electronics

Article

Resilient Consensus for Multi-Agent Systems in the Presence of Sybil Attacks

Xiaochen Dong, Yiming Wu, Ming Xu * and Ning Zheng

School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China; dongxiaochen@hdu.edu.cn (X.D.); ymwu@hdu.edu.cn (Y.W.); nzheng@hdu.edu.cn (N.Z.)
* Correspondence: mxu@hdu.edu.cn

Abstract: This paper investigates the problem of resilient consensus control for discrete-time linear multi-agent systems under Sybil attacks. We consider a node to be a Sybil node if it can generate a large number of false identities in the graph as a way of gaining disproportionate influence on the consensus performance of the network. Such attacks can easily invalidate existing resilient consensus algorithms that assume an upper bound on the number of malicious nodes in the network. To this end, we first built a new attack model based on the characteristics of the Sybil nodes. In addition, a quantized-data-based transmission scheme was developed for identifying and resisting Sybil nodes in the network. Then, an attack-resilient consensus algorithm was developed, where each normal node sends the quantitative data information with a specific label, which is generated by truncated normal distribution sampling to their neighbors. We give sufficient graphical conditions for attack models considering limited energy to ensure the consensus of linear multi-agent systems. Finally, numerical simulation examples are provided to validate the effectiveness of the proposed methods.

Keywords: multi-agent systems; Sybil attack; resilient consensus; attack-tolerant control

Citation: Dong, X.; Wu, Y.; Xu, M.; Zheng, N. Resilient Consensus for Multi-Agent Systems in the Presence of Sybil Attacks. *Electronics* **2022**, *11*, 800. https://doi.org/10.3390/electronics11050800

Academic Editor: Jose Eugenio Naranjo

Received: 19 January 2022
Accepted: 2 March 2022
Published: 4 March 2022

1. Introduction

In the last decade, cooperative control of multi-agent systems (MASs) has attracted significant attention from researchers and can be found in various fields [1–3], for instance, distributed computing, sensor networks, autonomous vehicles, and the Internet of Things. As one of the fundamental problems of MASs, consensus control is the combination of graph theory and control systems, which aims to make these distributed and locally cooperative agents reach an agreement in terms of some interests. However, as MASs are often in an open environment, some agents may become non-cooperative or crash when cyber attacks or network failures occur in the system, which will lead to the failure of consensus. Therefore, resilient consensus, the consensus in the face of some agents in the network subject to faults or attacks, has been investigated by many researchers in recent years.

Recently, many works have been dedicated to resilient consensus for MASs with different network situations or types of cyber-attacks [4–7]. Roughly speaking, the goal of resilient consensus is to prevent the malicious agents from influencing the system's consensus process by appropriate consensus strategy and designing sufficient redundancy in the underlying network [8–10]. For the case of a time-varying network, the authors in [11] provided sufficient and necessary conditions for the design of resilient consensus protocols. Furthermore, the author in [12] studied the resilient consensus problem for switched MASs. On the other hand, based on the impact of different cyber-attack characteristics on MASs, constructing corresponding control system attack models is also a research hotspot [13–15]. In MASs, the most commonly used attack strategy is data falsification (Byzantine) attack, where the attackers, in an adversarial manner, send inconsistent information to their neighboring nodes [16–19]. In [5], DoS attack models in multi-agent networks are discussed, and a resilient consensus control law is given based on the static output feedback mechanism.

Utilizing the information of the neighbors, a distributed event-triggered algorithm for resilient consensus is provided in [20] by considering the impact of deception attacks on the MASs. Moreover, crashed attack models are also one of the research fields that have recently been considered in the resilient consensus problem [21–23]. However, nearly all of the existing work designed the consensus protocol on the assumption that there is an upper bound on the number of malicious nodes in the multi-agent network.

A particularly challenging attack on this assumption is the so-called Sybil attack [24], in which a malicious agent can create multiple fake or captured identities to gain a disproportionate influence on the distributed networks. The Sybil attack was first studied by Douceur in the context of peer-to-peer networks [25]. Such an attack has a devastating effect on routing distributed networks such as voting, resource storage, and social networks. To date, there have been some works on MASs under Sybil attacks. Dong and Liu [26] considered the Sybil attack in sensor networks and proposed a robust and secure time-synchronization protocol with a graph-theoretical approach. Jamshidi et al. [27] recently introduced a precise and straightforward algorithm for detecting Sybil attacks in WSNs with the observer monitoring behavior of nodes. Huang et al. [28] introduced a so-called "ScatterID" system which attaches backscatter tags to single-antenna robots to defend against Sybil attacks in multi-robot networks. Wheeler et al. [29] proposed a secure consensus algorithm against Sybil attack by controlling the network communication topology through Wi-Fi signals. A proof-of-concept protocol called ReCon is presented in [30], where the nodes in the network achieve Sybil-resistant consensus by establishing blockchain technology.

It is important to note that the majority of the existing defensive methods against Sybil attack benefit from specific mechanisms, such as central trust authority, and reliable methods to distinguish the physical fingerprints of signals from neighboring nodes [14,31–33]. However, in practice, the trust central authority mechanism is not suitable for the large-scale distributed MAS network. In addition, fingerprint recognition and reputation mechanisms mean extra network components and computing expenditure, which may not be ideal for the MASs, because each agent is resource-limited. Thus, the lightweight attack-tolerant consensus algorithm becomes promising to effectively resist such scenarios that are challenging to be handled otherwise. To the best of our knowledge, the integration of an identity authentication mechanism with an agent update and lightweight Sybil defense algorithm is still a gap in the research of MASs.

Inspired by the above facts, the purpose of our paper is to investigate the resilient consensus problem for MASs under Sybil attacks, where the malicious agents attempt to multicast an excessive amount of state information with different fictitious identities to cause the normal agent state value to be unable to reach a consensus. To describe the characteristics induced by the Sybil attacks in the multi-agent networks, a novel attack model focus on the internal relationships of nodes is introduced for the first time in this paper. Based on the attack model, necessary network topology conditions are obtained for the MASs under a directed graph. Inspired by [34,35], we propose a random label generation and verification mechanism that uses the truncated normal distribution. Such a mechanism can help us to identify the Sybil nodes in the distributed MASs. Then, as a natural extension to our previous work [36], and inspired by the work of Dibaji et al. [37], here, we employ a quantized version of the mean sub-sequence reduced (MSR) algorithm called the QW-MSR algorithm to reach a consensus on the state value of all nodes, which can also effectively reduce the calculation and communication burden of the system. Specifically, we set the state value of each agent to be composed of the integer part generated by the quantization process and the fractional part generated by the label sampling mechanism. It is worth noting that the values generated by the sampling mechanism are time-sensitive and independent and will not affect the integer state value. Some numerical examples are given to demonstrate the effectiveness of our methods.

So, focused on the characteristics of the Sybil attack and compared with the existing literature, which utilizes detection and verification mechanisms against the Sybil attack with extra components, our contributions with this paper can be summarized as follows:

- Compared with [4,9,10], we give a novel network attack model to describe the behavioral characteristics of Sybil attack in MASs.
- Compared with [28–30], we propose a random label generation and verification mechanism incorporating the node information transmission process to detect and mitigate Sybil attacks in the network.
- Compared with [26,27], we propose a novel quantization mechanism during the computation and update of node state information, which makes the consensus algorithm more lightweight and consumes less energy.

The paper is organized as follows: In Section 2, we present some notions in graph theory and propose the Sybil attack model. The node label sampling and verification mechanism and the consensus algorithm are proposed in Section 3. The main results are given in Section 4. Simulation tests are performed to show the effectiveness of our results in Section 5, and the conclusions are presented in Section 6.

2. Preliminaries

2.1. Graph Theory

We model the MAS with a directed graph which is defined as a triple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, where $\mathcal{V}$ donates the node set, $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ represents the directed link set and $\mathcal{A} \in \mathbb{R}^{n \times n}$ represents the adjacency matrix. In this paper, the node set is composed of the normal node set $\mathcal{V}_n$ and Sybil node set $\mathcal{V}_S$, i.e., $\mathcal{V} = \mathcal{V}_n \cup \mathcal{V}_S$, $\mathcal{V}_n \cap \mathcal{V}_S = \emptyset$. The directed edge (j, i) is called the incoming edge of i, which means node i can receive information from node j. For node i, the set of its neighbors is denoted by $\mathcal{N}_i = \{j : (j, i) \in \mathcal{E}\}$, and the number of neighbors is noted by $|\mathcal{N}_i|$. The element a_{ij} in $\mathcal{A}$ is defined by $a_{ij} \in [\mu, 1)$ if $(i, j) \in \mathcal{E}$ where $\mu > 0$, otherwise $a_{ij} = 0$. Next, several concepts of the so-called robust graph are introduced below. Further details and examples can be found in [4].

