and the reason of defining Eq. (10) can be found in Eq. (34).

Now, one lets the first triggering step t_0^i for each agent $i \in \mathcal{V}$ be $t_0^i = 0$. Then, agent i determines the subsequent triggering steps $\left\{t_h^i\right\}_{h=1}^{\infty}$ through

$$t_{h+1}^i = \min\left\{t > t_h^i : g_i[t] > \alpha \mathrm{e}^{-\beta t}\right\}. \tag{11}$$

Notice that in Eq. (11), the left term $g_i[t]$ increases w.r.t. $t \in \left[t_h^i, t_{h+1}^i\right)$, the right term $\alpha \mathrm{e}^{-\beta t}$ decreases w.r.t. $t \in \left[t_h^i, t_{h+1}^i\right)$, and $g_i[t_h^i] = 0$. Therefore, given the latest triggering step t_h^i, agent i could estimate the next triggering step t_{h+1}^i by solving Eq. (11).

2.3. *Attack model and secure consensus*

In the context of adversarial attack, agents in the MAS are classified into healthy agents and faulty agents according to the following definitions:

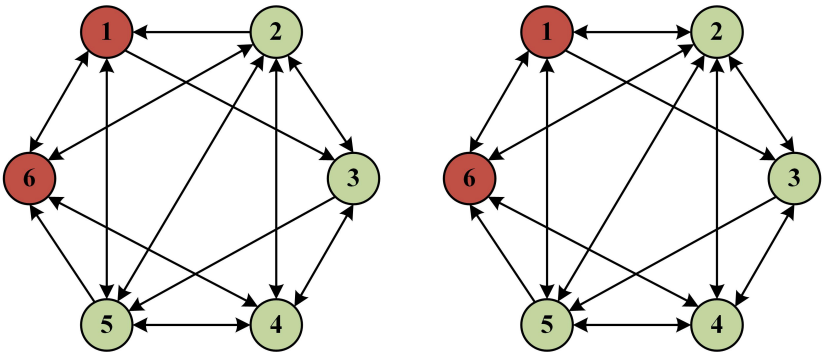
Definition 3 (Healthy agent).[9] An agent is said to be healthy if it sends its state variable $x_i[t]$ to all of its neighbors at each time step t and uses the rule (1) for state update.

Definition 4 (Faulty agent).[9] An agent is said to be faulty if it sends its state variable $x_i[t]$ to all of its neighbors at each time step t, but its state update is uncontrolled by the designed rule (manipulated by attackers).

We denote the sets of healthy and faulty agents as $\mathcal{H}$ and $\mathcal{F}$, respectively. Then, it holds $\mathcal{H} \subseteq \mathcal{V}$ and $\mathcal{F} := \mathcal{V} \backslash \mathcal{H}$. Note that the identities of faulty agents are unknown to healthy agents, while the faulty agents possess the knowledge about the network topology to perpetrate a more targeted adversarial attack (e.g. compromise vulnerable agents). In addition, we consider a limitation on the maximum number of faulty agents in the neighbor set of each healthy agent and introduce the following attack model:

Definition 5 (f-local attack model).[9] The set of faulty agent $\mathcal{F}$ is said to be an f-local attack model if there exist at most f faulty agents in the neighbor set of each healthy agent, i.e., $|\mathcal{N}_i \cap \mathcal{F}| \leq f,\ \forall i \in \mathcal{V} \backslash \mathcal{F}$, where $f \in \mathbb{Z}_{\geq 0}$.

The f-local attack model is usually reflected through the communication topology. Thus, a graphical example of the 1-local attack model is provided in Fig. 2 to understand the notion of attack model more intuitively. Note that the only difference between Fig. 2(a) and Fig. 2(b) is that there exits a directed edge from Agent 1 to Agent 2 in Fig. 2(b). However, Fig. 2(a) satisfies the 1-local attack model, while



(a) A digraph that satisfies the 1-local attack model.

(b) A digraph that does not satisfy the 1-local attack model.

Fig. 2. A graphical example of the 1-local attack model.

Fig. 2(b) does not. This is because in Fig. 2(b), the neighbor set of Agent 2 contains two faulty agents. Through Fig. 2, the notion of attack model is presented more intuitively.

Remark 2. The introduced f-local attack model originates from the pioneering work[9] and has been widely studied in the context of fault-tolerant broadcasting.[21,22] The purpose of using the f-local attack model is to pose an upper bound on the number of compromised nodes in each agent's neighborhood, thereby quantifying the scope of threats. In the subsequent theoretical analysis and numerical examples, the f-local attack model is also a key condition to achieve secure consensus with the proposed method.

To proceed, the definition of secure consensus is presented below.

Definition 6 (Secure consensus). The MAS is said to achieve secure consensus if the following two conditions hold for all initial state variables of agents and any possible faulty set:

- **Security condition:** The state variable of each healthy agent $i \in \mathcal{H}$ fulfills $x_i[t] \in \mathcal{S},\ \forall t \in \mathbb{Z}_{\geq 0}$, where $\mathcal{S} \subset \mathbb{R}$ is a safety interval.
- **Consensus condition:** For each pair of healthy agents $i, j \in \mathcal{H}$, the upper limit of the difference between their state variables is zero, i.e., $\limsup_{t\to\infty} |x_i[t] - x_j[t]| = 0$.

