面向工业互联网中数据不透明性的 隐私保护控制策略设计

沈逸¹, 周纯杰¹, 胡晓娅¹, 程骋¹, 何心², 邵若晨^{3*}

1. 华中科技大学人工智能与自动化学院, 武汉 430074;

2. 华中科技大学机械科学与工程学院, 武汉 430074;

3. 南洋理工大学电气与电子工程学院, 新加坡 639798, 新加坡

* E-mail: ruochen001@e.ntu.edu.sg

收稿日期: 2021-08-27; 接受日期: 2021-11-29; 网络版发表日期: 2021-12-21

国家重点研发计划(编号: 2018YFB1701202)和国家自然科学基金青年基金(批准号: 51905197)资助项目

摘要 随着信息通讯技术(information and communication technologies, ICT)的发展及其在工业环境中的应用, 工业互联网应运而生. 相比于传统工业相对封闭安全的生产环境, 工业互联网面临着严峻的数据隐私安全挑战. 为了提供能够抵御网络威胁的鲁棒防御策略, 本文研究了面向工业互联网中数据不透明性的隐私保护控制问题. 在此问题中, 受控系统被建模为网络化离散事件系统(discrete-event system, DES)并且系统中存在着可以利用观测数据来推测系统秘密信息的恶意窃听者. 本文的目标是通过设计网络化监督器从而实现以下两个目标: (1) 窃听者无法利用观测数据推测出系统秘密信息, 即不透明性强制; (2) 在网络化监督器的控制下, 系统能实现给定的控制目标. 为了解决此隐私保护控制问题, 首先, 本文提出了一套针对网络化离散事件系统的隐私保护控制架构; 其次, 本文将系统中各子模块建模为有限状态自动机以模拟其动力学; 最后, 基于构建的状态机模型, 通过将原隐私保护控制问题转化为经典的Ramadge-Wonham监督控制问题, 本文设计了网络化监督器合成算法. 进一步地, 本文证明了所设计合成算法的正确性与完备性以及最大网络化监督器的存在性.

关键词 工业互联网, 网络化离散事件系统, 隐私保护控制, 网络化监督器, 有限状态自动机, 不透明性

1 引言

随着5G网络、云平台、大数据等新兴技术的高速发展, 全球互联网平台实现了高速增长. 针对工业应用场景, 工业互联网平台通过建立人(使用者)-机(各类机械设备)-物(业务系统)的互联, 实现了数据跨系统的传输, 形成了基于混合系统建模、模式分析、以及数

据驱动的新生产模式^[1-4]. 但同时这些高速发展的信息技术也带来了相应的隐患: 网络应用的普遍化与信息系统的规模化和去中心化(大数据及云技术)使得数据量成指数级增长, 而海量的数据中夹杂着大量的隐私数据, 存储这些数据(特别是敏感隐私数据)的企业可能会成为网络攻击者的主要目标, 这便使得恶意攻击、病毒泛滥等随之而来的网络安全问题愈加凸显,

引用格式: 沈逸, 周纯杰, 胡晓娅, 等. 面向工业互联网中数据不透明性的隐私保护控制策略设计. 中国科学: 技术科学, 2022, 52: 152-164
Shen Y, Zhou C J, Hu X Y, et al. Design of privacy-preserving control strategy for data opacity of industrial internet (in Chinese). Sci Sin Tech, 2022, 52: 152-164, doi: 10.1360/SST-2021-0407

甚至会威胁到人民生命财产的安全. 因此, 为了保障网络信息系统的安全并提供健康的网络生态环境, 研究隐私保护问题刻不容缓且具有重要的现实意义. 在本文的隐私保护研究中, 我们将着重探讨一类攻击者, 即通过窃听数据以推测系统秘密的窃听者. 为了抵御此类窃听者并提高系统安全性能, 不透明性(opacity)的概念应运而生并首次于计算机科学领域中被提出, 用于分析密码协议^[5].

作为具有离散状态空间和事件触动力学的一类重要的信息物理系统, 离散事件系统(discrete-event systems, DES)可以准确地描述工业互联网中子模块(例如供应链设备、传输设备、加工仪器等)的工作模式, 因此, 基于离散事件系统的智能制造控制和优化算法受到了越来越多的关注. 本文我们将着重于研究面向离散事件系统的隐私保护控制问题. 在离散事件系统中, 不透明性是一种基于语言的用于描述系统安全性的属性. 不透明性的参数^[6]包括: 系统运行行为的子集和一个定义为从系统运行行为到一组观测值的观测映射. 基于此观测映射, 如果外界窃听者无法通过观测数据推断出系统是否达到秘密状态, 则称系统是不透明的. 不透明性一般包括两类: 基于状态的不透明性^[7,8]和基于语言的不透明性^[9,10]. 对于基于状态的不透明性, 有五类衍生出的不透明性概念是最常被研究的: 当前状态不透明性^[7,8]; 初始状态不透明性^[7]; 初始与最终状态不透明性^[11]; K步不透明性^[8]; 无穷步不透明性^[12].

