中国科学: 信息科学 2024 年 第 54 卷 第 4 期: 927–943
SCIENTIA SINICA Informationis
论文

隐私保护下的多智能体系统弹性一致性控制

张冲, 伍益明*, 徐明, 郑宁

杭州电子科技大学网络空间安全学院, 杭州 310018
* 通信作者. E-mail: ymwu@hdu.edu.cn

收稿日期: 2023–05–05; 修回日期: 2023–07–26; 接受日期: 2023–08–30; 网络出版日期: 2024–04–11

国家自然科学基金 (批准号: 62073109) 和浙江省公益技术应用研究计划 (批准号: LGF21F020011) 资助项目

摘要 针对多智能体系统通信网络遭受欺骗攻击与隐私窃取问题, 本文提出了一种新的具备隐私保护能力的多智能体系统弹性一致性算法. 首先考虑到多智能体系统在信息传输环节直接将真实状态信息暴露给外界的问题, 设计了一种基于状态分配思想的节点状态信息处理机制, 来保证系统中节点状态信息隐私的安全性. 其次考虑到外部攻击者对系统收敛的影响, 在状态分配机制上进一步引入了检测环节, 从而保证系统安全收敛. 然后通过数学理论分析证明了算法能够有效保护节点初始状态信息隐私的同时能够保证系统实现弹性一致性收敛. 最后通过数值仿真实验与对比实验, 进一步验证了本文算法的有效性.

关键词 多智能体系统, 弹性一致性, 欺骗攻击, 隐私保护, 网络安全

1 引言

近年来, 随着具有计算能力和通信感知能力的智能设备的广泛普及, 多智能体协同控制技术得到了研究人员的广泛关注, 并在很多控制系统中得到了应用, 如智能电网 [1]、多机器人系统 [2]、智慧交通 [3] 等. 一致性控制问题作为多智能体系统分布式协同控制领域中的一个基础研究内容, 要求多智能体系统仅通过利用本地信息就某些属性达成一致 [4]. 目前已广泛应用于无人机编队控制 [5]、传感器网络 [6] 等领域. 主要包括平均一致性 [7]、领导跟随一致性 [8,9] 以及弹性一致性 [10] 等. 在传统的一致性控制中, 智能体的邻居可以通过信息交互直接接收其状态信息. 在这种情况下, 恶意窃听者可以轻易地获取网络中每个智能体的初始状态信息, 然而这很可能会暴露节点的隐私. 在一些特殊的实际应用场景中, 这种情况是不被允许的, 智能体不希望将自身真实状态信息透露给其他邻居, 如社交网络中的意见值 [11]、电力系统中的成本信息 [12] 等. 由于智能体隐私的泄露可能会导致严重的经济损失, 因此在算法中考虑如何保护智能体敏感的隐私信息就显得至关重要.

引用格式: 张冲, 伍益明, 徐明, 等. 隐私保护下的多智能体系统弹性一致性控制. 中国科学: 信息科学, 2024, 54: 927–943, doi: 10.1360/SSI-2023-0124
Zhang C, Wu Y M, Xu M, et al. Resilient consensus control of multi-agent systems under privacy protection (in Chinese). Sci Sin Inform, 2024, 54: 927–943, doi: 10.1360/SSI-2023-0124

基于此问题, 衍生出了多智能体系统隐私保护的相关研究. 一种典型方案是将密码学机制加入到系统更新中. 部分研究工作 [13, 14] 利用同态加密技术将智能体的状态信息进行加密使得其他智能体只能得到处理过的加密值, 但这也使得智能体计算量增大导致系统资源开销增加. 相较于文献 [13,14], 文献 [15] 中基于 Shamir 共享密钥的方法相对减少了计算资源消耗, 但由于多智能体系统资源有限且对时延要求较高, 此方法仍不适用于大部分实际系统. 另一种较为有影响力的研究方法是向系统中引入差分隐私机制, 文献 [16] 在智能体交换信息的过程中加入了噪声, 使得窃听者难以准确推测出智能体真实的初始状态, 然而这种基于差分隐私的方法难以保证系统收敛到准确的目标. 而文献 [17] 通过在系统更新过程中添加呈指数衰减的零和高斯 (Gauss) 噪声, 使得基于差分隐私的多智能体系统中每个智能体能够渐进地收敛到精确的平均值, 但仍对智能体计算能力有一定要求. 此外, 一些研究工作也进行了一些其他尝试. 文献 [18] 通过向通信网络中添加随机偏移量的方法来保证节点的隐私, 文献 [19] 通过令智能体在更新时向邻居发送附加消息, 解决了不平衡有向图中的隐私保护问题. 与文献 [19] 相比, 文献 [20] 针对通信开销较小的不平衡有向图提出了隐私保护算法. 文献 [21] 提出了一种基于状态分解思想的隐私保护机制, 其中每个智能体将初始状态值分解为两个随机的子状态, 一个子状态参与邻居间状态更新, 另一个子状态只参与内部计算, 对于外界是完全不可见的, 因此该方法能够有效保护节点初始状态信息的隐私. 文献 [22] 通过在最大一致性更新中加入随机值来隐藏真实的状态信息, 保证系统在达成最大一致性收敛时有效保护最大状态值所有者的身份信息.

然而上述研究成果均假设在理想的安全环境下, 未考虑存在网络攻击时的情况, 难以保证在受限环境中系统的正常运行. 这导致传统多智能体系统一致性控制研究方法不再适用, 因此考虑抗外部网络攻击的安全一致性研究发展迅速并取得了显著成果.

当前研究领域所聚焦的攻击类型主要为欺骗攻击与 DoS 攻击. 近些年来, 针对欺骗攻击的研究成果不断产出, 相关学者着手从不同的角度开展研究. 文献 [23] 利用两跳通信信息使节点能够检测判断其邻居节点是否正常并对恶意节点或故障节点造成的误差进行补偿, 从而使系统能够达成精确的共识. 弹性一致性问题最早可以追溯到文献 [24,25], 文献 [10] 给出了平均子序列缩减 (mean-subsequence-reduced, MSR) 算法, 使得网络在存在至多 f 个恶意节点的情况下能够达成一致性. 文献 [26] 在 MSR 算法的基础上进行了优化改进, 提出了加权平均子序列缩减 (weighted-mean-subsequence-reduced, W-MSR) 算法, 使得节点在移除异常值操作时进一步地保留了正常值, 此外本文给出了网络鲁棒性的拓扑属性, 在局部算法的研究中发挥了重要作用.

然而通过上述文献不难发现, 现有文献往往单一考虑系统隐私保护问题或单纯进行抗网络攻击研究, 少有将两者同时考虑的研究. 事实上, 同时考虑隐私保护与抗网络攻击势必会为研究带来一定的困难. 如上文提到文献 [23] 利用两跳信息使节点判断其邻居节点是否正常, 利用节点传递的真实信息来计算推测邻居值下一时刻的变化并以此作为判断邻居节点正常与否的标志, 若引入隐私保护, 则其节点信息的真实将难以保证, 原有的检测攻击手段将会失效. 基于差分隐私机制的相关算法, 能够将节点传输的信息有效隐藏起来, 但这也使得恶意节点的异常信息更难以被区别出来, 抗攻击研究难度加大.

基于以上总结与分析, 本文致力于研究欺骗攻击下具备隐私保护能力的多智能体系统弹性一致性问题, 从而补充完善现有一致性控制的研究成果. 本文主要贡献如下:

(1) 针对多智能体系统中存在的窃听者, 提出了一种新的基于状态分配的节点信息隐私保护策略, 增大了窃听者窃取节点初始状态隐私的难度;

(2) 针对欺骗攻击在系统更新时的恶意行为, 在状态分配的基础上加入了检测环节, 使节点具备了识别异常信息值的能力;

(3) 与文献 [10,26] 相比, 在移除异常值的操作上加入了检测环节, 进一步保留了节点间的交互信息, 减少了信息资源的浪费;

(4) 与文献 [10,26] 相比, 在保证抗网络攻击能力的同时, 能够有效保护系统中节点的隐私.

本文内容组织结构如下. 第 2 节给出了所需要的图论知识, 对拟解决的问题以及攻击模型进行了描述. 第 3 节给出了相关定义, 提出了系统在存在欺骗攻击下多智能体一致性控制及隐私保护算法, 同时对算法的隐私保护能力与一致性收敛进行了分析. 第 4 节通过数值仿真实验验证所提算法的有效性. 第 5 节对本文进行了总结.

2 预备知识与问题描述

2.1 图论基础

考虑一个由 n 个智能体组成的多智能体系统, 其底层图可抽象地用一个无向网络拓扑 $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{A})$ 表示, 其中 $\mathcal{V}=\{1,2,\ldots,n\}$ 表示系统中节点集合, $\mathcal{E}=\mathcal{V}\times\mathcal{V}$ 表示边集. $(i,j)\in\mathcal{E}$ 表示节点 j 可收到节点 i 发送来的信息, $a_{ij}>0$; 否则, $a_{ij}=0$. 节点 i 的邻居集合由所有指向它的节点组成, 记为 $N_i=\{j\in\mathcal{V}|(j,i)\in\mathcal{E}\}$. 节点 i 的度表示其邻居数量, 记为 $d_i=|N_i|$. 对于本文提出的算法, 本文借助图的鲁棒性用来表征网络的拓扑特性. 鲁棒性的概念首先由文献 [27] 引入, 用于分析实值一阶多智能体系统的弹性一致性, 本文使用对其总结的 (r,s)- 鲁棒图概念.