Definition 1. *r-reachable set: A nonempty set $\mathcal{S} \subseteq \mathcal{V}$ is said to be r-reachable if there exists at least one node $i \in \mathcal{S}$ that $|\mathcal{E}_i| \geq r, r \in \mathbb{Z}^+$, where $\mathcal{E}_i$ donates the set of incoming edges that are outside the set $\mathcal{S}$.*

Definition 2. *(r, s)-reachable set: A nonempty set $\mathcal{S} \subseteq \mathcal{V}$ is said to be (r, s)-reachable if there are at least s nodes in node set S, each of which has at least r neighbors outside of S, where $r, s \in \mathbb{Z}^+$.*

Definition 3. *r-robustness: A directed graph is said to be r-robust if for every pair of nonempty disjoint subsets in $\mathcal{V}$, at least one of the subsets is r-reachable.*

Definition 4. *(r, s)-robustness: We say graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is (r, s)-robust if for each pair of nonempty disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of the following conditions is met (define $\mathcal{X}^r_{\mathcal{S}_k} = \{i \in \mathcal{S}_k : |\mathcal{V}_i \backslash \mathcal{S}_k| \geq r\}, k \in \{1, 2\}$):*

1. $|\mathcal{X}^r_{\mathcal{S}_1}| = |\mathcal{S}_1|$;
2. $|\mathcal{X}^r_{\mathcal{S}_2}| = |\mathcal{S}_2|$;
3. $|\mathcal{X}^r_{\mathcal{S}_1}| + |\mathcal{X}^r_{\mathcal{S}_2}| \geq s$.

2.2. Sybil Attack Model

As discussed earlier in this paper, we focus on the case of having a small number of Sybil nodes in a MAS network. Each Sybil node can generate multiple fabricated (or fake) identities to gain a disproportionate advantage to control or affect a large number of legitimate nodes in the network. It is worth noting that although the network with Sybil nodes seems to add multiple nodes (fake identities), the number of physical nodes in the network has not increased. Multiple fabricated identities, which are called child

nodes, generated by the first Sybil node, which is called parent node, simultaneously send malicious messages to their neighbors.

Next, we define two types of Sybil nodes in the MAS for Sybil attackers.

Definition 5. *(Sybil parent node and Sybil child node) Consider a MAS under Sybil attacks, where each agent is regarded as a node in a directed graph, denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. The first compromised or captured node is called the Sybil parent node, and the fabricated identities generated by the Sybil parent node are called the Sybil child nodes.*

We denote by $\mathcal{V}_{Sp}$ the set of Sybil parent nodes, and by $\mathcal{V}_{Sc}$ the set of Sybil child nodes. The whole Sybil node-set is composed of the Sybil parent nodes and the Sybil child nodes, which we denote by $\mathcal{V}_S = \mathcal{V}_{Sp} \cup \mathcal{V}_{Sc}$.

Remark 1. *In the previous research on Sybil attack or spoof attack [13,14,27,32], researchers usually identify these Sybil nodes by their attack behavior and characteristics. For the nature of Sybil attack-forged identities, we believe that the internal relationship and generation order of Sybil nodes can be used for Sybil node detection and exclusion. Thus, we divide Sybil nodes into parent nodes reprogrammed with physical entities and replica-forged identities as child nodes.*

In this paper, we assume the Sybil attack dimensions are direct, fabricated, and simultaneous, which means that Sybil nodes as well as fabricated identities can directly connect to their neighboring nodes and send malicious information simultaneously. An example of the type of Sybil nodes discussed in the above definition is shown in Figure 1. Here, we consider a group of six agents with the directed communication topology. The attacker compromised agent 1 and turned it into a Sybil parent node, as shown in Figure 1a. Then, the Sybil parent node generated two Sybil child nodes, i.e., node 1a and node 1b. The entire communication network under Sybil attacks is shown in Figure 1b.

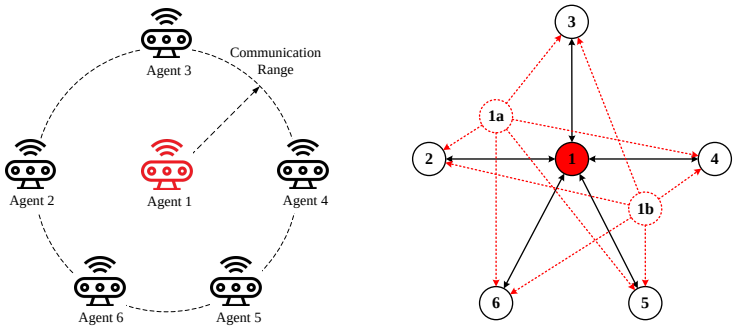


Figure 1. An example of the network under Sybil attacks. (**a**) Schematic diagram of network nodes' locations; (**b**) Network communication topology diagram.

Considering the limited ability and spending, the attackers often cannot compromise all nodes in the network. A standard assumption model in the area of resilient consensus problems is the so-called *F*-local attack model or *F*-total attack model. In such a model, the scope of the malicious nodes is usually assumed to be bounded by a constant *F* in the neighborhood of each node (*F*-local attack model) or the total area of all nodes (*F*-total attack model) [4,12,15]. The above assumptions are invalid for Sybil attacks that can arbitrarily generate fabricated nodes. Therefore, it is necessary to redefine the scope model of Sybil attacks according to the different types of Sybil nodes. In this paper, we have the following definitions.

Definition 6. *(F-parent local attack model): A network is considered to be under F-parent local Sybil attack if there are mostly F Sybil parent nodes in the neighborhood of each normal node.*

Definition 7. *(F-parent total attack model): A network is considered to be under F-parent total Sybil attack if there are mostly F Sybil parent nodes in the network.*

To achieve the purpose of destroying the system consensus process, both Sybil parent node and child node will behave as malicious nodes that will not obey the pre-designed control rules and update their state information arbitrarily. This paper assumes that the information generated and sent by the Sybil child node is the same as that of its own Sybil parent node.

3. Node Verification Mechanism and Consensus Algorithm

Here, we consider a MAS of N agents cooperating over a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. All the agents take the following form:

$$x_i[k+1] = x_i[k] + u_i[k], \tag{1}$$

where $k \in \mathbb{N}_0$, $i = 1, 2, \ldots, N$, $x_i[k] \in \mathbb{R}$ is the state of node i at time step k, and $u_i[k]$ is the control input of node i to be designed.

3.1. Information Processing and Verification Framework

The information that the agent communicates in the network consists of two parts, namely, the status value in integer form and the validation value in decimal form. The status value in integer form is achieved by designing an appropriate quantizer, while the validation value in decimal form is realized by designing a reasonable random distribution function.

3.1.1. State Value Probabilistic Quantization

Inspired by the consensus problem for MASs with quantized communication, in this section we propose a distributed local quantizer $q(\cdot) : \mathbb{R} \to \mathbb{Z}$ aiming to transform each agent's real-valued state to a quantized integer value.

Each agent adopts a uniform integer quantization function $q(\cdot)$ with quantization step Δ:

$$q(y) = s\Delta, \quad (s - \frac{1}{2}) \leq y \leq s + \frac{1}{2} \tag{2}$$

where $y \in \mathbb{R}$ is the real-valued state input, and s is the quantization level. In this problem, we only set quantization step $\Delta = 1$, as we wanted each agent to have an integer state value.

Consider the deterministic quantization error $e(y)$, which is defined as:

$$e(y) = y - q(y). \tag{3}$$

According to Equation (2), it is easy to find that $e(y)$ is variable in the interval $[-1/2, 1/2]$. Then, the quantizer for the MAS (1) is presented as follows:

$$q(y) = \begin{cases} s & \text{with probability} \quad \lceil y \rceil - y \\ s+1 & \text{with probability} \quad y - \lfloor y \rfloor \end{cases} \tag{4}$$

where the quantizer uses the percentage of the quantization step size occupied by the quantization error as the probability to select the upper and lower bounds of the quantization result. Combining the control input with the quantizer $q(\cdot)$ yields

$$u_i[k] = q(\sum_{j \in \mathcal{N}_i} a_{ij}(x_j[k] - x_i[k])), \tag{5}$$

where $a_{ij}[k]$ is the (i, j) entry of the adjacency matrix $\mathcal{A}$ of the graph at time k. We assume that the probabilistic quantizer in each node is independent but share the same quantization step at each time. Thus, the control law (5) can be implemented in a distributed fashion.

Next, we will present an identity verification mechanism using random labels in the following subsection, aiming to tackle Sybil child nodes' fabricating identities in the network.

3.1.2. Verification Using Random Labels

Using information labels to verify a node's identity is one of the typical methods in distributed networks against malicious attacks. It usually requests a specific label verification algorithm and labels information storage space. It should be noted that if a Sybil child node copies the legal label of their parent, the verification mechanism of the fixed label will be invalid. Thus, we develop a random decimal label instead of a fixed one to ensure that the label can be uniquely generated, used, and thrown away, not taking up any extra memory space. With the quantizer transforming the real-valued input into an integer value, the decimal part is used to restore the label information. Thus, the message sent by a normal node i to its neighbors is composed of the latest integer state value and its label value, i.e.,

$$\hat{x}_i[k] = x_i[k] + L_i[k], \quad i \in \mathcal{V}_n \tag{6}$$

where $\hat{x}_i[k]$ represents the message sent by node i, and $L_i[k]$ represents the decimal label generated by random sampling at time step k. Equation (7) below shows the messages sent by Sybil nodes in the network:

$$\hat{x}_i[k] = \begin{cases} x_i[k] + L_i[k] & i \in \mathcal{V}_{Sp} \\ x_i[k] + L_p[k] & i \in \mathcal{V}_{Sc}^p, p \in \mathcal{V}_{Sp} \end{cases} \tag{7}$$

where $\mathcal{V}_{Sc}^p$ denotes the set of Sybil child nodes generated by the Sybil parent node p. For the Sybil child node, due to the lack of a physical entity, we assume that all the child nodes in $\mathcal{V}_{Sc}^p$ use the same label, generated by its Sybil parent node.