2.4. *Secure consensus algorithm based on self-triggering strategy*

To tackle the secure consensus problem presented in Definition 6, one develops a secure consensus algorithm based on the self-triggering mechanism introduced in Sec. 2.2. The main steps are exhibited in Algorithm 1, and the corresponding flow

chart is illustrated in Fig. 3. Taking the healthy agent $i \in \mathcal{H}$ as an example. At each triggering step t_h^i, agent i firstly needs to filter out the potential malicious states in its neighbor set. This is achieved by the MSR idea. Specifically, agent i deletes the neighbors' broadcast states that are excessively large or excessively small. It has been demonstrated in Ref. 9 that this operation can efficiently eliminate the threat and enable the healthy agents to achieve secure consensus. Subsequently, the retained neighbor set $\mathcal{J}_i[t]$ is obtained, and agent i will use its own state and the states received from $\mathcal{J}_i[t]$ to compute the control input $u_i[t_h^i]$ and estimate the next triggering step t_{h+1}^i. Then, the state $x_i[t_h]$ and the step t_{h+1}^i will be sent to agent i's neighbors. In addition, at the triggering steps $t_{h'}^j \in \left[t_h^i, t_{h+1}^i\right)$ of the retained neighbors $j \in \mathcal{J}_i[t]$, agent i will also update its control input. When the next triggering step t_{h+1}^i arrives, agent i will perform the same operation as above.

Note that Algorithm 1 is an attack-tolerant method, which will be deployed on all agents in the MAS. However, since the faulty agents have been manipulated by attackers, only the healthy agents will execute Algorithm 1 for state update. It should be also noted that this algorithm consists of two essential parts: threat elimination (Step 2: (i)-(iii)) and self-triggering strategy (Step 2: (iv)-(vi)). The former eliminates the potential threats for each healthy agent, while the latter enables each agent to estimate the next triggering step at the current triggering step. Through Algorithm 1, not only the malicious information sent by neighbors can be efficiently eliminated, but also the frequent monitoring to neighbors' broadcast states can be avoided.

3. Secure Consensus Analysis

With Algorithm 1, the graph condition to achieve secure consensus is further derived in this section. To start with, let $\overline{M}[t] = \max_{i\in\mathcal{H}}\{x_i[t]\}$ and $\underline{m}[t] = \min_{i\in\mathcal{H}}\{x_i[t]\}$. Then, the following lemma presents a vital relation w.r.t. these two quantities.

Lemma 1. *Let Assumption 1 hold. Suppose that each healthy agent executes Algorithm 1. For each time step $t \in \mathbb{Z}_{\geq 0}$, it holds*

$$\begin{aligned} &\overline{M}[t+1] \leq \overline{M}[t] + 2\varepsilon\alpha \mathrm{e}^{-\beta t}, \\ &\underline{m}[t+1] \geq \underline{m}[t]. \end{aligned} \tag{15}$$

Proof. Consider a healthy agent $i \in \mathcal{H}$. Its state update adheres to

$$x_i[t+1] = x_i[t] + \varepsilon \sum_{j\in\mathcal{J}_i[t]} \theta_{ij}[t](\hat{x}_j[t] - \hat{x}_i[t]) \tag{16}$$

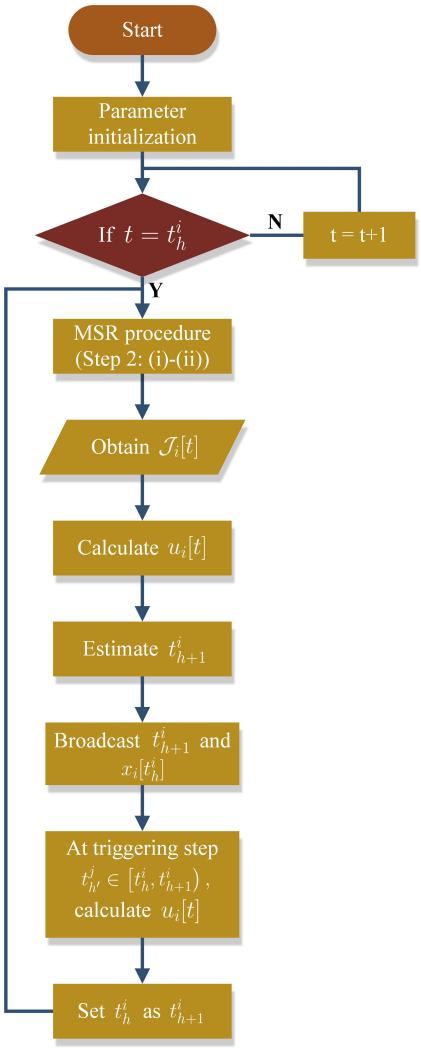


Fig. 3. Flow chart of Algorithm 1.

Since $e_i[t] = \hat{x}_i[t] - x_i[t]$, it yields

$$\begin{aligned} x_i[t+1] &= x_i[t] + \varepsilon \sum_{j\in\mathcal{J}_i[t]} \theta_{ij}[t](e_j[t] + x_j[t] - e_i[t] - x_i[t]) \\ &= \left(1 - \varepsilon \sum_{j\in\mathcal{J}_i[t]} \theta_{ij}[t]\right) x_i[t] + \varepsilon \sum_{j\in\mathcal{J}_i[t]} \theta_{ij}[t] x_j[t] \\ &\quad + \varepsilon \sum_{j\in\mathcal{J}_i[t]} \theta_{ij}[t](e_j[t] - e_i[t]). \end{aligned} \tag{17}$$

Algorithm 1 Secure consensus algorithm based on self-triggering mechanism

1: Initialize $\alpha, \beta \in \mathbb{R}_{>0}$, $h_i = 0$, and $t_0^i = 0$;

2: At triggering step t_h^i, agent i executes the following operations:

(i) Receive $\left\{\hat{x}_j[t_h^i] \mid j \in \mathcal{N}_i\right\}$ and sort them in ascending order.