定义1 ((r,s)- 鲁棒图) 如果对于每对非空不相交子集 $\mathcal{V}_1$, $\mathcal{V}_2\subset\mathcal{V}$, 至少满足以下条件之一:

(1) $\mathcal{X}_{\mathcal{V}_1}^r=\mathcal{V}_1$;

(2) $\mathcal{X}_{\mathcal{V}_2}^r=\mathcal{V}_2$;

(3) $|\mathcal{X}_{\mathcal{V}_1}^r|+|\mathcal{X}_{\mathcal{V}_2}^r|\geqslant s$.

那么称图 $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{A})$ 是 (r,s)- 鲁棒图. 其中 $r,s<n$, $\mathcal{X}_{\mathcal{V}_l}^r$ 是 $\mathcal{V}_l$ 中至少有来自外部 r 条入度边的节点集. 特别地, 当 $s=1$ 时, 图 $\mathcal{G}$ 是 $(r,1)$- 鲁棒图也称为 r- 鲁棒图.

引理1 一个 (r,s)- 鲁棒图 $\mathcal{G}$ 满足以下条件:

(1) 图 $\mathcal{G}$ 是 (r',s')- 鲁棒图, 其中 $0\leqslant r'\leqslant r$, $1\leqslant s'\leqslant s$.

(2) 图 $\mathcal{G}$ 是 $(r-1,s+1)$- 鲁棒图.

(3) 图 $\mathcal{G}$ 具有一个生成树. 此外, 当且仅当其有一个生成树时, 图 $\mathcal{G}$ 是 1- 鲁棒图.

(4) $r\leqslant\lceil n/2\rceil$. 进一步地, 如果 $r=\lceil n/2\rceil$, 那么图 $\mathcal{G}$ 是一个完全图.

此外, 如果图 $\mathcal{G}$ 是 $(r+s-1)$- 鲁棒图, 那么它是 (r,s)- 鲁棒图.

2.2 问题描述

假设网络中有 n 个节点, 每个节点 $i\in\mathcal{V}$ 在初始时刻拥有一个只有自身知道的初始状态值 $x_i[0]\in R$. 在后续更新中, 其在 k 时刻的状态值表示为 $x_i[k]$, $k\in\mathbb{Z}_+$. 每个智能体节点根据自身与来自邻居的状态值以及规定的控制规则来进行状态更新, 控制规则建模为

$$x_i[k+1]=F_i\left(\{x_j[k]\}\right),\ j\in N_i\cup\{i\},\ i\in\mathcal{V},\tag{1}$$

其中 $x_j[k]$ 表示在 k 时刻节点 i 的邻居 j 传输来的状态值. 更新协议 $F_i(\cdot)$ 可以为任意函数, 对于不同的节点可能具有不同的协议, 这取决于节点在网络中的身份.

本文研究对象为由 n 个智能体节点组成的一阶离散多智能体系统, 其预先设定的动力学方程为

$$x_i[k+1]=\sum_{j\in N_i\cup\{i\}} w_{ij}[k]\,x_j[k],\ i,j\in\mathcal{V}, \tag{2}$$

其中 $w_{ij}[k]$ 表示在 k 时刻智能体节点 i 与 j 之间的链路权重.

2.3 攻击模型

本文考虑的攻击模型为多智能体系统中存在的某些节点在系统更新过程中使用一些与预定规则不同的特殊更新协议从而扰乱系统的正常运行. 这些节点在未发起恶意行为时与正常节点无异, 其行为不端主要表现在更新方式上, 通过向正常节点传输恶意状态值来影响正常节点的更新, 因此正常节点无法分辨恶意节点的存在与数量.

定义2 若智能体节点遵循预先设定的动力学方程 (2) 更新状态值, 则称其为正常节点, 正常节点集合表示为 $\mathcal{N}$, 其数量表示为 $n_{\mathcal{N}}=|\mathcal{N}|$. 否则, 称其为恶意节点, 表示为 $\mathcal{M}=\mathcal{V}\backslash\mathcal{N}$, 其数量 $n_{\mathcal{M}}=|\mathcal{M}|$.

所以节点的更新规则可以改写成以下形式:

$$x_i[k+1]=\begin{cases}\sum_{j\in N_i\cup\{i\}} w_{ij}[k]\,x_j[k], i\in\mathcal{N};\\ F_i(\{x_j[k]\}), i\in\mathcal{M}.\end{cases} \tag{3}$$

假设网络中行为不端的恶意节点数量有限, 其数量上限用 f 表示.

定义3 对于分布式系统, 若在任意时刻网络中恶意节点的数量 $n_{\mathcal{M}}\leqslant f$, 则称此类攻击为 f- 全局攻击. 另一方面, 对于任一智能体节点, 若在任意时刻其邻居节点中恶意节点的数量 $n_{\mathcal{M}}\leqslant f$, 则称此类攻击为 f- 局部攻击.

2.4 多智能体弹性一致性

本文令 $M[k]$ 和 $m[k]$ 分别表示正常节点在时刻 k 的最大值和最小值.

定义4 (弹性一致性) 如果对于正常节点的任何初始状态, 恶意节点的任何可能集合以及任何恶意行为都满足以下条件:

- 安全性条件. 对于正常节点的每组初始状态值, 存在一个集合 $\mathcal{S}\in[m[0],M[0]]$, 对于所有正常节点 $i\in\mathcal{N}$, 使得其最终状态 $x_i[k]\in\mathcal{S}$, $k\to\infty$.
- 一致性条件. 存在一个有限时间 $k_c\geqslant 0$, 使得状态值 $x_i[k_c]=x^*$, $x^*\in\mathcal{S}$, 满足当 $k>k_c$ 时, $x_1[k]=\cdots=x_i[k]=x^*, i\in\mathcal{N}$.

那么称该网络达成了弹性一致性.

3 算法设计与分析

在给出本文算法之前, 对于本文所考虑的窃听者类型以及隐私保护给出以下定义.

定义5 (内部窃听者) 一个内部窃听者是网络中的一个节点, 它正确地遵循系统所有协议步骤但记录其收到的所有数据并试图使用这些数据来推测其他节点的私有信息. 此外, 它还可以与网络中其他内部窃听者相互串通, 即多个内部窃听者可以共享接收到的数据.

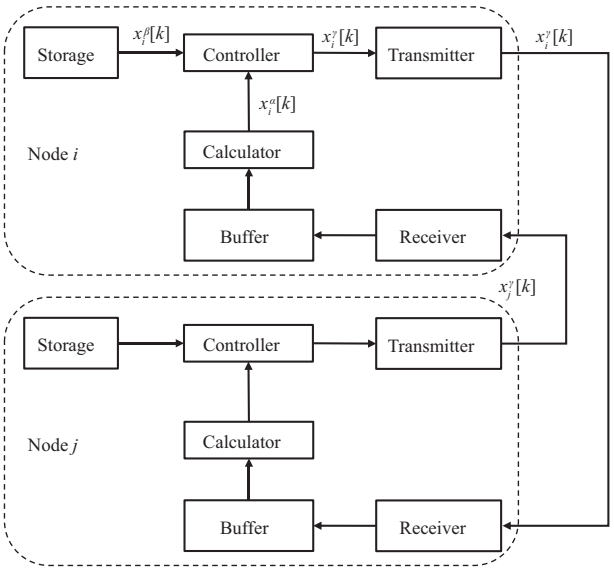

图 1 状态分配机制示例图

Figure 1 Example diagram of state allocation mechanism

定义6 (外部窃听者) 外部窃听者是具有全部网络拓扑知识的外部攻击者, 它不了解系统所采用的隐私保护方法但知道每对节点之间的网络拓扑和权重, 能够窃听部分传输数据.

定义7 (隐私保护) 如果窃听者无法以任何确定的精度准确估计节点 i 的初始状态信息 $x_i[0]$ 的值, 则称节点 i 的隐私得到了保护.

假设1 外部窃听者能力有限, 每次监听行为最多只能持续 k_m 个时刻, 之后需要花费一定的时间为下次监听做准备. 即在系统更新收敛周期 T_{k_w} 中, 系统在第 k_w 个时刻达成收敛, 外部窃听者最多只能在有限个连续时刻 k_m 中持续监听并记录节点传输的数据. 假设窃听者在系统更新中最多能够发起 m 次监听行为, 那么 $m \times k_m < k_w$.

注1 在实际应用环境中, 外部窃听者具有整个网络的拓扑信息并且可以对网络中的任意节点进行监听并记录数据, 但这需要花费一定的代价或资源. 外部窃听者在时刻 k 对某个节点 i 进行监听的过程中, 可以截获节点 i 与其所有邻居节点 $j \in N_i$ 之间交换的数据 $x_i[k]$ 及 $x_j[k]$. 因此对于被窃听的节点 i, 外部窃听者需将 $x_i[k]$ 及 $x_j[k]$ 记录在内存中, 对于窃听者来说是一种资源损耗, 而窃听者为了达成目的必须合理控制自身的资源损耗. 对于被窃听节点不确定的更新周期以及基于对监听行为的得失权衡, 窃听者需要对其恶意行为的资源消耗进行限制, 因此假设 1 是合理的.

3.1 状态分配机制设计

为了保护系统中节点初始状态的隐私以及使系统具备抗攻击能力, 本文设计了状态分配机制并在其基础上加入了检测操作. 状态分配机制示例图如图 1 所示, 同时根据状态分配机制, 本文提出具备隐私保护能力的多智能体系统弹性一致性算法, 如算法 1 所示.

为了便于表述, 本文首先进行以下定义: $x_i^\alpha[k]$ 为节点 i 在 k 时刻与邻居信息计算得到的计算值; $x_i^\beta[k]$ 为节点 i 在 k 时刻从初始状态值分配出来加入更新的分配值; $x_i^\gamma[k]$ 为节点 i 在 k 时刻发送给邻居的交互值.