Remark 2. *Each normal node will generate a random decimal label before it sends information to its neighbors each time. Then, the generated label is combined with the quantized state value to generate new information for transmission. The advantage of such processing is that no additional computing equipment is required and the computing consumption within the node itself is reduced.*

Since sending the messages of agents uses the integer value plus decimal value label, we have $\lfloor \hat{x}_i[k] \rfloor = x_i[k]$. Combining Equation (6), it is easy to find that the following formula is established.

$$L_i[k] = \hat{x}_i[k] - \lfloor \hat{x}_i[k] \rfloor. \tag{8}$$

All Sybil parent nodes have a physical entity that is used for executing the corresponding label generation algorithm, while all Sybil child nodes have no exclusive labels for their spoof identities. This is the break-point that we use to distinguish the Sybil child nodes. We give the relevant label generation requirement of such a quantization network in the face of Sybil attacks below:

- To avoid affecting the status value during the process of quantification, the label value shall be strictly less than one quantization step;
- To avoid different nodes generating the same label, the random label range of one agent shall be an open interval;
- To make labels time-sensitive, one node generates a random label at each time step.

According to the above requirements for labels, we set the sample range of random labels in one quantization step, i.e., $\Delta = 1$. Then, according to the number of agents, we ensure that the random distribution range can not coincide. When all agents generate labels by sampling from one normal distribution, the labels of different agents have a chance to be identical. Therefore, we use the sampling method based on a disjoint truncated normal distribution to generate labels to avoid being identified. Each agent generates a random label by sampling distribution to select one item from the range of legal values, using the probability density function as the probability of selection.

Figure 2 is a schematic diagram of our sampling process. In the figure, the sum of random label sample ranges of n agents is one quantization step $\Delta = 1$ in one time step, and

those ranges have the same length on the axis, i.e., $r_{L_i} = \frac{1}{n}, \forall i \in \mathcal{V}$, where $n = |\mathcal{V}_n \cup \mathcal{V}_{Sp}|$. As we know, the Sybil parent node is the first compromised agent. Thus, parameter n is the number of agents in the original system. According to the total number of agents n and the value of quantization step $\Delta = 1$, we have the maximum and minimum of the label of each node at time step k as $L_{i,\min} = \frac{i-1}{n}$ and $L_{i,\max} = \frac{i}{n}$.

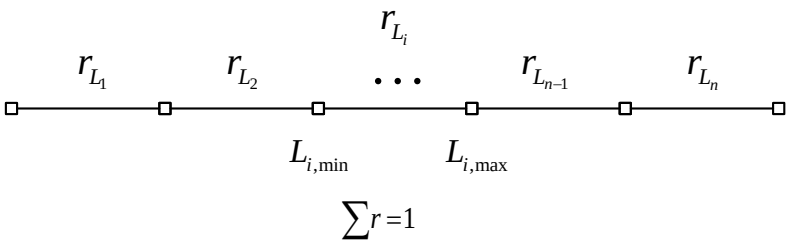


Figure 2. Random label sample ranges distributed along the length of quantization step $\Delta = 1$.

The mean and variance of truncated normal distribution based on r_{L_i} are $\mu_i = \frac{L_{i,max}+L_{i,min}}{2}$ and $\sigma = \frac{|r_{L_i}|}{2}$, respectively. To eliminate the probabilities of the same labels, the random label value distribution range of agent i is modified to an open interval $(L_{i,\min}, L_{i,\max})$. Thus, the probability density function of the corresponding label value is given as

$$f'(L_i) = \begin{cases} c_i \frac{1}{\sqrt{2\pi}\sigma} e^{-0.5\left(\frac{L_i-\mu_i}{\sigma}\right)^2}, & L_{i,min} < L_i < L_{i,max} \\ 0, & others \end{cases} \tag{9}$$

where μ_i and σ represent the mean of i' label and standard deviation, respectively. Parameter c_i is used to make sure the cumulative probability of $f'(L_i)$ is 1 in the interval $(L_{i,\min}, L_{i,\max})$. For the truncated normal distribution in the $(L_{i,\min}, L_{i,\max})$, $F'(L_i) = c_i[F(L_i) - F(L_{i,\min})]$; thus, we have parameter $c_i^{-1} = F(L_{i,\min} \leqslant L_i \leqslant L_{i,\max})$. According to the total number of agents and the quantization length, parameters in Equation (9) can be set as follows:

$$\mu_i = \frac{2i-1}{2n}, \tag{10}$$

$$\sigma = \frac{1}{2n}, \tag{11}$$

$$L_{i,min} = \frac{i-1}{n}, \tag{12}$$

$$L_{i,max} = \frac{i}{n}, \tag{13}$$

$$c_i^{-1} = \int_{L_{i,\min}}^{L_{i\max}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{1}{2}\left(\frac{L_i-\mu_i}{\sigma}\right)^2\right) \mathrm{d}L_i. \tag{14}$$

The truncated normal distribution function F' can be calculated according to the following three situations:

1. If $L_i \leq L_{i,\min}$, then $F'(L_i) = \int_{-\infty}^{L_i} f'(u)\mathrm{d}u = 0$.
2. If $L_{i,min} < L_i < L_{i,max}$, then
$$F'(L_i) = \int_{-\infty}^{L_i} f'(u)\mathrm{d}u = \int_{L_{i,\min}}^{L_i} f'(u)\mathrm{d}u$$
$$= c_i \left[\int_{-\infty}^{L_i} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-0.5\left(\frac{u-\mu_i}{\sigma}\right)^2} \mathrm{d}u - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-0.5\left(\frac{u-\mu_i}{\sigma}\right)^2} \mathrm{d}u \right]$$
$$= c_i[F(L_i) - F(L_{i,\min})].$$

3. If $L_i \geq L_{i,\max}$, then
$F'(L_i) = \int_{L_{i,\min}}^{L_i} f'(u)\mathrm{d}u = \int_{L_{i,\min}}^{L_{i,\max}} f'(u)\mathrm{d}u = c_i[F(L_{i,\max}) - F(L_{i,\min})] = 1.$

With label sampling from such a truncated normal distribution, each agent can generate the corresponding random label. The random label generation model complies with a truncated normal probability density function as

$$L_i \sim \psi(\frac{2i-1}{2n}, (\frac{1}{2n})^2, \frac{i-1}{n}, \frac{i}{n}; L_i). \tag{15}$$

In Figure 3, we give an example of the probability distribution function of 10 agents in quantization step $\Delta = 1$, where different colors represent the label probability density curve of 10 different nodes. The random label of each agent is distributed in the corresponding truncated range centered on the half of r_{L_i}.

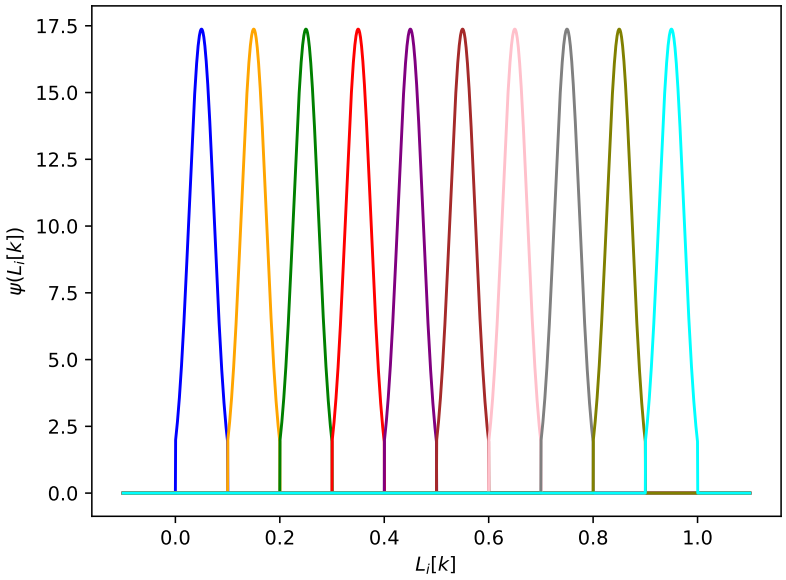


Figure 3. Probability distribution function of 10 agents' labels.

The main idea of the label generation design is that the label value merges with the quantized status value and becomes a quantified part. Since each agent label is sampled from the corresponding truncated normal distribution range $L_i[k] \in (L_{i,\min}, L_{i,\max})$, one can identify each individual agent in the MAS. At the same time, since the child node and the parent node have the same label value, the redundant fake child nodes that have been generated can be easily distinguished and ignored by the normal nodes. After label verification and the removal of redundant individuals with the same label, the total number of nodes in the network is maintained at n.

In the following subsection, we provide the solution to the resilient consensus problem with Sybil nodes in the network.