(ii) If there are fewer than f variables $\hat{x}_j[t_h^i]$ strictly smaller or larger than $x_i[t_h^i]$, then delete all these auxiliary variables; Otherwise, delete the f smallest and largest $\hat{x}_j[t_h^i]$.

(iii) Obtain $\mathcal{J}_i[t]$ as the set of retained neighbors for agent i.

(iv) Compute the control input $u_i[t_h^i]$ according to

$$u_i[t] = \varepsilon \sum_{j \in \mathcal{J}_i[t]} \theta_{ij}[t](\hat{x}_j[t] - \hat{x}_i[t]), \tag{12}$$

(v) Estimate the next triggering step according to

$$t_{h+1}^i = \min\left\{t > t_h^i : g_i[t] > \alpha \mathrm{e}^{-\beta t}\right\}, \tag{13}$$

where

$$g_i[t] = \varepsilon \left| \sum_{j \in \mathcal{J}_i[t]} \theta_{ij}[t](t_{ij}^1 - t_h^i) u_{ij}[t_h^i] \right| + \varepsilon \sum_{j \in \mathcal{J}_i[t]} \theta_{ij}[t] \sum_{m=t_{h+1}^j}^{t_{ij}^2 - 1} \left(2\alpha \mathrm{e}^{-\beta t} + f[m]\right). \tag{14}$$

(vi) Send t_{h+1}^i and $x_i[t_h^i]$ to agent i's neighbors.

3: At triggering step $t_{h'}^j$ of the retained neighbor $j \in \mathcal{J}_i[t]$ with $t_{h'}^j \in \left[t_h^i, t_{h+1}^i\right)$, agent i computes its control input $u_i[t]$ according to Eq. (12).

4: Set h_i to $h_i + 1$ and goes back to Step 2.

From the absolute value inequality, one produces

$$e_j[t] - e_i[t] \leq |e_j[t] - e_i[t]| \leq |e_j[t]| + |e_i[t]|. \tag{18}$$

Synthesizing Eq. (18) with Assumption 1 and Eq. (5), one rewrites Eq. (17) as

$$\begin{aligned} x_i[t+1] &\leq \left(1 - \varepsilon \sum_{j \in \mathcal{J}_i[t]} \theta_{ij}[t]\right) \overline{M}[t] + \varepsilon \sum_{j \in \mathcal{J}_i[t]} \theta_{ij}[t] \overline{M}[t] + 2\varepsilon\alpha \mathrm{e}^{-\beta t} \\ &= \overline{M}[t] + 2\varepsilon\alpha \mathrm{e}^{-\beta t}. \end{aligned} \tag{19}$$

Since Eq. (19) holds for all healthy agents, one concludes

$$\overline{M}[t+1] \leq \overline{M}[t] + 2\varepsilon\alpha \mathrm{e}^{-\beta t}. \tag{20}$$

Regarding $\underline{m}[t+1] \geq \underline{m}[t]$, it follows from the absolute value inequality that

$$e_j[t] - e_i[t] \geq -(|e_j[t]| + |e_i[t]|) \geq 0. \tag{21}$$

Subsequently, one can rewrite Eq. (17) as

$$x_i[t+1] \geq \left(1-\varepsilon \sum_{j\in\mathcal{J}_i[t]}\theta_{ij}[t]\right)\underline{m}[t]+\varepsilon\sum_{j\in\mathcal{J}_i[t]}\theta_{ij}[t]\underline{m}[t]$$
$$=\underline{m}[t]. \tag{22}$$

Since Eq. (22) holds for all healthy agents, one eventually concludes

$$\underline{m}[t+1]\geq \underline{m}[t]. \tag{23}$$

This completes the proof of Lemma 1. □

With Lemma 1, one further provides the condition on network topology to ensure secure consensus with Algorithm 1.

Theorem 1. Consider a multi-agent network modeled by $\mathcal{G}=(\mathcal{V},\mathcal{E})$. Let Assumption 1 hold. Suppose that the faulty set $\mathcal{F}$ satisfies the f-local attack model. Then, secure consensus is guaranteed with Algorithm 1 if $\mathcal{G}$ is $(2f+1)$-robust.

Proof. For $\epsilon\in\mathbb{R}$ and $t\in\mathbb{Z}_{\geq 0}$, let

$$\mathcal{X}_{\overline{M}}(t,\epsilon)=\left\{i\in\mathcal{V}: x_i[t]>\overline{M}[t]-\epsilon\right\},$$
$$\mathcal{X}_{\underline{m}}(t,\epsilon)=\left\{i\in\mathcal{V}: x_i[t]<\underline{m}[t]+\epsilon\right\}. \tag{24}$$

Moreover, one defines

$$\mathcal{Y}_{\overline{M}}(t,\epsilon)=\mathcal{X}_{\overline{M}}(t,\epsilon)\cap\mathcal{H},$$
$$\mathcal{Y}_{\underline{m}}(t,\epsilon)=\mathcal{X}_{\underline{m}}(t,\epsilon)\cap\mathcal{H}, \tag{25}$$

where $\mathcal{Y}_{\overline{M}}(t,\epsilon)$ and $\mathcal{Y}_{\underline{m}}(t,\epsilon)$ contain the healthy agents in $\mathcal{X}_{\overline{M}}(t,\epsilon)$ and $\mathcal{X}_{\underline{m}}(t,\epsilon)$, respectively.