如图 1 所示, 对于每个正常节点 $i \in \mathcal{N}$, 其自身维持一个存储器用于存储自身初始状态值的副本

算法 1 基于状态分配与检测的多智能体系统弹性一致性算法

输入: 节点初始状态值 $x_i[0]$, 分配比率 $0<\eta<1$, 容忍参数 $0<c\leqslant 1$;

初始化: 节点 i 分配一个空内存用于存储副本 $x_i'[k]$, 其中 $x_i'[0]=x_i[0]$; 设置 $x_i^{\alpha}[0]=0$;

1: 节点 i 随机分配 $x_i'[k]$ 的一部分作为分配值 $x_i^{\beta}[k]$, $x_i^{\beta}[k]\in\left[0,\eta\times x_i'[k]\right]$;

2: 节点 i 计算用于发送给邻居的交互值 $x_i^{\gamma}[k]$:

$$x_i^{\gamma}[k]=x_i^{\alpha}[k]+x_i^{\beta}[k]; \tag{4}$$

3: 节点 i 向邻居 $j\in N_i$ 发送交互值 $x_i^{\gamma}[k]$ 同时收到 $x_j^{\gamma}[k]$ 并将所有收到的信息值升序排列;

4: 根据网络中恶意节点数量上限 f 及排序好的列表, 按照式 (8) 进行计算 $\mathcal{F}_{\rm Sec}[k]$;

5: $f1=f2=f$. 其中 $f1$ 表示列表左侧中可能移除的信息值数量, $f2$ 表示列表右侧中可能移除的信息值数量;

6: **while** $f1>0$ **do**

7: **if** $|\mathcal{J}_i^{f1+1}[k]-\mathcal{J}_i^{f1}[k]|>\mathcal{F}_{\rm Sec}[k]$ **then**

8: 删除列表左侧 $f1$ 个值;

9: **else**

10: $f1=f1-1$;

11: **end if**

12: **end while**

13: **while** $f2>0$ **do**

14: **if** $|\mathcal{J}_i^{d_i-f2}[k]-\mathcal{J}_i^{d_i-f2+1}[k]|>\mathcal{F}_{\rm Sec}[k]$ **then**

15: 删除列表右侧 $f2$ 个值;

16: **else**

17: $f2=f2-1$;

18: **end if**

19: **end while**

20: 节点 i 根据更新后的列表计算下一时刻的计算值 $x_i^{\alpha}[k+1]$:

$$x_i^{\alpha}[k+1]=\sum_{j\in\mathcal{J}_i[k]\setminus\mathcal{R}_i[k]\cup i}w_{ij}[k]\,x_j^{\gamma}[k]; \tag{5}$$

其中 $\mathcal{R}_i[k]$ 表示节点 i 在 k 时刻删除值的集合, $w_{ij}[k]=\frac{1}{1+d_i-|\mathcal{R}_i[k]|}$.

21: 节点 i 更新内存至 $x_i'[k+1]$:

$$x_i'[k+1]=x_i'[k]-x_i^{\beta}[k]; \tag{6}$$

22: 节点 i 更新自身状态值至 $x_i[k+1]$:

$$x_i[k+1]=x_i^{\alpha}[k+1]+x_i'[k+1]; \tag{7}$$

输出: 更新后的状态值 $x_i[k+1]$.

$x_i'[0]$. 在每个时刻 k, 节点 i 从存储器中分配出副本 $x_i'[k]$ 的一部分作为分配值 $x_i^{\beta}[k]$ 发送到控制器并更新存储器中的副本为 $x_i'[k+1]$; 与此同时, 节点 i 的接收器将 $k-1$ 时刻收到邻居节点 $j\in N_i$ 发送来的交互值 $x_j^{\gamma}[k-1]$ 发送到缓存器, 缓存器将所有邻居发送来的交互值整理成列表并进行处理, 之后缓存器将处理完毕的列表发送给计算器进行计算得到计算值 $x_i^{\alpha}[k]$ 并将其发送到控制器; 控制器将存储器发送来的分配值 $x_i^{\beta}[k]$ 与计算器发送来的计算值 $x_i^{\alpha}[k]$ 进行计算得到交互值 $x_i^{\gamma}[k]$ 并将其发送到发送器. 最后通过发送器将交互值 $x_i^{\gamma}[k]$ 发送给所有邻居节点.

正常节点 $i\in\mathcal{N}$ 在 k 时刻接收邻居信息并将所有邻居信息发送到缓存器之后, 节点 i 在缓存器中将所有接收到的值进行升序排列整理成列表记为 $\mathcal{J}_i[k]$, $\mathcal{J}_i^s[k]$ 表示列表中第 s 个位置的值, $1\leqslant s\leqslant d_i$. 根据网络中恶意节点的数量 f 将列表中的 $\mathcal{J}_i^{f+1}[k]$ 与 $\mathcal{J}_i^{d_i-f}[k]$ 视为正常值. 由于分配机制使得节点间的差别足够小并且随着系统更新最终会使节点间的差别为 0, 因此 $\mathcal{J}_i^{f+1}[k]$ 与 $\mathcal{J}_i^{d_i-f}[k]$ 之间的差

别足以表征正常节点可容忍的误差, 因此本文设计正常节点在 k 时刻的检测函数为

$$\mathcal{F}_{\mathrm{Sec}}[k]=c\times\left|\mathcal{J}_i^{f+1}[k]-\mathcal{J}_i^{d_i-f}[k]\right|, \tag{8}$$

其中 c 为容忍参数, $0<c\leqslant 1$.

正常节点 i 以 $\mathcal{J}_i^{f+1}[k]$ 以及 $\mathcal{J}_i^{d_i-f}[k]$ 为基准, 分别判断 $\mathcal{J}_i^{f+1}[k]$ 与 $\mathcal{J}_i^{f}[k]$ 的误差以及 $\mathcal{J}_i^{d_i-f}[k]$ 与 $\mathcal{J}_i^{d_i-f+1}[k]$ 的误差是否在 $\mathcal{F}_{\mathrm{Sec}}[k]$ 内. 若各自误差在 $\mathcal{F}_{\mathrm{Sec}}[k]$ 内, 则以其为新的基准, 继续判断其左侧或右侧相邻位置的值; 否则, 删除包括 $\mathcal{J}_i^{f+1}[k]$ 或 $\mathcal{J}_i^{d_i-f}[k]$ 在内的左侧所有值或右侧所有值.

除容忍参数 c 外, 在算法 1 中本文还设计了参数: 分配比率 η. 其中分配比率 η 决定节点在每个时刻加入系统更新的状态值的大小, η 越大则节点加入更新的状态值越大, 更新速度更快, 否则相反; 容忍参数 c 决定节点在每个时刻 k 检测区间容忍误差范围的大小, c 越大则容忍范围越大, 从式 (8) 可以看到, 检测函数 $\mathcal{F}_{\mathrm{Sec}}[k]$ 受参数 c 的直接影响, 当 $c=1$ 时, 系统视正常节点之间的差值为安全检测区间. 从算法 1 中的步骤 $1\sim 3$ 可以看出, 不同于传统算法直接将节点真实状态值加入更新, 算法 1 将节点状态值随时间逐渐加入到系统更新中. 这样做可以给窃听者极大地增加窃听负担, 降低窃听者推测节点真实信息的概率, 因此分配比率 η 越小, 隐私保护程度越强. 在后续更新中, 与文献 [10,26] 类似的是在每个时刻 k, 节点间相互收发信息后, 会将接受到邻居信息整理成列表并移除列表中可疑的极端值, 但不同的是算法 1 优化了节点移除值操作, 对可疑值的选择性更强, 避免了正常信息值的浪费. 根据计算出的 $\mathcal{F}_{\mathrm{Sec}}[k]$, 在算法步骤 $6\sim 19$ 中, 节点对两侧各 f 个可疑值逐个进行判断, 不同于文献 [10] 删除左右两侧各 f 个值与文献 [26] 删除两侧至少 f 个值, 算法 1 将删除尽可能少的信息甚至不删. 这样做的好处是, 由于网络中恶意节点的数量上限为 f, 因此至多有 f 个恶意信息, 其可能为极小值也可能为极大值, 因此列表中左侧 f 个值与右侧 f 个值都将是可疑目标, 而文献 [10] 删除 $2f$ 个值将造成至少 f 个正常信息值被浪费, 同理文献 [26] 也将造成较大的正常信息浪费. 而算法 1 根据计算出的 $\mathcal{F}_{\mathrm{Sec}}[k]$, 依次判断左右两侧 f 个值中的每一个, 而 $\mathcal{F}_{\mathrm{Sec}}[k]$ 由正常信息值计算得来, 因此其足以表征正常节点在当前时刻的可容忍的误差, 若左右两侧 f 个值都在 $\mathcal{F}_{\mathrm{Sec}}[k]$ 中则当前时刻不删除任何值, 此时的恶意信息即使保留也不会对正常更新造成影响. 容忍参数 c 将影响 $\mathcal{F}_{\mathrm{Sec}}[k]$ 范围的大小, c 越小则容忍恶意值的程度越低, 可能移除的值越多, 随着时间更新 $\mathcal{F}_{\mathrm{Sec}}[k]$ 逐渐趋于 0, 恶意信息将被准确移除. 综上所述, 与文献 [10,26] 相比, 算法 1 识别攻击者恶意信息的能力更强, 并且进一步保留了正常信息, 减少了资源浪费.