3.2. Distributed Consensus Control Law Design

Recall that the malicious nodes in MAS shall be divided into a Sybil parent node and Sybil child node. Thus, $\mathcal{V}_S = \mathcal{V}_{Sp} \cup \mathcal{V}_{Sc}$. Combined with the local quantification process, the updated rule for the nodes can be written in the following form:

$$x_i[k+1] = \begin{cases} q(\sum\limits_{j \in \mathcal{N}_i \cup \{i\}} a_{ij}[k] x_j[k]) & i \in \mathcal{V}_n \\ x_i[k] + u_i[k] & i \in \mathcal{V}_s \end{cases} \tag{16}$$

The existing typical methods against Sybil attacks are adopting a strict identity distribution mechanism or using physical fingerprints as authentication. Such methods always

require extra modules and computing consumption. Before giving our distributed control protocol, we first need to introduce the notion of resilient consensus for the network of probabilistic quantized agents in the face of malicious attacks.

Definition 8. *(Quantized Resilient Consensus) For a quantization consensus system under malicious attacks, if the following conditions are met, then the network is said to reach a quantized resilient consensus:*

1. ***Safety condition:*** *For each set of quantized values of the normal agents, with the maximum value $M(0)$ and minimum value $m(0)$ of the initial state values of normal nodes, there exists a set $\mathcal{S} \subseteq [m(0), M(0)]$ such that for all normal agents $i \in \mathcal{V}_n$, it holds that $x_i[k] \in \mathcal{S}$ for $k \geq 0$.*
2. ***Agreement condition:*** *There exists a finite time $k_c \geq 0$ such that Prob$\{x^{\mathcal{V}_n}(k_c) \in \mathcal{C}_{\mathcal{V}_n} | x(0)\} = 1$, where the consensus $\mathcal{C}_{\mathcal{V}_n}$ is defined as*

$$\mathcal{C}_{\mathcal{V}_n} = \{x \in \mathbb{Z}^{\mathcal{V}_n} | x_1 = \cdots = x_{\mathcal{V}_n}\} \tag{17}$$

Next, we give our distributed consensus algorithm, which is mainly inspired by the weighted mean subsequence reduced (QW-MSR) algorithm [37], and incorporates the previously designed label generation and verification mechanism, referred to as the QWL-MSR algorithm.

Description of QWL-MSR Algorithm

At each time step k, the normal node i will receive the messages of its neighbors. We assume that there are at most f Sybil parent nodes among the node i' neighbors, and each Sybil parent node can generate up to n Sybil child nodes. Therefore, node i will receive at most $nf + f$ pieces of malicious messages. The pseudocode of QWL-MSR can be seen in Algorithm 1.

Algorithm 1 QWL-MSR

Input: $i, f, n, \mathcal{N}_i, \mathcal{A}, x_i[k], k$
Output: $x_i[k+1]$

$L_i[k] = rand(x)$
$x = \psi^{-1}(\frac{2i-1}{2n}, (\frac{1}{2n})^2, \frac{i-1}{n}, \frac{i}{n}; L_i)$
$\hat{x}_i[k] = x_i[k] + L_i[k]$
Send $MSG_i[k] = \{k, i, \hat{x}_i[k]\}$ to neighbor nodes
Empty LBL_i, ST_i
Receive $MSG_j[k]$ from neighbors, $j \in \mathcal{N}_i$
for $j \in \mathcal{N}_i$ **do**
  $L_j[k] = \hat{x}_j[k] - \lfloor \hat{x}_j[k] \rfloor$
  $x_j[k] = \lfloor \hat{x}_j[k] \rfloor$
  $LBL_i \leftarrow$ Append $L_j[k]$
end for
$\mathcal{N}_S[k] \leftarrow Label\ Verification(LBL_i)$
$ST_i \leftarrow$ Append $x_j[k]$ where j not in $\mathcal{N}_S[k]$
$\mathcal{N}_S[k] \leftarrow$ Append $State\ Filtration(x_i[k], ST_i, f)$
$x_i[k+1] = q(\sum_{j \in \{\mathcal{N}_i \cup \{i\}\} \setminus \mathcal{N}_S[k]} a_{ij}[k] x_j[k])$
return $x_i[k+1]$

In Algorithm 1, node i first generates a truncated normal distribution function based on the input values, and then generates label $L_i[k]$ by sampling in the generated function. Subsequently, node i merges the state value and the label value and sends it to its neighbor nodes. When node i receives the message value $MSG_j[k]$ from its neighbor node $j \in \mathcal{N}_i$, the label value and state value will be separated. The Sybil nodes are excluded into the Sybil node-set $\mathcal{N}_S$ though label verification and state filtration. Under this mechanism, node i will

update its own state value without interference from Sybil nodes. Specifically, the following five steps can be used to describe the whole process of node information processing.

1. **Label Generation:** At each time step k, each node $i \in \mathcal{V}$ (including Sybil nodes) calculates and generates an exclusive random label $L_i[k]$;
2. **Message Exchange:** Once node i updates its own state value, it will combine the message value with its newly generated status value and label value, i.e., $\hat{x}_i[k] = x_i[k] + L_i[k]$. Then, it sends this message to its neighbors, and receives the message values from its neighbors $j \in \mathcal{N}_i$.
3. **Label Verification:** After step 2, node i executes the label stripping algorithm (8) to the message value of the nodes in the list, then puts the labels into list LBL_i. If there are any nodes with the same label value in the list, node i groups these nodes into the Sybil neighbor set $\mathcal{N}_S[k]$, but only reserves the first node with that label value. It sorts the remaining state values into a descending order list ST_i at the same time.
4. **State Filtration:** If there are less than f nodes for which state values are strictly larger or smaller than $x_i[k]$ in the list ST_i, node i groups these nodes into the Sybil neighbor set $\mathcal{N}_S[k]$. Otherwise, it groups the f largest and f smallest values of the nodes in the list into the set $\mathcal{N}_S[k]$.
5. **State Update:** According to the set $\mathcal{N}_S[k]$ obtained by the above operation, node i applies the following update:

$$x_i[k+1] = q\Big(\sum_{j \in \{\mathcal{N}_i \cup \{i\}\} \setminus \mathcal{N}_S[k]} a_{ij}[k] x_j[k]\Big). \tag{18}$$

Figure 4 shows the data flow model of QWL-MSR for normal agent i. In the figure, the state value $x_i[k]$ of node i, is subtracted from each of the other states, including its own state in the memory. The resulting relative states are verified and filtered by order. Finally, the remaining elements are weighted, summed, and quantized to the next state.

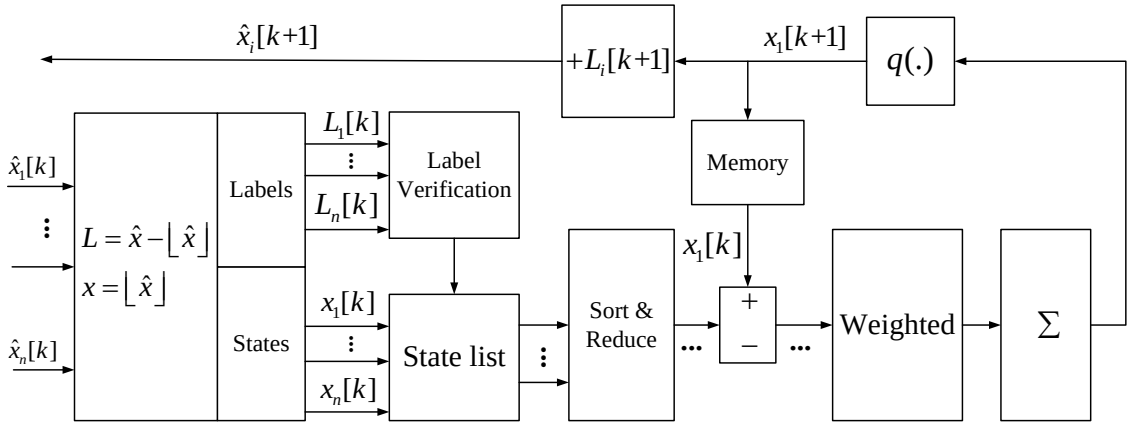


Figure 4. Data flow model of QWL-MSR for node i.

4. Main Results

We now present our main theorems for quantized resilient consensus in the presence of Sybil attacks. The network underlying a robust graph that guarantees sufficient connectivity and information redundancy for agents strengthens the incoming counter attacks. Let us revisit the definition of security interval S by

$$\mathcal{S} = [\min x^{\mathcal{V}_n}[0], \max x^{\mathcal{V}_n}[0]], \tag{19}$$

where $\min x^{\mathcal{V}_n}[0]$ and $\max x^{\mathcal{V}_n}[0]$ represent the minimum and maximum initial state values of all nodes in the MAS, respectively. Note that we assume that the initial value of each agent is an integer, i.e., $x_i[0] \in \mathbb{Z}, i \in \mathcal{V}$.

Since we choose to run probability quantization during the node state value update process, the discussions of consensus under probabilistic conditions are inevitable. The following three conditions, which are argued in [37] for the probabilistic quantized consensus problem, are also adopted in our paper.

1. C1. There is a finite state set $\mathcal{S}$ for each normal node i, such that, for any $k \geq 0$, $x_i[k] \in \mathcal{S}$.
2. C2. There exists a finite time k_x, such that, for state value $x[k] = x_0 \in \mathcal{S}$, $\text{Prob}\{x^{\mathcal{V}_n}[k + k_x] \in \mathcal{C}_{\mathcal{V}_n} | x[k] = x_0\} > 0$.
3. C3. If $x[k] \in \mathcal{C}_{\mathcal{V}_n}$, then $x[k'] \in \mathcal{C}_{\mathcal{V}_n}, \forall k' > k$.