Subsequently, one constructs the Lyapunov function candidate $L[t]$ as

$$L[t]=\overline{M}[t]-\underline{m}[t]. \tag{26}$$

Then, one focuses on two nonempty and disjoint sets $\mathcal{Y}_{\overline{M}}(t,\epsilon_0)$ and $\mathcal{Y}_{\underline{m}}(t,\epsilon_0)$, where $\epsilon_0=L[t]/2$. Since $\mathcal{G}$ is $(2f+1)$-robust, there exists an agent i in either $\mathcal{Y}_{\overline{M}}(t,\epsilon_0)$ or $\mathcal{Y}_{\underline{m}}(t,\epsilon_0)$ that has at least $2f+1$ neighbors outside its respective set. Suppose $i\in\mathcal{Y}_{\overline{M}}(t,\epsilon_0)$. Since the faulty set $\mathcal{F}$ satisfies the f-local attack model and each healthy agent executes Algorithm 1, at least one of agent i's $2f+1$ neighbors will be retained and its state will be used for agent i's update, i.e., $x_j[t]\leq\overline{M}[t]-\epsilon_0,\ \ \exists j\in\mathcal{J}_i[t]$. From Lemma 1, one obtains that the largest state that agent i will use for update at time step t is $\overline{M}[t]+2\varepsilon\alpha\mathrm{e}^{-\beta t}$. By placing the largest possible weight $1-\omega$ on $\overline{M}[t]+2\varepsilon\alpha\mathrm{e}^{-\beta t}$ and placing the smallest possible weight ω on $\overline{M}[t]-\epsilon_0$, one produces

$$x_i[t+1]\leq(1-\omega)(\overline{M}[t]+2\varepsilon\alpha\mathrm{e}^{-\beta t})+\omega(\overline{M}[t]-\epsilon_0)$$
$$=M[t]-\omega\epsilon_0+2(1-\omega)\varepsilon\alpha\mathrm{e}^{-\beta t}. \tag{27}$$

Note that Eq. (27) also holds for healthy agents in , since agent i will use its own state for update. This fact means

$$M[t+1] \leq M[t] - \omega\epsilon_0 + 2(1-\omega)\varepsilon\alpha \mathrm{e}^{-\beta t}. \tag{28}$$

On the other hand, it follows from Lemma 1 that

$$m[t+1] \geq m[t]. \tag{29}$$

Synthesizing Eq. (28) with Eq. (29) yields

$$M[t+1] - m[t+1] \leq M[t] - m[t] - \omega\epsilon_0 + 2(1-\omega)\varepsilon\alpha \mathrm{e}^{-\beta t}, \tag{30}$$

By invoking Eq. (26) and the definition of ϵ_0, Eq. (30) can be reorganized as

$$\begin{aligned} L[t+1] &\leq L[t] - \omega\epsilon_0 + 2(1-\omega)\varepsilon\alpha \mathrm{e}^{-\beta t} \\ &= L[t] - \omega\frac{L[t]}{2} + 2(1-\omega)\varepsilon\alpha \mathrm{e}^{-\beta t} \\ &= (1-\frac{\omega}{2})L[t] + 2(1-\omega)\varepsilon\alpha \mathrm{e}^{-\beta t} \\ &= AL[t] + B\mathrm{e}^{-\beta t}. \end{aligned} \tag{31}$$

By iteration, one further derives

$$L[t+n] \leq A^n L[t] + B\sum_{l=0}^{n-1} A^l \mathrm{e}^{-\beta(t+n-1-l)}. \tag{32}$$

Let $t=0$ and $n=t$. Then, Eq. (32) can be rewritten as

$$\begin{aligned} L[t] &\leq A^t L[0] + B\sum_{l=0}^{t-1} A^l \mathrm{e}^{-\beta(t-1-l)} \\ &\leq A^t L_{\max} + \frac{A^t - \mathrm{e}^{-\beta t}}{A - \mathrm{e}^{-\beta}} B. \end{aligned} \tag{33}$$

To proceed, one seeks an upper bound for $|x_i[t] - x_j[t]|$, i.e.,

$$\begin{aligned} |x_i[t] - x_j[t]| &\leq M[t] - m[t] = L[t] \\ &\leq A^t L_{\max} + \frac{A^t - \mathrm{e}^{-\beta t}}{A - \mathrm{e}^{-\beta}} B \\ &= f[t]. \end{aligned} \tag{34}$$

Then, the upper bound for $d_{ij}[t]$ is

$$\begin{aligned} |d_{ij}[t]| &= |\hat{x}_i[t] - \hat{x}_j[t]| \\ &\leq |\hat{x}_i[t] - x_i[t]| + |\hat{x}_j[t] - x_j[t]| + |x_i[t] - x_j[t]| \\ &\leq 2\alpha \mathrm{e}^{-\beta t} + f[t] \end{aligned} \tag{35}$$