3.2 隐私保护分析

定理1 对于内部窃听者, 若网络中节点采用算法 1 进行状态值更新并且每个节点的所有邻居并非全部都是内部窃听者, 即每个节点至少有一个正常邻居, 则网络中正常节点的隐私可以得到保护.

证明 对于每个节点 i, 其在 k 时刻具有的信息有: 自身状态值 $x_i[k]$、传输给邻居的交互值 $x_i^{\gamma}[k]$、自身分配值 $x_i^{\beta}[k]$、邻居发送来的交互值 $x_j^{\gamma}[k]$, 因此每个节点 i 在 k 时刻自身可使用的信息集可表示为

$$I_i[k]=\left\{x_i[k],x_i^{\gamma}[k],x_i^{\beta}[k];x_j^{\gamma}[k]\right\},\ i\in\mathcal{V},j\in N_i. \tag{9}$$

k 时刻节点 i 向其邻居节点传输的信息只有交互值 $x_i^{\gamma}[k]$. 因此一个内部窃听者 a 在 k 时刻可使用的信息集可表示为

$$I_a[k]=\left\{x_a[k],x_a^{\gamma}[k],x_a^{\beta}[k];x_i^{\gamma}[k]\right\},\ i\in N_a. \tag{10}$$

假设一组内部窃听者 $A=\{a|a\in\mathcal{V}\}$ 在 k 时刻可使用的信息集为

$$I_A[k]=\{I_a[k]\},\ a\in A. \tag{11}$$

在该情况下, 一组内部窃听者 A 可使用的信息集表示为

$$I_A[k]=\left\{x_a[k],x_a^{\gamma}[k],x_a^{\beta}[k];x_i^{\gamma}[k]\right\},\ a\in A\cap N_i. \tag{12}$$

一组内部窃听者 A 从节点 i 处直接获取的可用信息仅有 $x_i^{\gamma}[0]$, 而 $x_i^{\gamma}[0]$ 是设计过的传输值, 由于其值是随机生成的, $x_i^{\gamma}[0]\neq x_i[0]$, 因此保证了节点真实状态信息未被泄露.

此外, 若一组内部窃听者 A 直接与节点 i 相邻且互相勾结. 由于节点 i 的邻居中至少有一个节点为正常节点, 所以一组内部窃听者 A 缺少信息 $x_l^{\gamma}[k]$, 其中 $l\in\mathcal{J}_i[k]\backslash\mathcal{R}_i[k]$, 因此一组内部窃听者 A 无法通过式 (5) 计算节点 i 下一时刻的计算值 $x_i^{\alpha}[k+1]$. 根据式 (4), 节点 i 向邻居发送的交互值 $x_i^{\gamma}[k+1]$ 为在计算值 $x_i^{\alpha}[k+1]$ 的基础上加入 $x_i^{\beta}[k+1]$, $x_i^{\beta}[k+1]$ 为节点 i 在该时刻加入更新的一部分真实初始状态值. 因此一组内部窃听者 A 无法计算 $x_i^{\alpha}[k+1]$ 便不能通过与从节点 i 得到的 $x_i^{\gamma}[k+1]$ 进行对比从而得到 $x_i^{\beta}[k+1]$, 而 $x_i^{\beta}[k+1]$ 直接与节点真实初始状态相关, 因此一组内部窃听者无法通过观测正常节点的更新来推测节点的初始状态信息, 所以正常节点的隐私可以得到保护.

定理2 对于外部窃听者, 在满足假设 1 条件下, 若网络中节点采用算法 1 进行状态值更新, 则网络中所有节点的隐私可以得到保护.

证明 根据定义 6, 外部窃听者知道网络的拓扑信息, 能够窃听部分耦合权重和传输数据. 因此外部窃听者 e 可使用全部的 $x_i^{\gamma}[0],i\in\mathcal{V}$, 由于 $x_i^{\gamma}[0]$ 是设计过的随机传输值, 外部窃听者 e 无法通过 $x_i^{\gamma}[0]$ 推测节点初始状态信息 $x_i[0]$.

此外, 由于外部窃听者 e 可以截获并记录节点 i 与其邻居 $j\in N_i$ 传输的所有数据且知道网络拓扑信息, 所以外部窃听者 e 可得到 k 时刻的 $x_i^{\gamma}[k]$, $x_j^{\gamma}[k]$ 以及 $w_{ij}[k]$. 因此, 外部窃听者 e 在 k 时刻可使用的信息集表示为

$$I_e[k]=\left\{x_i^{\gamma}[k],x_j^{\gamma}[k];w_{ij}[k]\right\},\ i\in\mathcal{V},j\in N_i. \tag{13}$$

根据式 (5), 外部窃听者 e 可估计节点 i 在 $k+1$ 时刻的计算值 $x_i^{\alpha}[k+1]$. 根据式 (4) 可知, $k+1$ 时刻节点 i 的交互值 $x_i^{\gamma}[k+1]$ 是在其计算值 $x_i^{\alpha}[k+1]$ 基础上加入了分配值 $x_i^{\beta}[k+1]$, 因此外部窃听者 e 可以通过比较 $x_i^{\alpha}[k+1]$ 与 $x_i^{\gamma}[k+1]$ 来推测 $x_i^{\beta}[k+1]$. 由于 $x_i^{\beta}[k]\in[0,\eta\times x_i'[k]]$ 且根据式 (6), 易得

$$x_i^{\beta}[0]+x_i^{\beta}[1]+\cdots+x_i^{\beta}[k]=\lim_{k\to\infty}\sum_{T=0}^{T=k}x_i^{\beta}[T]=x_i'[0]=x_i[0]. \tag{14}$$

根据假设 (1), 外部窃听者 e 在整个更新周期 T_{k_w} 中只能进行有限次监听且每次监听最多持续 k_m 个时刻. 假设窃听者 e 在更新中进行了 m_e 次监听, 每次持续 k_{m_e} 个时刻, 其中 $0\leqslant m_e\leqslant m,0\leqslant k_{m_e}\leqslant k_m$, 那么其进行监听的时间步骤总数 $m_e\times k_{m_e}\leqslant m\times k_m<k_w$. 由于外部窃听者 e 得到的分配值总个数 $m_e\times k_{m_e}<k_w$, 因此其对节点 i 的估计值 $x_i^e[0]<x_i[0]$, 所以节点的隐私可以得到保护.

3.3 弹性一致性分析

定理3 在 f- 全局攻击恶意模型下, 当且仅当网络通信拓扑满足 $(f+1,f+1)$- 鲁棒时, 使用算法 1 的多智能体网络中的所有正常节点能够实现弹性一致性.

证明 (必要性) 本文采用反证法来证明定理 3 的必要性. 如果系统通信拓扑不满足 $(f+1,f+1)$-鲁棒, 那么网络包含的两个非空不相交节点子集 $\mathcal{V}_1$ 和 $\mathcal{V}_2$ 不满足定义 1 中 3 个条件的任意一个, 因此可以得出以下结论:

(1) $|\mathcal{X}_{\mathcal{V}_1}^{f+1}|<|\mathcal{V}_1|$;

(2) $|\mathcal{X}_{\mathcal{V}_2}^{f+1}|<|\mathcal{V}_2|$;

(3) $|\mathcal{X}_{\mathcal{V}_1}^{f+1}|+|\mathcal{X}_{\mathcal{V}_2}^{f+1}|\leqslant f$.

因此对于 $i=1,2$, $\mathcal{V}_i$ 中来自 $\mathcal{V}\backslash\mathcal{V}_i$ 至少有 $f+1$ 条入度边的节点总数小于 $f+1$. 假设恶意节点都处于集合 $\mathcal{X}_{\mathcal{V}_1}^{f+1}$ 与 $\mathcal{X}_{\mathcal{V}_2}^{f+1}$ 中, 即 $\mathcal{V}_1\backslash\mathcal{V}_1^{f+1}$ 与 $\mathcal{V}_2\backslash\mathcal{V}_2^{f+1}$ 是正常节点的两个非空不相交子集. 假设子集 $\mathcal{V}_1$, $\mathcal{V}_2$ 及其余节点的值分别为 a,b,c, 其中为 $a,b,c\in\mathbb{Z}$ 且 $a<c<b$. 假设恶意节点执行欺骗攻击, 不遵循正确的更新方式, 在更新周期中始终保持自身值不变. 根据结论 (1), 可以得出在 $\mathcal{X}_{\mathcal{V}_1}^{f+1}$ 中至少有一个正常节点具有外部小于 $f+1$ 条入度边; 同理根据结论 (2), 可以得出在 $\mathcal{X}_{\mathcal{V}_2}^{f+1}$ 中至少有一个正常节点具有外部小于 $f+1$ 条入度边; 根据结论 (3), 可以得到 $\mathcal{V}_1$ 与 $\mathcal{V}_2$ 中具有外部小于 $f+1$ 条入度边的节点总数小于等于 f. 由于 $\mathcal{V}_1,\mathcal{V}_2$ 中的节点在进行更新时, 将忽略从集合外部接收到的与自身状态不同的值, 因此它们将保持自身的状态值 a 或 b 不变. 因此包含在 $\mathcal{V}_1\backslash\mathcal{V}_1^{f+1}$ 与 $\mathcal{V}_2\setminus\mathcal{V}_2^{f+1}$ 中的正常节点的值将始终保持在 a 和 b, 无法满足弹性一致性的一致性条件, 因此网络中的正常节点无法达成弹性一致性.