Theorem 1. *Under the f-parent total Sybil attack model, the MAS (1) for which each node updates its state according to the QWL-MSR algorithm with parameter f reaches quantized resilient consensus almost surely if, and only if, the system's network topology is $(f+1, f+1)$-robust.*

Proof of Theorem 1. (Necessity) Although all Sybil child nodes could be removed after the step of label verification, there are still f values of Sybil parent nodes retained in the network. Then, in the step of state filtration, if the underlying graph $\mathcal{G}$ is not $(f+1, f+1)$-robust, the node-set includes two disjointed and nonempty subsets, $\mathcal{V}_1$ and $\mathcal{V}_2$, that do not meet any conditions in Definition 4. Thus, the incoming edges of any disjointed and nonempty subset $\mathcal{V}_1$ or $\mathcal{V}_2$ is less than $f+1$. With the QWL-MSR algorithm, the normal node will ignore all of its neighbors' values that are different from its state and will not update its state, which will cause the state value consensus process to fail to complete.

(Sufficiency) We first show that the state value of each normal node in the update process satisfies the safety condition. We denote the minimum and maximum values of normal nodes at time step k by

$$\underline{x}[k] = \min x^{\mathcal{V}_n}[k], \ \overline{x}[k] = \max x^{\mathcal{V}_n}[k]. \tag{20}$$

Since we consider the f-parent total Sybil attack model, that means there are at most f Sybil parent nodes in the network. Each Sybil parent node will fabricate n Sybil child nodes. Therefore, it is equivalent to the existence of at most $(n+1)f$ malicious nodes in the entire network. At each time k, all the Sybil child nodes' values will be ignored in the step of label verification of the QWL-MSR algorithm. Then, in the step of state filtration of our algorithm, the normal node i will remove the first f and last f values in the sorted list of its all neighbors. This ensures that the information used in the update of node i is not beyond the interval $[\underline{x}[k], \overline{x}[k]]$.

According to the update rule (18) and the remain neighbor set $\phi_i[k] = \{\mathcal{N}_i \cup \{i\}\} \backslash \mathcal{N}_S[k]\}$, the value of next time step of normal node i will be upper bounded by

$$x_i[k+1] \leq q(\sum_{j \in \phi_i[k]} a_{ij}[k] x_j[k]) \leq q(\sum_{j \in \phi_i[k]} a_{ij}[k] \overline{x}[k]) = q(\overline{x}[k]) = \overline{x}[k].$$

This implies that $x_i[k+1] \leq \overline{x}[k]$ and $\overline{x}[k]$ is a monotonically non-increasing function of time. Hence, node i's value at the next time step has a lower bound value function $\underline{x}[k]$, i.e.,

$$x_i[k+1] \geq q(\sum_{j \in \phi_i[k]} a_{ij}[k] x_j[k]) \geq q(\sum_{j \in \phi_i[k]} a_{ij}[k] \overline{x}[k]) = q(\underline{x}[k]) = \underline{x}[k].$$

Consequently, node i's state value satisfies $x_i[k] \in [\underline{x}[k], \overline{x}[k]]$ for all time steps, which implies that the safety condition of MAS (1) is ensured.

We next show that the state value of each normal node $i \in \mathcal{V}_n$ in the update process (18) satisfies the agreement condition. Let the set $\mathcal{S}$ be a set of all normal nodes' state values at all times. Since $\underline{x}[k]$ and $\overline{x}[k]$ are both monotone functions, they will eventually converge to the values $\underline{x}^*$ and $\overline{x}^*$ with probability at a finite time k', respectively. We will show that $\underline{x}^*$ and $\overline{x}^*$ tend to be the same as the time becomes infinite. We prove this by contradiction. Assume

that $\underline{x}^* < \overline{x}^*$. In this case, the shortest distance between $\underline{x}^*$ and $\overline{x}^*$ is one quantization step Δ, thus $\overline{x}[k] - \underline{x}[k] = \Delta$. As the state values tend to remain unchanged, we denote the set of nodes (including Sybil nodes) for which state values are greater than or equal to $\overline{x}^*$ and the set of nodes (including Sybil nodes) for which state values are less than or equal to $\underline{x}^*$ by $\mathcal{X}_1[k]$ and $\mathcal{X}_2[k]$, respectively, i.e.,

$$\begin{aligned} \mathcal{X}_1[k] &= \{i \in \mathcal{V} : x_i[k] \geq \overline{x}^*\}, \\ \mathcal{X}_2[k] &= \{i \in \mathcal{V} : x_i[k] \leq \underline{x}^*\}. \end{aligned} \tag{21}$$

Since the network of MAS (1) satisfies a $(f+1, f+1)$-robust graph, we know that for any pair of nonempty disjointed subsets of $\mathcal{V}$, at least one of the three conditions in Definition 4 is satisfied. We let $\mathcal{X}_1[k]$ and $\mathcal{X}_2[k]$ be such a pair of subsets. In this case, there always exists a normal node i, either in $\mathcal{X}_1[k]$ or $\mathcal{X}_2[k]$, with $(f+1)$ incoming edges from $\mathcal{V}\backslash\mathcal{X}_1[k]$ or $\mathcal{V}\backslash\mathcal{X}_2[k]$. We suppose node i in $\mathcal{X}_1[k]$ has this property and its state value is $x_i[k] = \overline{x}^*$. If there are any Sybil nodes around node i, in the step of label verification, node i will ignore at most nf incoming edges from them. Then, in the step of state filtration, node i will neglect at most f values from $\mathcal{V}\backslash\mathcal{X}_1[k]$ and at most f values larger than $\overline{x}^*$ which are sent from f Sybil nodes. That means node i will receive at least one node's value in $\mathcal{X}_2[k]$, which is smaller than $\overline{x}^*$. Thus, by the update rule (18), we have

$$x_i[k+1] \leq q((1-a)\overline{x}^* + a\underline{x}^*) = q((1-a)\overline{x}^* + a(\underline{x}^* - 1) = q(\overline{x}^* - a), \tag{22}$$

where $a < 1$ is node i' self-weight. With random quantizer (4), we have $q(\overline{x}^* - a) = \overline{x}^* - 1$ with probability $1 - a$. Thus,

$$x_i[k] \leq \overline{x} - 1 = \underline{x}^*. \tag{23}$$

This indicates that there is at least one node in $\mathcal{X}_1[k]$ that will update and decrease its state value at time step k.

We now show that there is a positive probability that none of the nodes in $\mathcal{V}\backslash\mathcal{X}_1[k]$ will be placed into set $\mathcal{X}_1[k+1]$ at the next time step. According to (18) and (22), the normal node i will choose the value $\overline{x}^* - 1$ with probability a. In another words, node i will not be placed into $\mathcal{X}_1[k+1]$ with probability $1 - a$. By applying the same argument, nodes in $\mathcal{V}\backslash\mathcal{X}_2[k]$ will not be placed into $\mathcal{X}_2[k+1]$ according to their updated state values.

By the above analysis, we conclude that for any $k \geq k' + |\mathcal{V}_n|$, one of sets ($\mathcal{X}_1[k]$ and $\mathcal{X}_2[k]$) will be empty with a positive probability. Clearly, it arrives at a contradiction with the definition of a $(f+1, f+1)$-robust graph. Hence, we proved $\underline{x}^* = \overline{x}^*$.

According to the above discussion, one can clearly see that the safety condition of node i in MAS (1) also satisfies the C1 in the quantized consensus problem, while the agreement condition of node i in MAS (1) also satisfies the C2 and C3 in the quantized consensus problem. Thus, the proof is completed. □

Theorem 2. *Under the f-parent local Sybil attack model, the MAS (1) that each normal node updates its state value with according to the QWL-MSR algorithm with parameter f reaches quantized resilient consensus almost surely if, and only if, the network's topology is $(2f+1)$-robust.*

Proof of Theorem 2. (Necessity) We will prove by contradiction. Assume the network topology does not satisfy a $(2f+1)$-robust graph; each node in the nonempty disjoint subsets $\mathcal{V}_1$ or $\mathcal{V}_2$ will have at most f neighbors from the outside. Although the Sybil child nodes could be removed after the step of label verification, the normal nodes in $\mathcal{V}_1$ and $\mathcal{V}_2$ will ignore all of the neighbors that have different values. This will cause the normal nodes not to update their state values. This also means that consensus cannot be achieved.

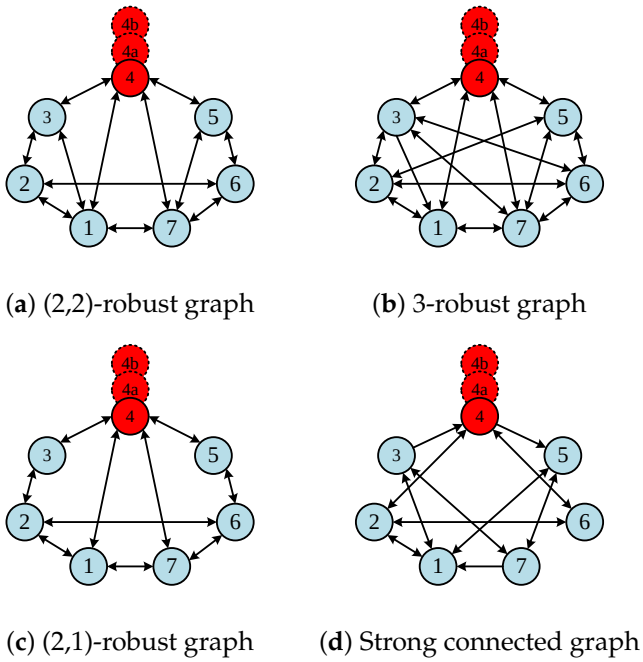
(Sufficiency) The proof is similar to the proof of the sufficiency of Theorem 1 and hence is omitted here. Note that by the steps of label verification and state filtration of QWL-MSR

algorithm, the safety condition and agreement condition are guaranteed if the network is $(2f+1)$-robust. □

5. Numerical Simulation

In this section, a numerical simulation is proposed to illustrate the effectiveness of the theoretical results for MAS under f-parent total model and f-parent local Sybil attack model, respectively.