Next, one needs to seek an upper bound for $e_i[t]$. To this end, one focuses on triggering step t_h^i, at which agent i has known t_h^j, t_{h-1}^j, and $x_j[t_h^j]$ w.r.t. its neighbors $j \in \mathcal{N}_i$. Note that $d_{ij}[t]$ is constant for $t \in [t_h^i, t_{ij}^1)$, where t_{ij}^1 has been defined in Eq. (8). For $t > t_{ij}^1$, the difference $d_{ij}[t]$ is upper bounded by $2\alpha \mathrm{e}^{-\beta t} + f[t]$ from Eq. (35). Consequently, one produces

$$e_i[t] = \left| (t - t_h^i)\varepsilon \sum_{j \in \mathcal{J}_i[t]} \theta_{ij}[t] d_{ij}[t] \right| \leq g_i[t], \quad t \in [t_h^i, t_{h+1}^i), \tag{36}$$

which indicates that $|e_i[t]|$ is bounded by $g_i[t]$. Thus, with the self-triggering condition (11), one eventually obtains

$$|e_i[t]| \leq \alpha \mathrm{e}^{-\beta t}, \tag{37}$$

which holds for all $t \in [t_h^i, t_{h+1}^i)$. This fact means that all state errors will decrease exponentially. Moreover, one produces $\lim_{t \to \infty} L[t] = 0$ from Eq. (31). Hence, secure consensus is guaranteed with the proposed self-triggering strategy. This completes the proof of Theorem 1. □

Remark 3. From the viewpoint of practical applications, Theorem 1 provides an attack-tolerant approach to defend against potential adversarial attacks. For example, Santilli *et al.*[23] proposed an attack-tolerant strategy to achieve static secure containment for multi-robot systems. Regarding the practical multi-microgrid systems, Yassaie *et al.*[24] addressed the secure consensus problem when some microgrids are subject to false data injection and replay attacks, and Liao *et al.*[12] addressed an economic dispatch problem (EDP) under adversarial environments. The design of secure algorithms in both studies is based on the attack-tolerant idea. In addition, Shahabadi *et al.*[25] demonstrated that the introduction of self-triggering mechanism can efficiently eliminate the need for continuous monitoring of the triggering condition for islanded microgrids. Thus, extending the proposed self-triggering secure-consensus algorithm to practical systems will be also one of our future research work.

4. Numerical Example

In this section, one provides three numerical examples to exhibit the effectiveness, scalability, and superior performance of Algorithm 1. To start with, the effectiveness of the proposed method is validated through a comparative case study with an existing self-triggering strategy. Subsequently, the obtained results are extended to a larger network topology, where two faulty agents aim to compromise the healthy agents in the network. Finally, another comparative case study is conducted to demonstrate the superior performance of the proposed self-triggering strategy compared with the existing static and dynamic event-triggering strategies.[15,16]

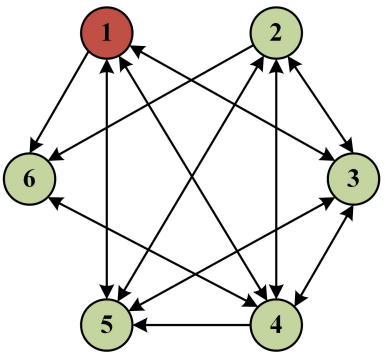


Fig. 4. A 3-robust digraph with six nodes.

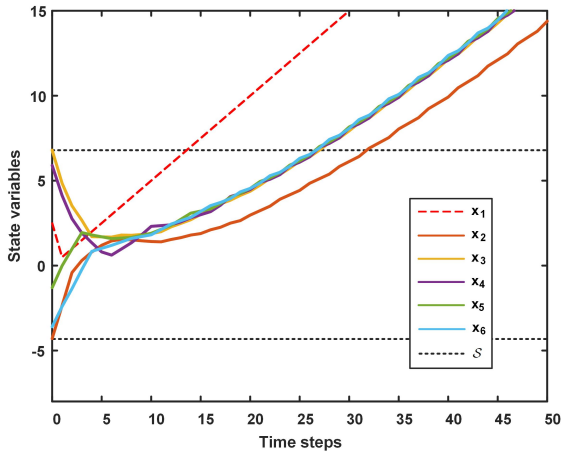
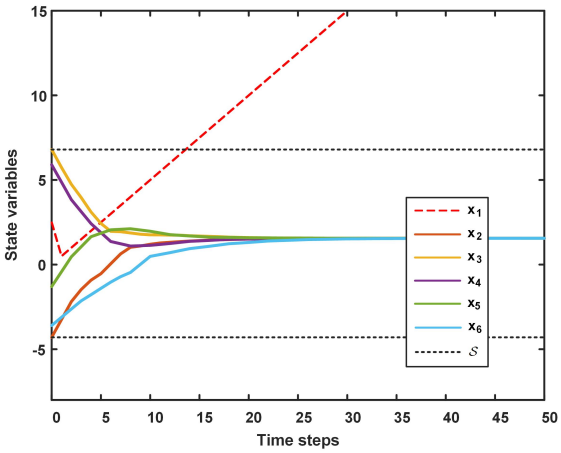


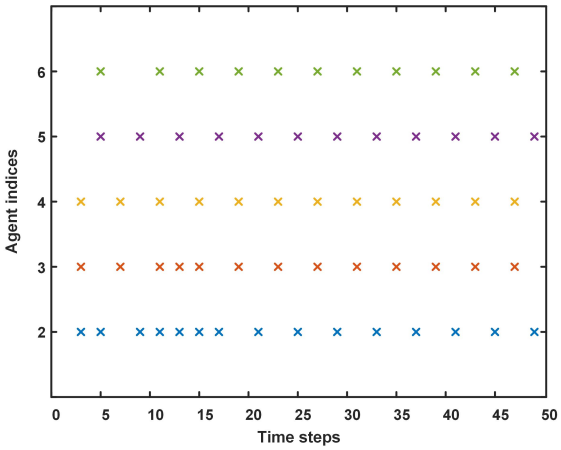
Fig. 5. Trajectories of agents using the algorithm in Ref. 18. The healthy agents fail to achieve secure consensus.