(充分性) 首先证明定义 4 中的安全性条件. 由于 $x_i'[k]$ 为节点 i 自身初始状态值副本, 其只参与节点内部计算. 因此根据式 (7), 节点 i 的状态值 $x_i[k]$ 受外部节点交互影响的部分只有 $x_i^{\alpha}[k]$, 所以节点 i 状态值 $x_i[k]$ 的安全性直接受 $x_i^{\alpha}[k]$ 影响. 即需要通过证明 $x_i^{\alpha}[k]$ 在合法的范围内更新变化来保证节点 i 状态值 $x_i[k]$ 的安全性.

根据式 (6) 可知, $x_i'[k]$ 将会在更新中逐渐减小至 0, 其减小速度受 $x_i^{\beta}[k]$ 影响. 而计算值 $x_i^{\alpha}[k+1]$ 的变化直接受邻居节点的交互值 $x_j^{\gamma}[k]$ 影响.

将正常节点在 k 时刻的最小交互值和最大交互值分别表示为

$$\underline{x^{\gamma}}[k]=\min x^{\gamma}[k],\ \overline{x^{\gamma}}[k]=\max x^{\gamma}[k]. \tag{15}$$

同理, 对于后文中正常节点在 k 时刻状态值的上下界, 本文定义: 假设系统中的节点 i 在 k 时刻的状态值为 $x_i[k]$, 那么该时刻系统中的状态值上界表示为 $\overline{x}[k]$, 下界表示为 $\underline{x}[k]$, 即 $x_i[k]\in[\underline{x}[k],\overline{x}[k]]$.

由于网络中最多有 f 个恶意节点, 因此在每个更新周期每个节点通过检测操作最多删除 $2f$ 个邻居的值, 所以每个正常节点 i 在更新时的计算值 $x_i^{\alpha}[k]$ 变化仅受 $[\underline{x^{\gamma}}[k],\overline{x^{\gamma}}]$ 内的值影响. 因此根据式 (5), 正常节点 i 在 $k+1$ 时刻的计算值上限为

$$x_i^{\alpha}[k+1]\leqslant\left[\sum_{j\in\mathcal{J}_i[k]\backslash\mathcal{R}_i[k]\cup\{i\}}w_{ij}[k]\,x_j^{\alpha}[k]\right]\leqslant\left[\sum_{j\in\mathcal{J}_i[k]\backslash\mathcal{R}_i[k]\cup\{i\}}w_{ij}[k]\,\overline{x^{\gamma}}[k]\right]=\overline{x^{\gamma}}[k], \tag{16}$$

即 $x_i^{\alpha}[k+1]\leqslant\overline{x^{\gamma}}[k]$.

相似的正常节点 i 在 $k+1$ 时刻的计算值下限为

$$x_i^{\alpha}[k+1]\geqslant\left[\sum_{j\in\mathcal{J}_i[k]\backslash\mathcal{R}_i[k]\cup\{i\}}w_{ij}[k]\,x_j^{\alpha}[k]\right]\geqslant\left[\sum_{j\in\mathcal{J}_i[k]\backslash\mathcal{R}_i[k]\cup\{i\}}w_{ij}[k]\,\underline{x^{\gamma}}[k]\right]=\underline{x^{\gamma}}[k], \tag{17}$$

即 $x_i^{\alpha}[k+1]\geqslant\underline{x^{\gamma}}[k]$.

因此 $x_i^{\alpha}[k+1] \in [\underline{x^{\gamma}}[k], \overline{x^{\gamma}}[k]]$, 即正常节点 i 在 $k+1$ 时刻的计算值 $x_i^{\alpha}[k+1]$ 不小于 k 时刻的最小交互值 $\underline{x^{\gamma}}[k]$ 并且不大于 k 时刻的最大交互值 $\overline{x^{\gamma}}[k]$. 由式 (4) 得

$$x_i^{\gamma}[k+1] - x_i^{\beta}[k+1] \in [\underline{x^{\gamma}}[k], \overline{x^{\gamma}}[k]]. \tag{18}$$

进一步可得

$$\underline{x^{\gamma}}[k] + x_i^{\beta}[k+1] \leqslant x_i^{\gamma}[k+1] \leqslant \overline{x^{\gamma}}[k] + x_i^{\beta}[k+1]. \tag{19}$$

由于 $x_i^{\beta}[k] \geqslant 0$, 因此 $x_i^{\gamma}[k]$ 对于时刻 k 单调不减, 进一步可得 $x_i^{\alpha}[k]$ 对于时刻 k 单调不减.

由式 (6) 可知, 存在一个时刻 k' 使得当 $k \geqslant k'$ 时 $x_i'[k] = 0$, $x_i^{\beta}[k] = 0$. 由式 (4) 中 $x_i^{\gamma}[k] = x_i^{\alpha}[k] + x_i^{\beta}[k]$ 可得

$$x_i^{\gamma}[k] = x_i^{\alpha}[k] + x_i^{\beta}[k] = x_i^{\alpha}[k] + 0 = x_i^{\alpha}[k], \tag{20}$$

$$x_i[k] = x_i^{\alpha}[k] + x_i'[k] = x_i^{\alpha}[k] + 0 = x_i^{\alpha}[k]. \tag{21}$$

因此对于节点 i 的最终状态

$$\lim_{k\to\infty} x_i[k] = \lim_{k\to\infty} x_i^{\gamma}[k] = \lim_{k\to\infty} x_i^{\alpha}[k]. \tag{22}$$

由于 $x_i^{\alpha}[k+1] \in [\underline{x^{\gamma}}[k], \overline{x^{\gamma}}[k]]$, 因此

$$\underline{x^{\alpha}[k] + x^{\beta}[k]} \leqslant x_i^{\alpha}[k+1] \leqslant \overline{x^{\alpha}[k] + x^{\beta}[k]}. \tag{23}$$

进一步可得

$$\lim_{k\to\infty} \underline{x^{\alpha}[k] + x^{\beta}[k]} \leqslant \lim_{k\to\infty} x_i^{\alpha}[k+1] \leqslant \lim_{k\to\infty} \overline{x^{\alpha}[k] + x^{\beta}[k]}. \tag{24}$$

由于 $\lim_{k\to\infty} x_i^{\beta}[k] = 0$, 因此式 (24) 可改写为

$$\lim_{k\to\infty} \underline{x^{\alpha}[k]} \leqslant \lim_{k\to\infty} x_i^{\alpha}[k+1] \leqslant \lim_{k\to\infty} \overline{x^{\alpha}[k]}. \tag{25}$$

根据式 (22) 进一步可得

$$\lim_{k\to\infty} \underline{x^{\gamma}[k]} \leqslant \lim_{k\to\infty} x_i^{\alpha}[k+1] \leqslant \lim_{k\to\infty} \overline{x^{\gamma}[k]}. \tag{26}$$

根据式 (4), 当 $x_i^{\alpha}[0] = 0$ 时 $x_i^{\gamma}[0] = x_i^{\beta}[0]$. 根据 $x_i^{\alpha}[k+1] \in [\underline{x^{\gamma}}[k], \overline{x^{\gamma}}[k]]$ 得 $x_i^{\alpha}[1] \in [\underline{x^{\gamma}}[0], \overline{x^{\gamma}}[0]] = [\underline{x^{\beta}}[0], \overline{x^{\beta}}[0]]$, 因此

$$\begin{aligned} x_i^{\gamma}[1] = x_i^{\alpha}[1] + x_i^{\beta}[1] &\in \left[\underline{x^{\beta}}[0] + x_i^{\beta}[1], \overline{x^{\beta}}[0] + x_i^{\beta}[1]\right] \\ &= \left[\underline{x^{\beta}[0] + x^{\beta}[1]}, \overline{x^{\beta}[0] + x^{\beta}[1]}\right]. \end{aligned} \tag{27}$$

同理

$$\begin{aligned} x_i^{\gamma}[2] = x_i^{\alpha}[2] + x_i^{\beta}[2] &\in \left[\underline{x^{\beta}}[1] + x_i^{\beta}[2], \overline{x^{\beta}}[1] + x_i^{\beta}[2]\right] \\ &= \left[\underline{x^{\beta}[1] + x^{\beta}[2]}, \overline{x^{\beta}[1] + x^{\beta}[2]}\right]. \end{aligned} \tag{28}$$

进一步可得

$$\begin{aligned} x_i^{\gamma}[k] = x_i^{\alpha}[k-1] + x_i^{\beta}[k-1] &\in \left[\underline{x^{\beta}}[k-1] + x_i^{\beta}[k], \overline{x^{\beta}}[k-1] + x_i^{\beta}[k]\right] \\ &= \left[\underline{x^{\beta}[0] + x^{\beta}[1] + \cdots + x^{\beta}[k]}, \overline{x^{\beta}[0] + x^{\beta}[1] + \cdots + x^{\beta}[k]}\right], \end{aligned} \tag{29}$$

即

$$\lim_{k\to\infty}\underline{x^{\gamma}[k]}=\lim_{k\to\infty}\underline{x^{\beta}[0]+x^{\beta}[1]+\cdots+x^{\beta}[k]}=\lim_{k\to\infty}\underline{\sum_{T=0}^{T=k}x^{\beta}[T]},\tag{30}$$

$$\lim_{k\to\infty}\overline{x^{\gamma}[k]}=\lim_{k\to\infty}\overline{x^{\beta}[0]+x^{\beta}[1]+\cdots+x^{\beta}[k]}=\lim_{k\to\infty}\overline{\sum_{T=0}^{T=k}x^{\beta}[T]}.\tag{31}$$

因此可得

$$\lim_{k\to\infty}\underline{\sum_{T=0}^{T=k}x^{\beta}[T]}\leqslant\lim_{k\to\infty}x_i^{\alpha}[k+1]\leqslant\lim_{k\to\infty}\overline{\sum_{T=0}^{T=k}x_i^{\beta}[T]},\tag{32}$$

即

$$\underline{x'}[0]\leqslant\lim_{k\to\infty}x_i^{\alpha}[k+1]\leqslant\overline{x'}[0],\tag{33}$$

$$\underline{x}[0]\leqslant\lim_{k\to\infty}x_i^{\alpha}[k+1]\leqslant\overline{x}[0],\tag{34}$$

$$\underline{x}[0]\leqslant\lim_{k\to\infty}x_i[k+1]\leqslant\overline{x}[0].\tag{35}$$

因此证得 $\lim_{k\to\infty}x_i[k+1]\in[\underline{x}[0],\overline{x}[0]]$, 定义 4 中的安全性条件成立.