对于不透明性的研究, 前人工作主要包括其验证和强制. 不透明性的验证侧重于检验给定系统是否满足不透明性的条件, 而不透明性的强制则侧重于设计隐私保护策略使得给定系统满足不透明性的条件. 本文将着重研究面向离散事件系统的不透明性强制问题. 对于不透明性强制, 通常有三类策略:

(1) 监督控制^[13~19], 即设计监督器来限制系统的行为以满足当前状态不透明性^[18]、初始状态不透明性^[19]、以及无穷步不透明性^[16]. 在设计强制不透明性的监督器的过程中, 为了减小监督器合成算法的计算复杂度, 转换系统的交替模拟关系可以被用于基于抽象的控制器合成过程中^[14];

(2) 设计编辑函数^[20~30], 即改变传感器数据以欺骗外界窃听者从而保证系统秘密状态无法被窃听者推测出. 插入函数是一种常用的隐私保护模型, 通过在特定

观测事件序列后加入额外事件信息以欺骗窃听者, 插入函数可实现当前状态不透明性强制和基于语言的不透明性强制^[22]. 进一步地, 当假设窃听者有或无关于插入函数结构的先验知识时, 强制当前状态不透明性的问题已得以解决^[24]. 除此之外, 删除函数也是一种常用的强制当前状态不透明性的隐私保护模型^[25];

(3) 调整传感器观测能力^[31~33], 即有选择地打开或关闭部分传感器, 从而影响外界窃听者的可观测数据以防止系统秘密的泄露. 离线或在线调整传感器观测能力^[32]是两种常用的不透明性强制方法, 基于此策略, 系统的可观测输出会被限制于不会暴露系统秘密的语言中, 从而强制当前状态不透明性. 无限步不透明性作为一种更强的隐私安全属性, 也可通过合成在线调整传感器观测能力的策略得以强制^[33]. 在确保基于语言不透明性的前提下, 最大信息泄露问题可通过调整传感器观测能力得以解决^[33].

基于以上文献介绍, 尽管前人针对离散事件系统中的不透明性强制问题开展了大量的研究工作, 但面向网络化离散事件系统强制不透明性的工作尚未展开. 一方面, 确保数据的不透明性为工业互联系统的平稳运行与安全防护提供了保障; 另一方面, 在工业互联网基于云平台实现海量数据的接入、存储、分析和服务体系中^[1], 时延不可避免地发生于通讯网络中, 这会导致网络化离散事件系统中的信息流可能不再满足先进先出(first-in-first-out, FIFO)的原则, 从而造成监督器无法根据接收到的观测事件序列进行准确的系统状态估计, 这便极大地增加了在有网络时延情况下设计监督器的难度. 而在前人关于不透明性强制的工作中, 由于其只考虑了理想情况下的不透明性强制问题, 即网络时延并未被纳入闭环监督器设计中考虑, 而基于以上分析, 我们已知时延对于系统的鲁棒监督器设计具有重要影响, 因此, 针对非网络化离散事件系统而设计的强制不透明性的监督器将未必再适用于网络化离散事件系统的不透明性强制. 在本文中, 为了提高隐私保护控制的实用性, 我们首次研究了面向网络化离散事件系统强制不透明性的隐私保护控制问题, 特别地, 我们将通过设计监督器实现隐私保护与闭环控制两个目标, 即设计面向网络化离散事件系统强制不透明性的监督器同时保证系统仍能够在监督器控制下实现给定控制目标. 此研究课题由于将通讯网络时延纳入系统建模中考虑, 不仅更加符合实际工业系统设置,

而且也与工业互联网系统所需求的隐私保护控制初衷相吻合. 本文的主要贡献包括以下三点:

(1) 通过解决网络化监督器的合成问题, 本文实现了网络化离散事件系统的隐私保护控制. 据我们所知, 本文是第一篇研究网络化离散事件系统不透明性强制问题的研究. 本文建立了一套网络化离散事件系统的隐私保护控制框架, 该框架中的观测信道与控制信道无需满足FIFO原则的限制, 相比于前人工作中研究的基于非网络化离散事件系统下的隐私保护, 本文提出的控制框架由于考虑了时延, 更符合实际工业场景;