4.1. *Effectiveness validation*

Consider a multi-agent network modeled by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; see Fig. 4. Note that $\mathcal{G}$ is 3-robust, which satisfies the sufficient condition for achieving secure consensus with Algorithm 1 when the MAS is subject to the 1-local attack model. Let the initial states of all agents be $[x_1[0], \cdots, x_6[0]]^{\mathrm{T}} = [2.5, -4.3, 6.8, 5.9, -1.3, -3.6]^{\mathrm{T}}$. The safety interval is obtained as $= [\min_{i\in\mathcal{H}} x_i[0], \max_{i\in\mathcal{H}} x_i[0]] = [-4.3, 6.8]$. In addition, one assumes that Agent 1 is attacked and becomes a faulty agent at the first time step, whose motion adheres to $x_1[t] = 0.5 \times t$. It can be verified that the MAS satisfies the 1-local attack model. According to Theorem 1, secure consensus can be guaranteed with Algorithm 1. Regarding the self-triggering mechanism, one lets $\alpha = 8, \beta = 0.1$. Moreover, the control gain is set as $\varepsilon = 0.4$.



(a) The healthy agents achieve secure consensus.



(b) Triggering steps of healthy agents.

Fig. 6. Trajectories and triggering behaviors of agents using the proposed secure consensus algorithm.

Firstly, Fig. 5 illustrates the evolution of the state variables of agents using the consensus algorithm in Ref. 18, which does not consider the security factor, but pursues average consensus. One observes that all healthy agents are severely affected by Agent 1 and their state variables exceed the safety interval, thus the consensus cannot be guaranteed in this case.

Subsequently, one applies Algorithm 1 and obtains the numerical result in Fig. 6. From Fig. 6(a), it is observed that secure consensus is guaranteed, since the state variables of healthy agents converge to the same consensus value within the safety interval, regardless of the misbehavior of Agent 1. Furthermore, the introduction of self-triggering mechanism reduces the communication times between agents, as

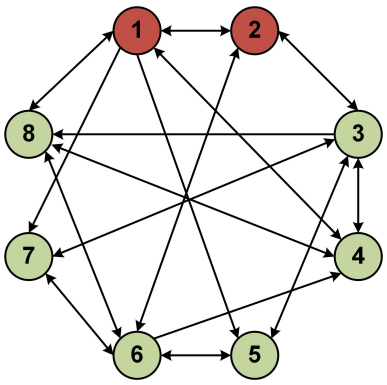


Fig. 7. A 3-robust digraph with eight nodes.

illustrated in Fig. 6(b). The communication overhead is thereby mitigated. Overall, the effectiveness of the proposed method is validated.

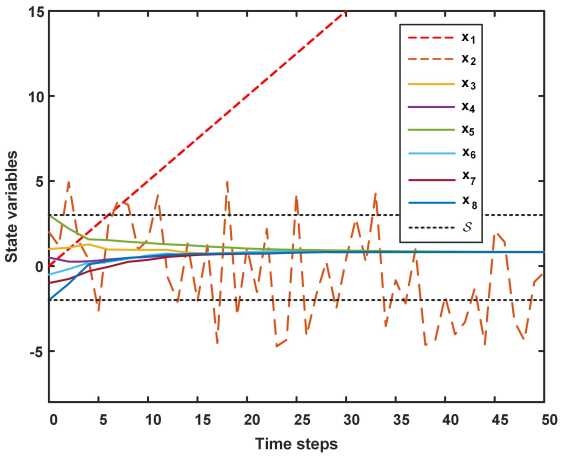
4.2. *Scalability validation*

Consider a larger multi-agent network modeled by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; see Fig. 7. Note that $\mathcal{G}$ is still 3-robust, which satisfies the sufficient condition for achieving secure consensus with Algorithm 1 when the MAS is subject to the 1-local attack model. Let the initial states of all agents be $[x_1[0], \cdots, x_8[0]]^{\mathrm{T}} = [0, 2, 1, 0.5, 3, -0.5, -1, -2]^{\mathrm{T}}$. The safety interval is obtained as $= [\min_{i\in\mathcal{H}} x_i[0], \max_{i\in\mathcal{H}} x_i[0]] = [-2, 3]$. In addition, one assumes that Agents 1 and 2 are attacked and become faulty agents at the first time step, whose motions adhere to $x_1[t] = 0.5 \times t$ and $x_2[t] = rand(-5, 5)$, where $rand(-5, 5)$ refers to a random value in the interval $(-5, 5)$. It can be verified that the MAS still satisfies the 1-local attack model, since the neighbor set of each agent contains at most one faulty agent. According to Theorem 1, secure consensus can be guaranteed with Algorithm 1. Other settings are the same as that in Section 4.1.