接下来证明定义 4 中的一致性条件. 根据式 (22) 且 $x_i^{\alpha}[k+1]\in[\underline{x^{\gamma}}[k],\overline{x^{\gamma}}[k]]$, 因此可得

$$\lim_{k\to\infty}x_i^{\alpha}[k+1]\in\left[\lim_{k\to\infty}\underline{x^{\gamma}}[k],\lim_{k\to\infty}\overline{x^{\gamma}}[k]\right]=\left[\lim_{k\to\infty}\underline{x_i^{\alpha}}[k],\lim_{k\to\infty}\overline{x_i^{\alpha}}[k]\right],\tag{36}$$

即当节点副本 $x'[k]$ 分配完毕并全部加入更新之后, 计算值 $\underline{x_i^{\alpha}}[k]$ 对于时刻 k 单调不减, $\overline{x_i^{\alpha}}[k]$ 对于时刻 k 单调不增, 交互值 $x_i^{\gamma}[k]$ 同理.

分别设 $\underline{x^*}[k]$ 与 $\overline{x^*}[k]$ 为 $\underline{x}[k]$ 与 $\overline{x}[k]$ 的最终状态值, 即 $\lim_{k\to\infty}\underline{x}[k]=\underline{x^*}[k]$, $\lim_{k\to\infty}\overline{x}[k]=\overline{x^*}[k]$. 用 $\mathcal{X}_1[k]$ 表示在时间 $k\geqslant k'$ 时包括恶意节点在内的所有交互值 $x_i^{\gamma}[k]$ 等于或大于 $\overline{x^*}$ 的节点的集合, 同样用 $\mathcal{X}_2[k]$ 表示所有交互值 $x_i^{\gamma}[k]$ 等于或小于 $\underline{x^*}$ 的节点集. 因此可得

$$\begin{aligned}\mathcal{X}_1[k]&=\left\{i\in\mathcal{V}:x_i^{\gamma}[k]\geqslant\overline{x^*}\right\},\\ \mathcal{X}_2[k]&=\left\{i\in\mathcal{V}:x_i^{\gamma}[k]\geqslant\underline{x^*}\right\}.\end{aligned}\tag{37}$$

假设两个集合非空且不相交. 由于网络拓扑满足 $(f+1,f+1)$- 鲁棒, 那么必须满足定义 4 中的条件之一, 特别地, 在 $\mathcal{X}_1[k]$ 或 $\mathcal{X}_2[k]$ 中总是存在一个正常节点 i 具有分别来自 $\mathcal{V}\backslash\mathcal{X}_1[k]$ 或 $\mathcal{V}\backslash\mathcal{X}_2[k]$ 的 $f+1$ 条边. 假设节点 i 处于 $\mathcal{X}_1[k]$ 中, 由于正常节点在更新过程中不会超过区间 $[\underline{x}[k],\overline{x}[k]]$, 因此 $x_i[k]=\overline{x^*}$, 即 $x_i^{\gamma}[k]=\overline{x^*}[k]$. 在算法检测隔离操作中, 最多删除 f 个来自 $\mathcal{V}\backslash\mathcal{X}_1[k]$ 中大于 $\overline{x^*}[k]$ 的交互值 $x_j^{\gamma}[k]$, $j\in\mathcal{V}\backslash\mathcal{X}_1[k]$. 因此其至少保留一个小于 $\overline{x^*}$ 的邻居的交互值用来更新子集的状态. 所以取得最大值 $\overline{x^*}$ 的正常节点将会在后续更新中减小. 同样, 若正常节点 i 处于 $\mathcal{X}_2[k]$ 中, 取得最小值 $\underline{x^*}$ 的正常节点将会在后续更新中增大. 因此存在一个有限时间 k_c 使得当 $k\geqslant k_c$ 时, 节点更新至 $\underline{x^*}=\overline{x^*}$, 此时 $\underline{x^*}=x_i^{\gamma}[k]=\overline{x^*},i\in\mathcal{N}$. 由此可得 $\underline{x^*}=x_i[k]=\overline{x^*},i\in\mathcal{N}$, 证得 $x_1[k]=\cdots=x_i[k]$, 定义 4 中一致性条件成立.

推论1 若网络中不存在恶意节点, 当且仅当底层图具有生成树时, 使用算法 1 的多智能体系统能够以概率 1 达成弹性一致性.

证明 在 $f=0$ 的情况下, 由定理 3 可知, 在正常网络中达成共识的充分必要条件是底层图满足 $(1,1)$- 鲁棒图. 根据引理 1, 当图具有一个生成树时, 图是 1- 鲁棒图. 因此证得, 当图具有生成树时, 使用算法 1 的多智能体系统能够达成弹性一致性.

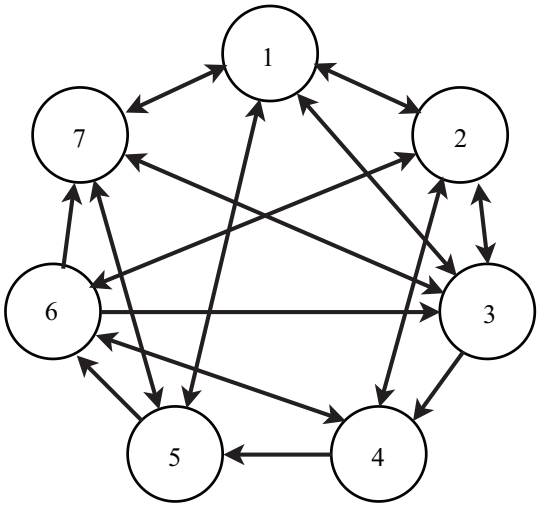

图 2　7 个节点组成的通信拓扑图

Figure 2　Communication topology with 7 nodes

4　数值仿真与分析

本节将通过数值仿真实验来对文中提出算法的有效性进行验证, 实验仿真采用了与文献 [27] 相同的实验设置, 同时和传统的一致性算法以及 W-MSR 算法进行了性能比较, 通过进行相同的模拟数值仿真实验来验证本文算法的有效性及特性.

考虑一个由 7 个节点组成的多智能体系统分布式网络, 其抽象网络通信拓扑如图 2 所示, 其满足 $(2,2)$- 鲁棒. 网络中每个节点的初始状态值分别指定为 $x_1[0]=1, x_2[0]=2, x_3[0]=3, x_4[0]=4, x_5[0]=5, x_6[0]=6, x_7[0]=7$. 假设网络中恶意节点数量上限 $f=1$, 恶意节点身份为节点 7, 其在网络更新中执行欺骗攻击, 对于每个时刻 k 始终保持自身状态值不变, 即 $x_7[0]=\cdots=x_7[k]=7, k\in\mathbb{Z}_+$. 网络中每个节点在 $k=0$ 时刻开始执行本文提出的算法 1, 本文设置参数 $c=1/4, \eta=1/5$.

采用 W-MSR 算法与算法 1 的系统收敛情况如图 3(a) 与 (b) 所示. 从图 3(a) 中可以看到, 采用 W-MSR 算法的系统能够有效地降低恶意节点的影响并快速达成一致性收敛. 从图 3(b) 中可以看到, 采用算法 1 的系统也能够有效地达成一致性收敛, 并且拥有不弱于 W-MSR 算法的抗攻击能力. 此外, 通过观察图 3(a) 与 (b) 的最终收敛值可以发现, 对于系统的期望收敛值 4, 算法 1 的收敛值 4.1 相对于 W-MSR 算法的收敛值 3.8 误差更小. 这是由于算法 1 在移除异常值操作时, 对每一个可疑值进行了检测判断. 在每个时刻, 每个节点根据接收列表中的值计算一个安全区间 $\mathcal{F}_{\mathrm{Sec}}[k]$, 只有差值严格大于 $\mathcal{F}_{\mathrm{Sec}}[k]$ 的极值才会被删除, 减少了正常信息值的浪费. 因此当恶意值位于列表中的非极值位置时, 即节点接收到的极值信息都是正常值时, 该时刻会大概率对其进行保留而非移除, 避免造成正常信息的浪费; 其次, 算法 1 限制了节点的更新方式, 时变地控制每个节点状态值的变化范围, 使正常节点状态值的差始终处于一个较小区间之内, 随着时间更新, 该区间会不断缩小直至为 0, 因此恶意值将更加明显地区别于正常值, 从而更易被移除. 但另一方面, 可以发现算法 1 收敛速度要慢于 W-MSR 算法, 这是由状态分配机制带来的性能影响, 即算法 1 通过牺牲一定的时间来保证系统中节点的隐私以及安全性. 从状态分配机制中可以看到, 算法限制节点在每个时刻的状态值变化, 使节点状态值在相邻时刻内的变化始终位于一定的安全区间内, 通过将正常更新的变化范围划定为一个个独立的安全区间从而使节点具备了检测异常信息的能力, 即正常节点具有相同的特征而不遵循正常更新的恶意节点明显