(2) 通过将网络化离散事件系统中各模块的模型建立为有限状态自动机, 在本文提出的控制框架中, 合成用于强制不透明性的网络化监督器问题可以被转化为经典的Ramadge-Wonham监督控制问题. 基于此建模方法, 本文所提出的合成算法具有较高的泛化性, 即无论针对FIFO信息流或non-FIFO信息流, 该合成算法都将有效;

(3) 证明了所提算法的正确性与完备性. 进一步地, 基于本文构造的模型, 证明了最大的网络化监督器的存在且可以通过基于normality的合成方法将其合成.

本文结构如下: 第2章介绍关于形式化语言与自动机的基本知识; 第3章介绍网络化离散事件系统中各子模块的模型构建; 第4章介绍本文提出的合成用于强制不透明性的网络化监督器的算法; 第5章针对工业互联网环境下供应链中的物料调度场景进行仿真并展示所提算法的有效性; 第6章对全文进行总结, 并探讨更多实现隐私保护控制的可行策略.

2 预备知识

在本章中, 我们将介绍关于形式化语言与有限状态自动机的一些基本概念^[34-36].

给定一个有限集合 Σ , 令 Σ^* 表示其自由幺半群, 其中, ε 为单位元素; 字符串连接为幺半群操作. 对于一个字符串 s , $|s|$ 定义为其长度. 给定两个字符串 $s, t \in \Sigma^*$, 我们称 s 是 t 的一个前缀子字符串, 记作 $s \leq t$, 如果存在 $u \in \Sigma^*$ 使得 $su=t$ (su 表示 s 与 u 的连接). 一组语言 $L \subseteq \Sigma^*$ 是一个包含字符串的集合. L 的前缀闭包定义为 $\bar{L} = \{u \in \Sigma^* \mid (\exists v \in L) u \leq v\}$. 在本文中, 集合 Σ 被分为 $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc} = \Sigma_o \dot{\cup} \Sigma_{uo}$. 集合 $\Sigma_c(\Sigma_o)$ 和 $\Sigma_{uc}(\Sigma_{uo})$ 定义为

可控事件集合(可观测事件集合)和不可控事件集合(不可观测事件集合). $P_o: \Sigma^* \rightarrow \Sigma_o^*$ 是一个自然投影, 定义如下:

$$\begin{aligned} P_o(\varepsilon) &= \varepsilon, \\ (\forall \sigma \in \Sigma) P_o(\sigma) &= \begin{cases} \sigma, & \sigma \in \Sigma_o, \\ \varepsilon, & \text{其他}, \end{cases} \\ (\forall s \in \Sigma^*, \sigma \in \Sigma) P_o(s\sigma) &= P_o(s)P_o(\sigma). \end{aligned} \quad (1)$$

一个定义在 Σ 上的有限状态自动机 G 通常表示为一个五元组 $G = \{Q, \Sigma, \xi, q_0, Q_m\}$. 在 G 中, Q 为状态集合, $\xi: Q \times \Sigma \rightarrow Q$ 为转移函数, $q_0 \in Q$ 为初始状态, $Q_m \subseteq Q$ 为标记状态集. 如果 $\xi(q, \sigma)$ 有定义, 我们将其记作 $\xi(q, \sigma)!$. 我们也将 $\xi \subseteq Q \times \Sigma \times Q$ 视作一个关系. 在 G 中, 状态 q 处所使能的事件集合定义为 $En_G(q) = \{\sigma \in \Sigma \mid \xi(q, \sigma)!\}$. $L(G)$ 与 $L_m(G)$ 表示 G 的闭合行为与标记行为^[34]. 当 $Q_m = Q$ 时, 将 G 写作 $G = (Q, \Sigma, \xi, q_0)$. 给定事件集合 $\Sigma' \subseteq \Sigma$, 状态 $q \in Q$ 的“不可观测范围”定义为 $UR_{G, \Sigma - \Sigma'}(q) = \{q' \in Q \mid [\exists s \in (\Sigma - \Sigma')^*] q' = \xi(q, s)\}$. 对于 G , 将其子集构建 $P_{\Sigma'}(G)$ 定义为一个有限状态自动机 $(2^Q, \Sigma, \delta, UR_{G, \Sigma - \Sigma'}(q_0))$. $P_{\Sigma'}(G)$ 的转移函数 $\delta: 2^Q \times \Sigma \rightarrow 2^Q$ 定义如下:

(1) 对于任意的 $\emptyset \neq Q' \subseteq Q$ 和任意的 $\sigma \in \Sigma'$, $\delta(Q', \sigma) = UR_{G, \Sigma - \Sigma'}(\xi(Q', \sigma))$;

(2) 对于任意的 $\emptyset \neq Q' \subseteq Q$ 和任意的 $\sigma \in \Sigma - \Sigma'$, $\delta(Q', \sigma) = Q'$.