Consider the MAS with $n = 7$ agents. The connections among agents are represented in Figure 5a,b. According to Definition 4, one can verify that the graph in Figure 5a is $(2,2)$-robust, and the graph in Figure 5b is 3-robust. Suppose node 4 is infected by a Sybil attack and becomes a malicious Sybil parent node. Then, node 4 generates two Sybil child nodes, 4a and 4b. The figure shows that the Sybil child and Sybil parent nodes share the same adjacency matrix. Hence, the normal nodes connected to the Sybil parent node will also receive malicious information from Sybil child nodes. The initial values of all nodes are $x[0] = [2,3,4,10,5,6,7]^T$. Thus, the safety interval is $[2,7]$. Let all Sybil nodes maintain their values $x_4[k] = x_{4a}[k] = x_{4b}[k] = 10$ for the entirety of the time step k. In this scenario, the adversary is aimed at driving the system out of the safe set $[2,7]$.



(**a**) (2,2)-robust graph (**b**) 3-robust graph

(**c**) (2,1)-robust graph (**d**) Strong connected graph

Figure 5. Network topology of MAS with 7 agents.

First, we will check the state value trajectories of all nodes applying the existing QW-MSR algorithm in the $(2,2)$-robust graph under 1-parent total Sybil attack. It is observed in Figure 6 that all normal nodes are misguided by the malicious values and reach the consensus value $x^* = 10$, which is out of the safety interval $[2,7]$. This means that the traditional QW-MSR algorithm is insufficient for such Sybil attacks.

Next, we examine the performance of the case with the same network communication topology, but each node runs our proposed QWL-MSR algorithm. First, each node generates its own label according to a truncated normal distribution in the value range of the corresponding label. With a total number of agents $n = 7$, the sampling of each agent's label conforms to the truncated normal distribution $L_i[k] \sim \psi(\frac{2i-1}{14}, (\frac{1}{14})^2, \frac{i-1}{7}, \frac{i}{7}; L_i[k])$. Figure 7a shows the probability distribution function curve of random label sampling for each agent. The scatter points in Figure 7b more intuitively indicate that the random labels of each agent generated at each time step k are distributed in the corresponding range and will not overlap with each other.

Under the one-parent total Sybil attack, with the QWL-MSR algorithm applied to the MAS of seven agents, the trajectory of each agent state value is shown in Figure 8. From the figure, we can see that all normal agents reach a consensus value $x^* = 6$ after 38 steps. This simulation result verifies the validity of Theorem 1.

Now, we consider the case when the MAS topology satisfies a 3-robust graph (see Figure 5b). The attack model, in this case, is a one-parent local Sybil attack model. The simulation results are shown in Figure 9. The figure shows that all normal agents acquire a consensus value $x^* = 6$ after 19 time steps. This simulation result verifies the validity of Theorem 2.

In order to verify the necessity of network robustness for our proposed algorithm, we design a (2,1)-robust graph by deleting links (1,3), (3,1), (5,7) and (7,5) from Figure 5b. The newly designed network communication topology is shown in Figure 5c. Likewise, we newly design a graph (see Figure 5d) that no longer satisfies 3-robust but that is still strongly connected by reducing links. Under the same network attack situation and running the same control algorithm, the state value trajectories of all normal agents are shown in Figures 10 and 11, respectively. It can be seen from the simulation results that the node state value cannot reach consensus, which further verifies the correctness of our theories.

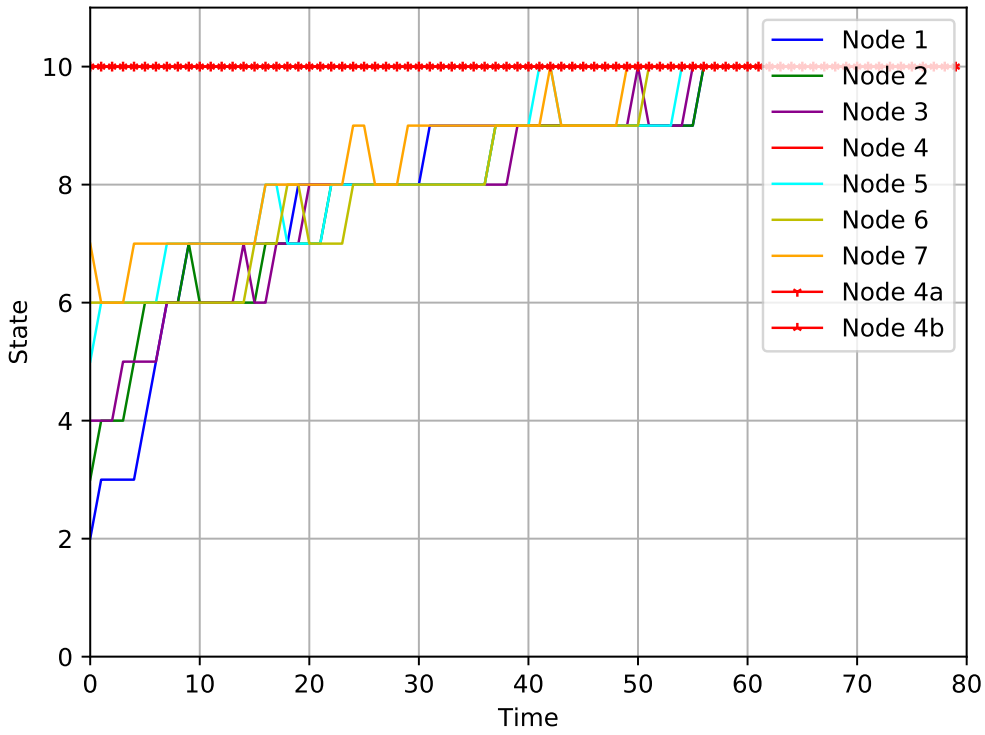


Figure 6. The MAS with QW-MSR algorithm fails to reach consensus under Sybil attacks.

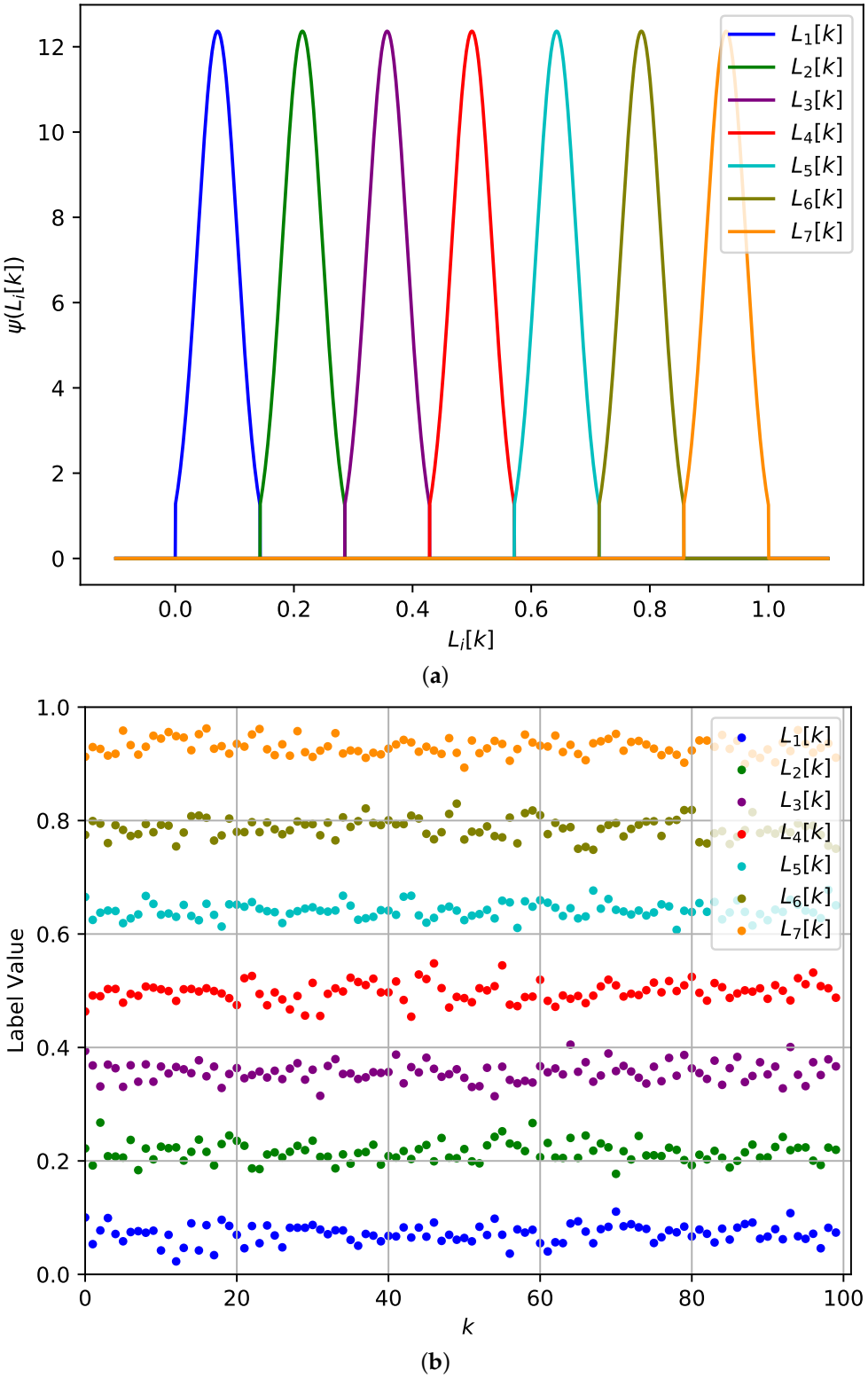


Figure 7. Random labels of 7 agents. (**a**) The sampling interval of each node. (**b**) The distribution of label values at each time step.