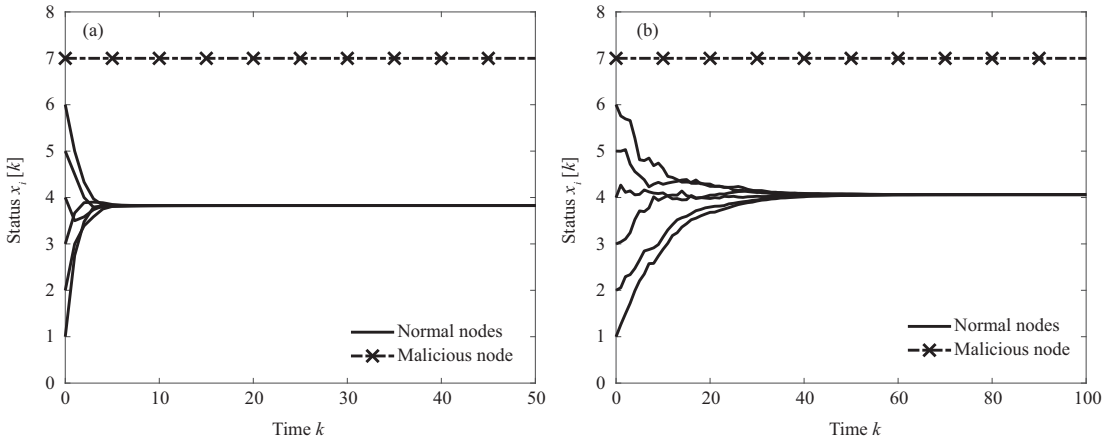

图 3 W-MSR 算法与算法 1 抗攻击能力对比

Figure 3 Comparison of attack resistance between (a) W-MSR algorithm and (b) Algorithm 1

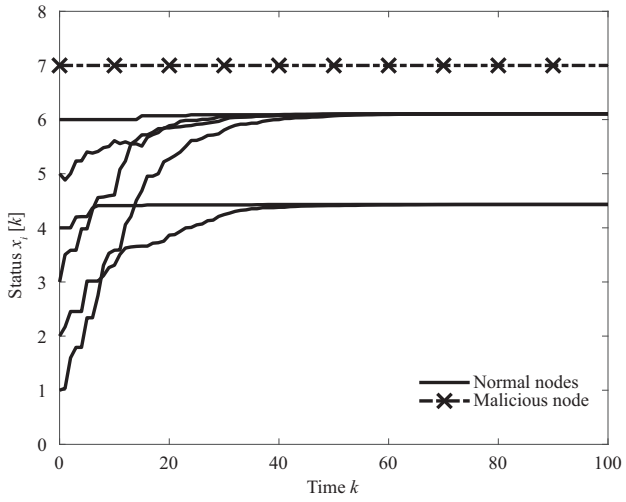

图 4 通信拓扑为 (2, 1)- 鲁棒时算法 1 收敛情况

Figure 4 System consensus with Algorithm 1 when the communication topology is (2, 1)-robust

区别于正常节点, 更容易被正常节点检测到从而移除其恶意信息, 这也证实了算法 1 能够有效地减少正常信息的浪费. 此外, 根据弹性一致性的定义可知, 算法 1 的收敛时间仍处于合理范围内, 因此算法的收敛速度是可接受的. 通过观察图 3(b) 中前 20 个时刻可以发现算法 1 与传统一致性算法相比节点收敛轨迹较为曲折, 并且与 W-MSR 算法相比每个节点在相邻时刻状态值的变化是有限的, 这说明了应用算法 1 的多智能体系统对节点行为进行了限制, 控制节点在一定时刻内状态值的变化, 保证了正常节点间的区别始终维持在一个较小的范围之内, 所有恶意节点的行为都将明显区别于正常节点, 保证正常节点能够将其识别出来并避免受到影响, 从而保证节点能够达成弹性一致性.

由于网络中恶意节点数量上限 $f = 1$, 根据定理 3 可知网络拓扑需满足 $(2,2)$- 鲁棒, 因此采用图 2 通信拓扑的多智能体系统可以实现收敛. 我们在通信拓扑为 $(2,1)$- 鲁棒的多智能体系统中进行了同样的数值仿真, 其收敛情况如图 4 所示.

如图 4 所示, 由于网络鲁棒性不满足定理 3 中的鲁棒性要求, 网络中正常节点收敛到了两个值,

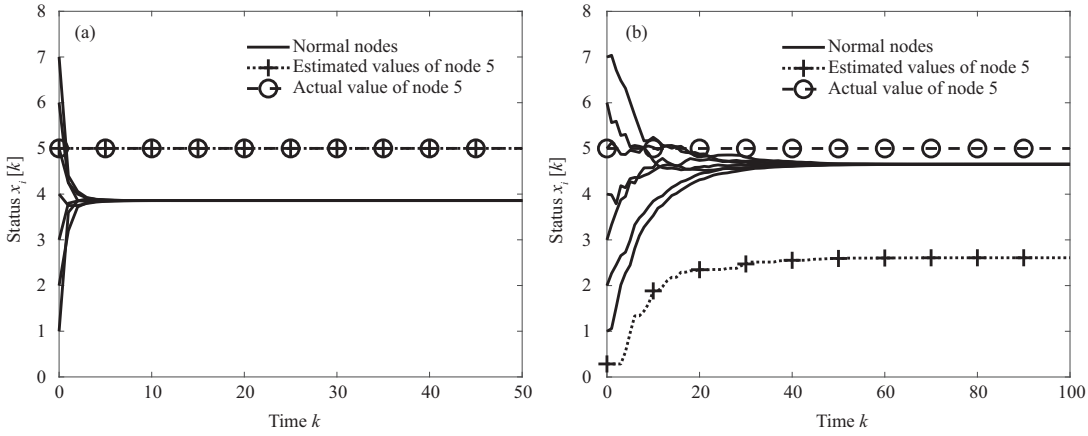

图 5 W-MSR 算法与算法 1 隐私保护能力对比

Figure 5 Comparison of privacy-preserving capabilities between (a) W-MSR algorithm and (b) Algorithm 1

一部分正常由于未与恶意节点 7 直接相邻, 因此其受恶意节点影响相对较小. 而另一部分直接与恶意节点 7 直接相连, 并且由于通信拓扑不满足 $(2,2)$- 鲁棒, 使得节点在更新时无法正常地删除极端值以及保留正常值, 导致节点收敛走向严重受恶意节点影响. 通过对比图 3(b) 与图 4, 证明了若网络拓扑不满足定理 3 中的鲁棒性要求, 则应用算法 1 的多智能体系统无法达成收敛.

接下来进行隐私保护分析, 在进行数值模拟仿真之前, 先进行以下设计.

本文参考文献 [28], 在仿真实验中建立了一个观测器作为外部窃听者来推测节点的真实初始状态. 根据假设 1, 外部窃听者能力有限, 因此设计窃听者每一时刻有 50% 的概率决定当前时刻是否进行窃听, 连续窃听的概率为 80%, 即当前时刻进行窃听后, 下一时刻有 80% 的概率决定继续窃听. 由于初始传输周期的重要性, 设计窃听者在前 10 个时刻中一定进行窃听行为, 之后窃听者最多连续进行 5 次窃听后至少停止一个时刻为下次窃听进行准备. 窃听者的更新规则如下:

(1) 窃听者的初始状态为 0, 即

$$z[0]=0. \tag{38}$$

(2) 窃听者的更新是基于传输状态值与预测状态值之间的累积的, 即

$$z[k+1]=z[k]+\tilde{x}_i[k+1]-\sum_{j\in N_i\cup i} w_i[k]\,\tilde{x}_j[k], \tag{39}$$

其中 $\tilde{x}_i[k]$ 表示窃听者在 k 时刻截获到的节点 i 传输的状态值. 选择被窃听节点的身份为节点 5.

图 5(a) 显示了采用 W-MSR 算法的系统中不存在恶意节点时, 外部窃听者推测节点 5 真实初始状态值的仿真图像. 在相同情况下, 对采用算法 1 的系统进行同样的实验, 如图 5(b) 所示.

在不存在恶意节点攻击时, 使用算法 1 的系统对系统中节点的隐私保护能力如图 5(b) 所示. 节点 5 的真实初始状态值为 5, 而窃听者预测节点 5 的状态值约为 2.6, 偏离了其真实的初始状态值, 说明窃听者无法准确估计节点 5 的初始状态, 显示了应用算法 1 的多智能体系统隐私保护能力. 从图 5(a) 中可以看到, 在采用 W-MSR 算法的系统中, 外部窃听者能够准确推断出节点 5 的初始状态值. 这是因为 W-MSR 算法在第一个时刻就将节点真实初始状态值向外传输出去, 并且 W-MSR 算法使得系统在很短的周期内快速实现收敛, 使得外部窃听者能够以较小的代价准确推测出节点的初始状态. 通过对比图 5(a) 与 (b) 说明了算法 1 相较于 W-MSR 算法具有较强的隐私保护效果, 体现了本文所提算法的贡献.