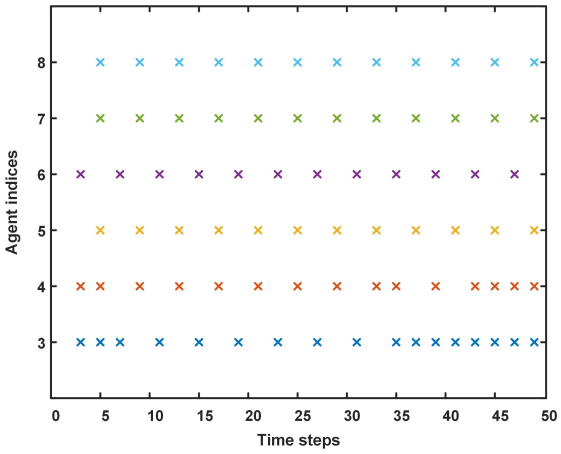
Now one applies Algorithm 1 and obtains the numerical result in Fig. 8. From Fig. 8(a), it is observed that secure consensus is achieved, since the state variables of healthy agents converge to the same consensus value within the safety interval, regardless of the misbehaviors of Agents 1 and 2. Furthermore, the introduction of self-triggering mechanism reduces the communication times between agents, as illustrated in Fig. 8(b). Overall, the scalability of the proposed method is thereby validated.

4.3. *Comparison with the existing event-based algorithms*

Next, we compare the proposed self-triggering secure consensus algorithm with the existing secure algorithms based on static and dynamic event-triggering mechanisms.[15,16] Note that all of them can ensure secure consensus, while we mainly



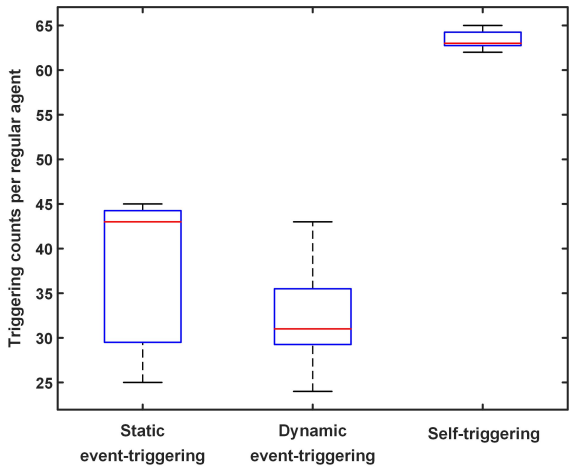
(a) The healthy agents achieve secure consensus.



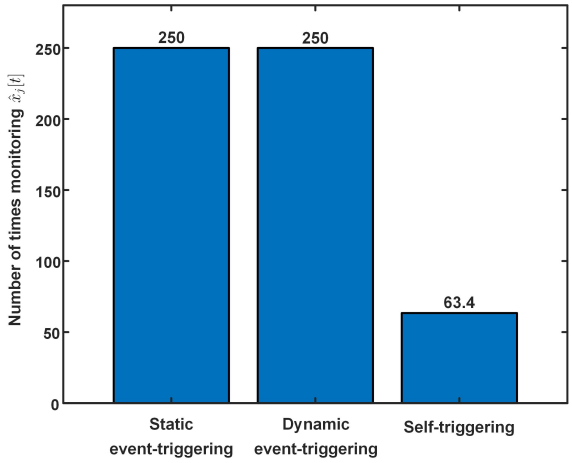
(b) Triggering steps of healthy agents.

Fig. 8. Trajectories and triggering behaviors of agents using the proposed secure consensus algorithm.

focus on comparing their triggering behaviors. Fig. 9(a) shows the triggering counts of healthy agents using three different triggering strategies. The average triggering counts for the static, dynamic, and self-triggering strategies are 37.6, 32.4, and 63.4, respectively. This result indicates that the communication burden using the self-triggering strategy is slightly heavier than using the static or dynamic event-triggering strategy. However, from Fig. 9(b), one observes that the static or dynamic event-triggering strategy needs to monitor agent j in order to receive $\hat{x}_j[t]$ at each time step, while the self-triggering strategy only needs to monitor $\hat{x}_j[t]$ at triggering steps. Overall, the self-triggering strategy is beneficial for reducing the consumption



(a) Triggering counts between three event-based strategies.



(b) Monitoring times between three event-based strategies.

Fig. 9. Comparison of trigger behaviors using three event-based strategies within 250 time steps.

of communication resources.

5. Conclusion

In this work, an attack-tolerant algorithm is developed to tackle the secure consensus problem for MASs. The idea of MSR and self-triggering mechanism are applied in the algorithm design. The numerical results demonstrate that the MAS achieves consensus within the safety interval by implementing the proposed secure consensus algorithm, and the heavy communication burden is mitigated. Furthermore, the frequent monitoring of $\hat{x}_j[t]$ is avoided with the introduction of self-triggering

strategy. In the future, we will investigate the secure consensus problem for more complex situations and systems, e.g., multi-dimensional space or nonlinear MASs. Moreover, we will consider more application-oriented secure coordination tasks, e.g., secure state estimation and secure distributed optimization.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant Nos. 62173147, 62303030, 62403028, U2233212), Beijing Municipal Natural Science Foundation (Grant No. L221008), Open Fund of Science and Technology on Thermal Energy and Power Laboratory (Grant No. TPL2022C02).