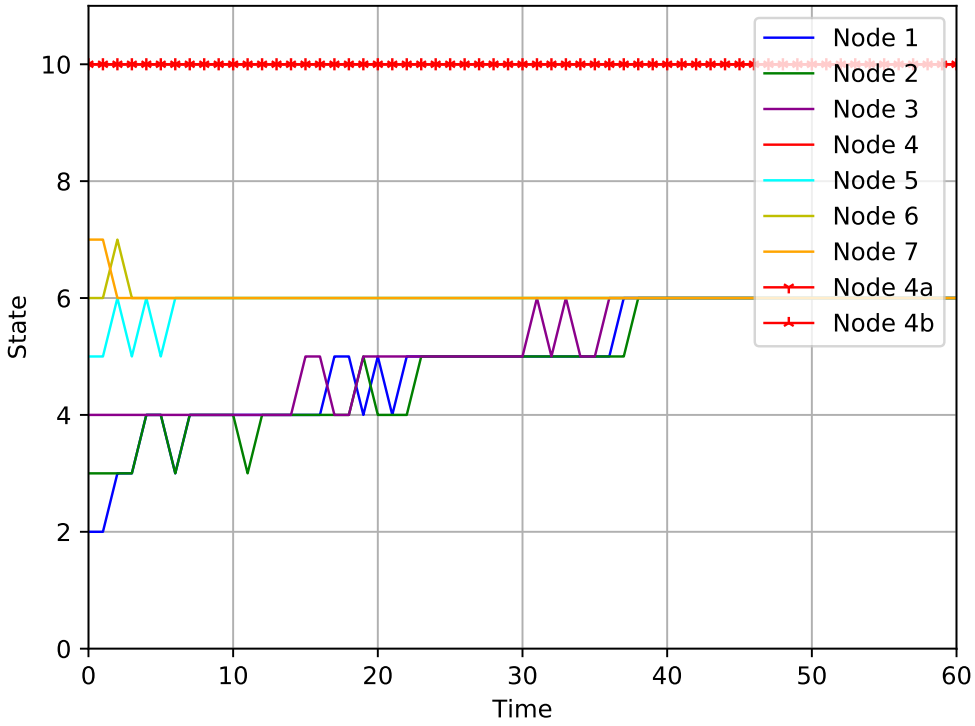


Figure 8. Simulation results of our proposed algorithm under $(2, 2)$-robust network.

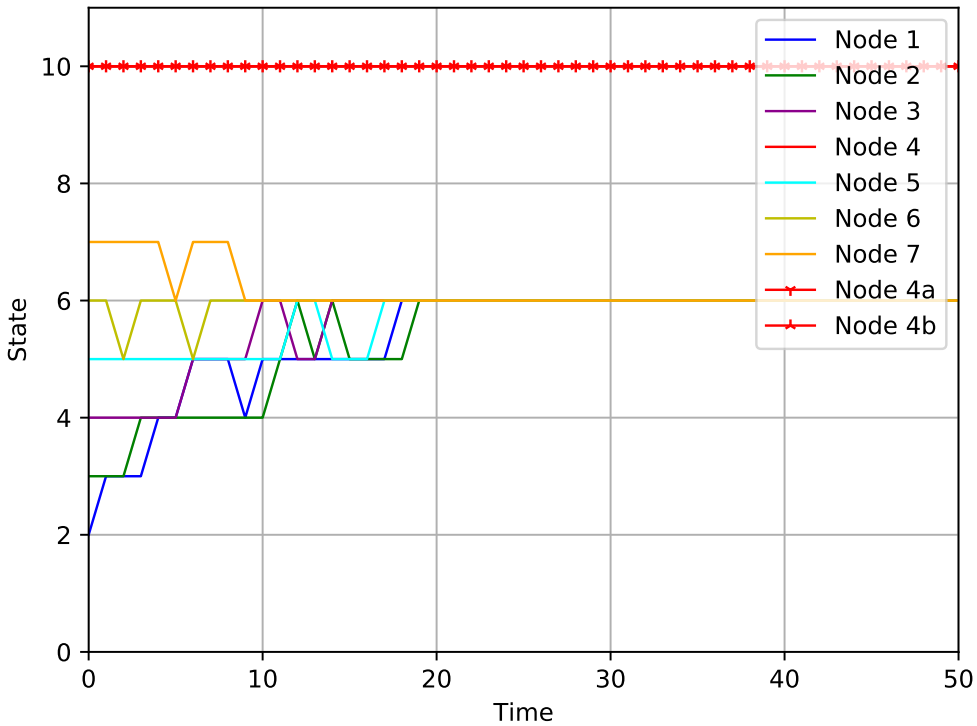


Figure 9. Simulation results of our proposed algorithm under 3-robust network.

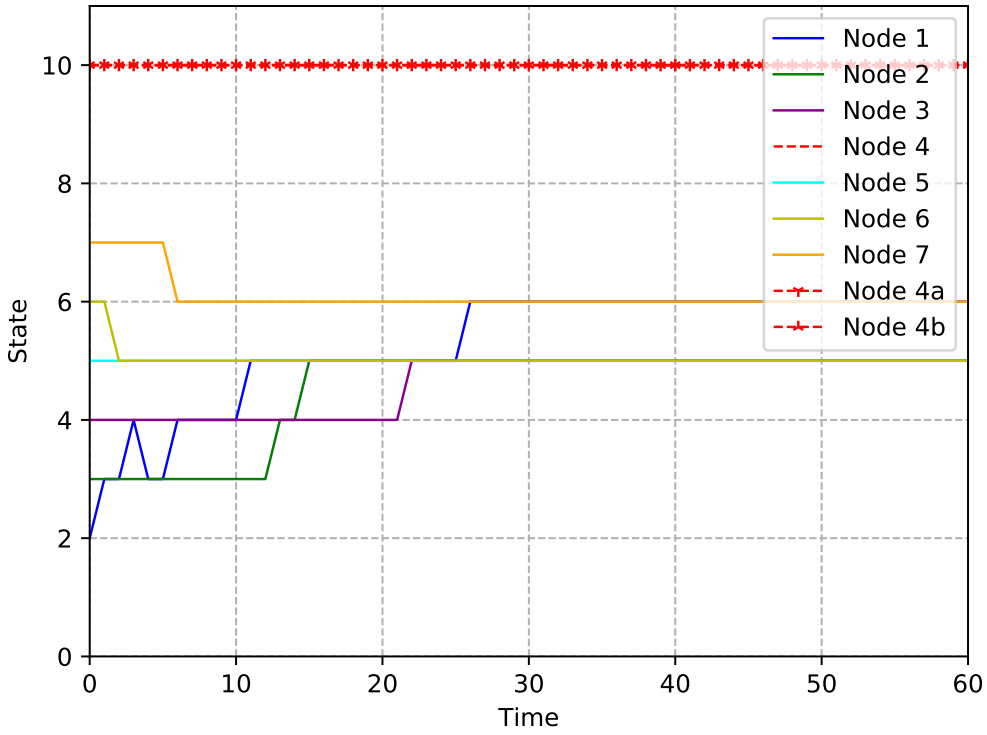


Figure 10. Simulation results of proposed algorithm under (2,1)-robust network.