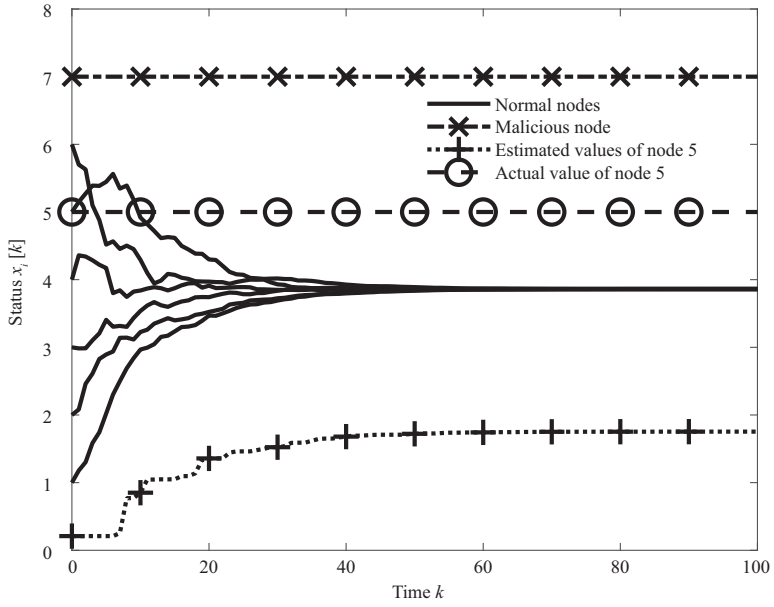

图 6 存在攻击下采用算法 1 时系统隐私保护效果

Figure 6 Privacy-preserving effect with Algorithm 1 under attacks

表 1 算法 1 与 MSR 算法及 W-MSR 算法性能比较

Table 1 Comparison of Algorithm 1 with MSR algorithm and W-MSR algorithm

	MSR algorithm [10]	W-MSR algorithm [26]	Algorithm 1
Privacy-preserving	No	No	Yes
Detection ability	No	No	Yes

接下来我们检验应用算法 1 的系统在存在恶意攻击时系统的隐私保护能力. 通过结合上文中抗攻击仿真以及隐私保护仿真中的相关设置, 图 6 显示了应用算法 1 的系统性能.

如图 6 所示, 窃听者对于节点 5 的状态值推测约为 1.8, 说明系统在能够达成一致性收敛的同时有效保护了节点隐私, 即使系统中存在恶意节点对系统进行干扰时, 算法 1 依然具备隐私保护能力. 此外可以看到此次窃听者对节点 5 的推测值与上文中的有所区别, 这是由于随机性的加入使得节点每一时刻向外传输的状态值都是随机的, 并且由于恶意节点的存在, 使得正常节点在更新时会对邻居传来的值进行选择取舍, 由此导致了节点更新的变化, 使得窃听者累积的窃听值发生变化.

通过以上实验对比可以看出算法 1 相对于 MSR 算法与 W-MSR 算法具有新颖的隐私保护机制, 在具备优秀抗攻击能力的同时能够较好保护系统中各节点的初始状态隐私. 此外, 算法性能进一步优化, 在弹性一致性常见的移除值操作中加入了检测环节, 对正常信息的保留程度更强, 减少了正常资源信息的浪费. 表 1 显示了算法 1 相比于 MSR 算法与 W-MSR 算法的改进.

5 结语

本文针对实际环境中多智能体系统面临的网络攻击与隐私窃取问题, 提出了一种基于状态分配机制的具备隐私保护能力的多智能体弹性一致性控制算法. 通过精心设计的状态分配机制加大了窃听者

推测节点真实信息的难度, 从而有效保护系统内节点隐私. 此外在此基础上加入检测环节, 使算法识别恶意信息能力更强并减少了信息资源的浪费. 本文通过数学理论分析证明了算法的隐私保护性能与抗攻击性能, 最后通过数值仿真实验验证了本文算法的有效性.

同时, 本文提出的一致性控制算法仍存在一定的不足. 第一, 算法通过牺牲部分收敛时间来换取隐私保护能力, 在一定程度上增加了通信成本. 第二, 所考虑窃听者能力受限, 对于具有较强能力的窃听者, 隐私保护程度将降低, 因此针对强力窃听者的隐私保护手段值得作为未来的研究工作.

参考文献

1 Singh V P, Kishor N, Samuel P. Distributed multi-agent system-based load frequency control for multi-area power system in smart grid. IEEE Trans Ind Electron, 2017, 64: 5151–5160

2 Usevitch J, Panagou D. Resilient trajectory propagation in multirobot networks. IEEE Trans Robot, 2022, 38: 42–56

3 Safi M, Dibaji S M, Pirani M. Resilient coordinated movement of connected autonomous vehicles. Eur J Control, 2022, 64: 100613

4 Zhai Y, Liu Z W, Guan Z H, et al. Resilient consensus of multi-agent systems with switching topologies: a trusted-region-based sliding-window weighted approach. IEEE Trans Circuits Syst II, 2021, 68: 2448–2452

5 Fax J A, Murray R M. Information flow and cooperative control of vehicle formations. IEEE Trans Automat Contr, 2004, 49: 1465–1476

6 Ge X, Han Q L, Zhong M, et al. Distributed Krein space-based attack detection over sensor networks under deception attacks. Automatica, 2019, 109: 108557

7 Zhao Y, Liu Y, Wen G, et al. Distributed average tracking for Lipschitz-type of nonlinear dynamical systems. IEEE Trans Cybern, 2018, 49: 4140–4152

8 Wen G, Zheng W X. On constructing multiple Lyapunov functions for tracking control of multiple agents with switching topologies. IEEE Trans Automat Contr, 2018, 64: 3796–3803

9 Antoniadis K, Benhaim J, Desjardins A, et al. Leaderless consensus. J Parallel Distr Com, 2023, 176: 95–113

10 Kieckhafer R M, Azadmanesh M H. Reaching approximate agreement with mixed-mode faults. IEEE Trans Parallel Distrib Syst, 1994, 5: 53–63

11 Proskurnikov A V, Matveev A S, Cao M. Opinion dynamics in social networks with hostile camps: consensus vs. polarization. IEEE Trans Automat Contr, 2015, 61: 1524–1536

12 Fang X, Misra S, Xue G, et al. Smart grid — the new and improved power grid: a survey. IEEE Commun Surv Tutorials, 2011, 14: 944–980

13 Hendriks R C, Erkin Z, Gerkmann T. Privacy preserving distributed beamforming based on homomorphic encryption. In: Proceedings of the 21st European Signal Processing Conference (EUSIPCO 2013), 2013. 1–5

14 Hendriks R C, Erkin Z, Gerkmann T. Privacy-preserving distributed speech enhancement forwireless sensor networks by processing in the encrypted domain. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2013. 7005–7009

15 Li Q, Christensen M G. A privacy-preserving asynchronous averaging algorithm based on Shamir's secret sharing. In: Proceedings of the 27th European Signal Processing Conference (EUSIPCO), 2019. 1–5

16 Katewa V, Pasqualetti F, Gupta V. On privacy vs. cooperation in multi-agent systems. Int J Control, 2018, 91: 1693–1707

17 Mo Y, Murray R M. Privacy preserving average consensus. IEEE Trans Automat Contr, 2016, 62: 753–765

18 Kefayati M, Talebi M S, Khalaj B H, et al. Secure consensus averaging in sensor networks using random offsets. In: Proceedings of IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, 2007. 556–560

19 Gao H, Zhang C, Ahmad M, et al. Privacy-preserving average consensus on directed graphs using push-sum. In: Proceedings of IEEE Conference on Communications and Network Security (CNS), 2018. 1–9

20 Charalambous T, Manitara N E, Hadjicostis C N. Privacy-preserving average consensus over digraphs in the presence of time delays. In: Proceedings of the 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2019. 238–245

21 Wang Y. Privacy-preserving average consensus via state decomposition. IEEE Trans Automat Contr, 2019, 64: 4711–4716
22 Duan X, He J, Cheng P, et al. Privacy preserving maximum consensus. In: Proceedings of the 54th IEEE Conference on Decision and Control (CDC), 2015. 4517–4522
23 Zheng W, He Z, He J, et al. Accurate resilient average consensus via detection and compensation. In: Proceedings of the 60th IEEE Conference on Decision and Control (CDC), 2021. 5502–5507
24 Dolev D. The Byzantine generals strike again. J Algorithms, 1982, 3: 14–30
25 Lamport L, Shostak R, Pease M. The Byzantine generals problem. In: Concurrency: the Works of Leslie Lamport. New York: Association for Computing Machinery, 2019. 203–226
26 LeBlanc H J, Zhang H, Koutsoukos X, et al. Resilient asymptotic consensus in robust networks. IEEE J Sel Areas Commun, 2013, 31: 766–781
27 Nakamura M, Ishii H, Dibaji S M. Resiliency against malicious agents in maximum-based consensus. SICE J Control Meas Syst Integr, 2021, 14: 279–290
28 Ruan M, Gao H, Wang Y. Secure and privacy-preserving consensus. IEEE Trans Automat Contr, 2019, 64: 4035–4049

Resilient consensus control of multi-agent systems under privacy protection

Chong ZHANG, Yiming WU*, Ming XU & Ning ZHENG

School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018*, China*
* Corresponding author. E-mail: ymwu@hdu.edu.cn

Abstract This paper presents a new privacy-preserving resilient consensus algorithm for multi-agent systems whose communication network suffers from deception attacks and privacy theft. First, considering the multi-agent system directly exposes the real state information to the outside in the information transmission process, a node state information processing mechanism based on state allocation is designed to ensure the security of node state information privacy in the system. Secondly, considering the influence of external attackers on system consensus, the detection step is further introduced into the state allocation mechanism to ensure the safe consensus of the system. Third, mathematical theory analysis proved that the algorithm can effectively protect the privacy of node initial state information and ensure the system achieves resilient consensus. Finally, the effectiveness of this algorithm is further verified by numerical simulation and comparison experiments.

Keywords multi-agent systems, resilient consensus, deception attack, privacy-preserving, cyber security