References

1. R. Olfati-Saber, J. A. Fax and R. M. Murray, *Proc. IEEE* **95**, 215 (2007).
2. Q. Zhou, G. Feng and X. Xu, *Guid. Navig. Control* **2**, p. 2250019 (2022).
3. Z. Sun, H. G. de Marina, G. S. Seyboth, B. D. Anderson and C. Yu, *IEEE Trans. Control Syst. Technol.* **27**, 192 (2018).
4. M. Huo, H. Duan and Y. Fan, *Guid. Navig. Control* **1**, p. 2150004 (2021).
5. L. Li, P. Shi and C. K. Ahn, *IEEE T. Cybern.* **52**, 4647 (2020).
6. A. Teixeira, I. Shames, H. Sandberg and K. H. Johansson, *Automatica* **51**, 135 (2015).
7. Y. Yang, Y. Xiao and T. Li, *IEEE T. Cybern.* **52**, 12805 (2021).
8. Z. Liao, J. Shi, Y. Zhang, S. Wang and Z. Sun, *arXiv preprint arXiv:2402.10505* (2024).
9. H. J. LeBlanc, H. Zhang, X. Koutsoukos and S. Sundaram, *IEEE J. Sel. Areas Commun.* **31**, 766 (2013).
10. Y. Yang and W. Sun, *Automatica* **169**, p. 111834 (2024).
11. G. Wen, Y. Lv, W. X. Zheng, J. Zhou and J. Fu, *IEEE Trans. Autom. Control* **68**, 6466 (2023).
12. Z. Liao, S. Wang, J. Shi, M. Li, Y. Zhang and Z. Sun, *ISA Trans.* **149**, 1 (2024).
13. S. Koushkbaghi, M. Safi, A. M. Amani, M. Jalili and X. Yu, *IEEE Transactions on Cybernetics* (2024).
14. X. Gong, Y. Chen, F. Zou, W. Liu, J. Shen and Z. Shu, *IEEE Transactions on Cybernetics* (2024).
15. Y. Wang and H. Ishii, *IEEE Trans. Control Netw. Syst.* **7**, 471 (2020).
16. Z. Liao, J. Shi, S. Wang, Y. Zhang and Z. Sun, *IEEE Trans. Circuits Syst. II-Express Briefs* **71**, 3463 (2024).
17. X. Yi, K. Liu, D. V. Dimarogonas and K. H. Johansson, *IEEE Trans. Autom. Control* **64**, 3300 (2018).
18. R. K. Mishra and H. Ishii, *Int. J. Robust Nonlinear Control* **33**, 159 (2023).
19. H. Matsume, Y. Wang and H. Ishii, *Nonlinear Anal.-Hybrid Syst.* **42**, p. 101091 (2021).
20. H. Matsume, Y. Wang, H. Ishii and X. Défago, *Nonlinear Anal.-Hybrid Syst.* **52**, p. 101473 (2024).
21. A. Ichimura and M. Shigeno, *Inf. Process. Lett.* **110**, 514 (2010).
22. H. Wei, H. Zhang, A.-H. Kamal and Y. Shi, *Engineering* **33**, 35 (2024).
23. M. Santilli, M. Franceschelli and A. Gasparri, *Automatica* **143**, p. 110456 (2022).
24. N. Yassaie, M. Hallajiyan, I. Sharifi and H. Talebi, *ISA Trans.* **110**, 238 (2021).
25. M. Z. Shahabadi, H. Atrianfar, G. B. Gharehpetian and A. Yazdani, *IEEE Transactions on Industrial Informatics* (2024).

Photo and Bibliography

Zirui Liao received the B.E. degree in mechanical engineering from China Agricultural University, Beijing, China, in 2020.

He is currently pursuing the Ph.D. degree in mechanical engineering with Beihang University, Beijing, China. His research interests include cyber-physical system, resilient control, distributed optimization.

Jian Shi received the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China, in 2007.

He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University. His major research interests include fault diagnosis, health management, and network reliability.

Shaoping Wang received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Beihang University, Beijing, China, in 1988, 1991, and 1994, respectively.

She has been with the School of Automation Science and Electrical Engineering, Beihang University since 1994 and promoted to the rank of Professor in 2000. She was honored as a Changjiang Scholar Professor by the Ministry of Education of China in 2013. Her research interests include engineering reliability, fault diagnostic, prognostic and health management, and fault tolerant control.

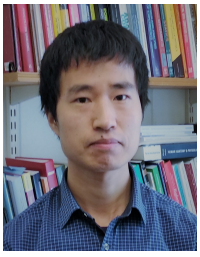
Prof. Wang is the corresponding author of this paper.

Yuwei Zhang received the B.S. degree in mathematics and applied mathematics and the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China, in 2016 and 2021, respectively.

He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University. His research interests include nonlinear control and cooperative control of multi-robot system.

Rui Mu received the B.S.and M.S. degrees in applied mathematics from Shandong Normal University, Jinan, China, in 2015 and 2018, respectively, and the Ph.D.degree in control theory and control engineering from Shandong University,Jinan, in 2023.

She is currently a Postdoctoral Researcher with the School of Automation Science and Electrical Engineering, Beihang University. Her research interests include cooperative control,event-triggered control,and multiagent systems.



Zhiyong Sun received the Ph.D. degree in control engineering from The Australian National University (ANU), Canberra, ACT, Australia, in February 2017.

He was a Research Fellow with ANU, and then a postdoctoral researcher at the Department of Automatic Control, Lund University of Sweden. Since January 2020, he has joined Eindhoven University of Technology (TU/e) as an assistant professor. He joins Peking University of China as a faculty member in the summer of 2024. His research interests include multi-agent systems, control of autonomous formations, distributed control and optimization.