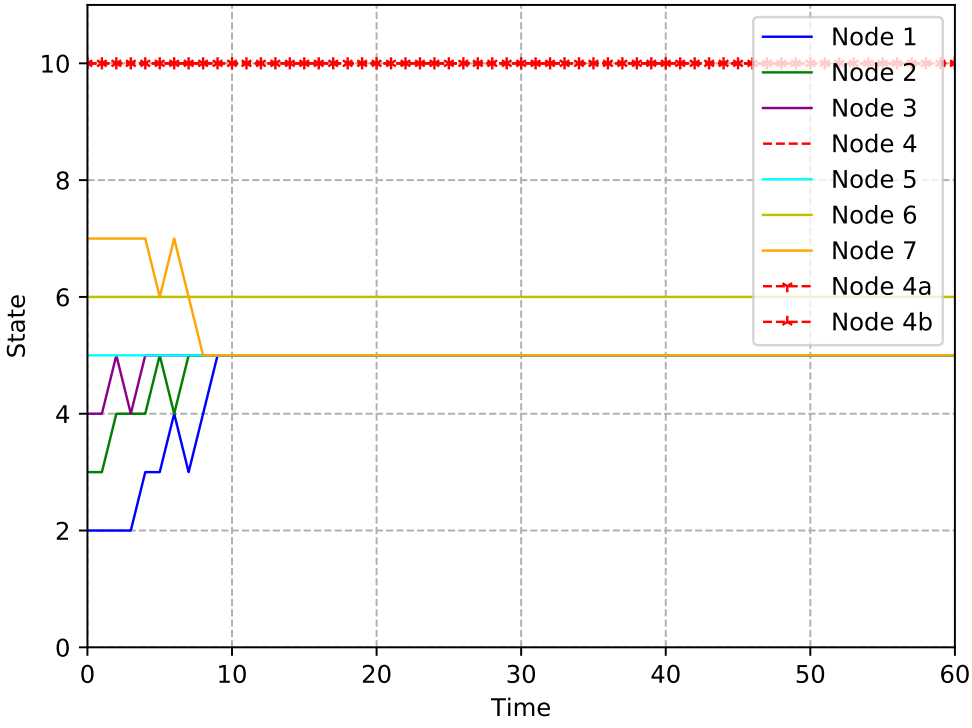


Figure 11. Simulation results of proposed algorithm under strongly connected network.

6. Conclusions

In this paper, we considered the distributed resilient consensus problem of MASs under Sybil attacks. A consensus algorithm based on random label values was applied to each agent to determine valid neighbor agents and realize the state value convergence under Sybil attacks. The proposed algorithm is computationally lightweight and suitable for use in distributed MASs. Using graph theory and mathematical analysis, we have proved the

effectiveness of the consensus algorithm under consideration. Numerical simulations are also provided to confirm our methods.

Author Contributions: Data curation, X.D.; Funding acquisition, Y.W. and M.X.; Investigation, M.X.; Methodology, X.D. and Y.W.; Resources, N.Z.; Supervision, M.X. and N.Z.; Writing—original draft, X.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China: 61803135, and Zhejiang Province Public Welfare Technology Application Research Project: LGF21F020011.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, Y.; Ma, J.; Wang, L. Consensus of Hybrid Multi-Agent Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1359–1365. [CrossRef] [PubMed]
2. Dong, X.; Hu, G. Time-Varying Output Formation for Linear Multiagent Systems via Dynamic Output Feedback Control. *IEEE Trans. Control. Netw. Syst.* **2017**, *4*, 236–245. [CrossRef]
3. Meng, D.; Jia, Y.; Du, J. Finite-time consensus protocols for networks of dynamic agents by terminal iterative learning. *Int. J. Syst. Sci.* **2014**, *45*, 2435–2446. [CrossRef]
4. LeBlanc, H.J.; Zhang, H.; Koutsoukos, X.; Sundaram, S. Resilient Asymptotic Consensus in Robust Networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 766–781. [CrossRef]
5. Zhang, D.; Feng, G. A New Switched System Approach to Leader–Follower Consensus of Heterogeneous Linear Multiagent Systems With DoS Attack. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 1258–1266. [CrossRef]
6. He, W.; Mo, Z.; Han, Q.L.; Qian, F. Secure impulsive synchronization in Lipschitz-type multi-agent systems subject to deception attacks. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 1326–1334. [CrossRef]
7. Shang, Y. Resilient Consensus for Robust Multiplex Networks with Asymmetric Confidence Intervals. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 65–74. [CrossRef]
8. Vaidya, N.H.; Tseng, L.; Liang, G. Iterative approximate byzantine consensus in arbitrary directed graphs. In Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing, Madeira, Portugal, 16–18 July 2012; pp. 365–374.
9. Dibaji, S.M.; Ishii, H. Resilient multi-agent consensus with asynchrony and delayed information. *IFAC-PapersOnLine* **2015**, *48*, 28–33. [CrossRef]
10. Wang, D.; Zheng, N.; Xu, M.; Wu, Y.; Hu, Q.; Wang, G. Resilient Privacy-Preserving Average Consensus for Multi-agent Systems under Attacks. In Proceedings of the 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), Shenzhen, China, 13–15 December 2020; pp. 1399–1405.
11. Wen, G.; Lv, Y.; Zhou, J.; Fu, J. Sufficient and Necessary Condition for Resilient Consensus under Time-varying Topologies. In Proceedings of the 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), IEEE, Guangzhou, China, 13–15 November 2020; pp. 84–89.
12. Shang, Y. Resilient consensus of switched multi-agent systems. *Syst. Control Lett.* **2018**, *122*, 12–18. [CrossRef]
13. LeBlanc, H.J.; Koutsoukos, X.D. Consensus in networked multi-agent systems with adversaries. In Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control, Chicago, IL, USA, 12–14 April 2011; pp. 281–290.
14. Gil, S.; Kumar, S.; Mazumder, M.; Katabi, D.; Rus, D. Guaranteeing spoof-resilient multi-robot networks. *Auton. Robot.* **2017**, *41*, 1383–1400. [CrossRef]
15. Wu, Y.; He, X. Secure Consensus Control for Multi-Agent Systems with Attacks and Communication Delays. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 136–142. [CrossRef]
16. Cheng, C.F.; Tsai, K.T. A recursive Byzantine-resilient protocol. *J. Netw. Comput. Appl.* **2015**, *48*, 87–98. [CrossRef]
17. Li, C.; Hurfin, M.; Wang, Y. Approximate Byzantine consensus in sparse, mobile ad hoc networks. *J. Parallel Distrib. Comput.* **2014**, *74*, 2860–2871. [CrossRef]
18. Kailkhura, B.; Brahma, S.; Varshney, P.K. Data Falsification Attacks on Consensus-Based Detection Systems. *IEEE Trans. Signal Inf. Process. Over Networks* **2017**, *3*, 145–158. [CrossRef]
19. Shang, Y. Consensus of hybrid multi-agent systems with malicious nodes. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *67*, 685–689. [CrossRef]
20. Wu, Y.; Xu, M.; Zheng, N.; He, X. Event-Triggered Resilient Consensus for Multi-Agent Networks Under Deception Attacks. *IEEE Access* **2020**, *8*, 78121–78129. [CrossRef]
21. Tseng, L.; Vaidya, N.H. Fault-tolerant consensus in directed graphs. In Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, Donostia-San Sebastián, Spain, 21–23 July 2015; pp. 451–460.
22. Li, J. Fault tolerant consensus of multi-agent systems with linear dynamics. *Math. Probl. Eng.* **2013**, *2013*, 465671. [CrossRef]
23. Deng, C.; Yang, G.H. Distributed adaptive fault-tolerant control approach to cooperative output regulation for linear multi-agent systems. *Automatica* **2019**, *103*, 62–68. [CrossRef]
24. Pu, C. Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses. *IEEE Internet Things J.* **2020**, *7*, 4937–4949. [CrossRef]

25. Douceur, J.R. The sybil attack. In Proceedings of the International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 7–8 March 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 251–260.
26. Dong, W.; Liu, X. Robust and secure time-synchronization against sybil attacks for sensor networks. *IEEE Trans. Ind. Informatics* **2015**, *11*, 1482–1491. [CrossRef]
27. Jamshidi, M.; Darwesh, A.M.; Lorenc, A.; Ranjbari, M.; Meybodi, M.R. A precise algorithm for detecting malicious sybil nodes in mobile wireless sensor networks. *IEIE Trans. Smart Process. Comput.* **2018**, *7*, 457–466. [CrossRef]
28. Huang, Y.; Wang, W.; Wang, Y.; Jiang, T.; Zhang, Q. Lightweight sybil-resilient multi-robot networks by multipath manipulation. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 2185–2193.
29. Wheeler, T.; Bharathi, E.; Gil, S. Switching topology for resilient consensus using Wi-Fi signals. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2018–2024.
30. Biryukov, A.; Feher, D. ReCon: Sybil-resistant consensus from reputation. *Pervasive Mob. Comput.* **2020**, *61*, 101109. [CrossRef]
31. Levine, B.N.; Shields, C.; Margolin, N.B. A survey of solutions to the sybil attack. *Univ. Mass. Amherst. Amherst.* **2006**, *7*, 224.
32. Renganathan, V.; Summers, T. Spoof resilient coordination for distributed multi-robot systems. In Proceedings of the 2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Los Angeles, CA, USA, 4–5 December 2017; pp. 135–141.
33. Jiang, Y.; Li, D.; Wu, X.; Xu, Y. Resilient Consensus of Second-Order Multi-Agent Systems Based on WiFi Signals. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 5865–5870.
34. Breslaw, J.A. Random sampling from a truncated multivariate normal distribution. *Appl. Math. Lett.* **1994**, *7*, 1–6. [CrossRef]
35. Burkardt, J. *The Truncated Normal Distribution*; Department of Scientific Computing, Florida State University: Tallahassee, FL, USA, 2014.
36. Wu, Y.; He, X.; Liu, S. Resilient consensus for multi-agent systems with quantized communication. In Proceedings of the American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 5136–5140.
37. Dibaji, S.M.; Ishii, H.; Tempo, R. Resilient Randomized Quantized Consensus. *IEEE Trans. Autom. Control.* **2018**, *63*, 2508–2522. [CrossRef]