如果 G 中的每一个可达状态都可以到达 Q_m 中的一个标记状态, 我们称一个有限状态自动机 $G = (Q, \Sigma, \xi, q_0, Q_m)$ 是非阻塞的. 对于任意两个有限状态自动机 $G_1 = (Q_1, \Sigma_1, \xi_1, q_{1,0}, Q_{1,m})$ 和 $G_2 = (Q_2, \Sigma_2, \xi_2, q_{2,0}, Q_{2,m})$ 有: G_1 与 G_2 的同步积^[35]记作 $G_1 \parallel G_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \zeta, (q_{1,0}, q_{2,0}), Q_{1,m} \times Q_{2,m})$, 并且转移函数 ζ 定义如下:

$$\begin{aligned} \zeta((q_1, q_2), \sigma) &:= \\ &\begin{cases} (\xi_1(q_1, \sigma), \xi_2(q_2, \sigma)), & \sigma \in En_{G_1}(q_1) \cap En_{G_2}(q_2), \\ (\xi_1(q_1, \sigma), q_2), & \sigma \in En_{G_1}(q_1) \setminus \Sigma_2, \\ (q_1, \xi_2(q_2, \sigma)), & \sigma \in En_{G_2}(q_2) \setminus \Sigma_1. \end{cases} \end{aligned} \quad (2)$$

在本文中, $\Gamma = 2^{\Sigma_c} - \{\emptyset\}$ 表示所有控制命令的集

合, 即每个控制命令只包含使能的可控事件. $\mathbb{N}$ 表示非负整数集合, $\mathbb{N}^+$ 表示正整数集合. 对于任意的 $m \in \mathbb{N}$ 和任意的 $n \in \mathbb{N}$, 令 $[m:n] = \{m, m+1, \dots, n\}$, $(m:n) = \{m+1, m+2, \dots, n-1\}$. 当 $m > n$ 时, $[m:n] = \emptyset$; 当 $m+1 > n-1$ 时, $(m:n) = \emptyset$. $tick$ 表示单位时间^[37]. $\mathbb{N}^S$ 表示所有从 S 中生成的多重集合. 为了避免无穷的情形, 我们将给 $\mathbb{N}^S$ 加上一个约束参数, 记作 $\mathbb{N}^S(l)$ (l 表示多重集合的基数上限).

3 网络化离散事件系统模型

本章节我们将介绍网络化离散事件系统中的模型构建. 当存在窃听者(窃听系统秘密)的情况下, 网络化离散事件系统的隐私保护控制架构如图1所示, 其中包含六个子模块: 系统 G , 命令执行模块 CE , 观测信道 OC , 网络化监督器 NS , 控制信道 CC , 窃听者 I . 窃听者 I 是一个(被动)外部观测者, 其目的是通过观测到的传感器事件推测系统是否已经到达秘密状态. 我们假设窃听者 I 只能观测到系统 G 执行的事件而不能观测到网络化监督器 NS 发出的控制命令.

3.1 系统 G

本文将系统 G 建模为离散事件系统, 记作 $G = (Q, \Sigma, \xi_G, q^{init}, Q_m)$. 我们需要设计监督器使得: G 能够到达 Q_m 中的状态; 隐私保护控制能够实现. 因此, 将系统 G 的秘密状态集合记作 $Q_{sec} \subseteq Q$. 同时, 只考虑当前状态不透明性(current-state opacity, CSO), 表示窃听者无法通过观测事件数据推测系统当前状态是否已到达秘密状态, 其定义如下.

定义 3.1 (当前状态不透明性^[8]) 给定映射 P , $G = (Q, \Sigma, \xi_G, q^{init}, Q_m)$ 满足当前状态不透明性, 如果以下关系式

图 1 (网络版彩图) 网络化离散事件系统的隐私保护控制架构

Figure 1 (Color online) Privacy-preserving control architecture of networked DES.

成立:

$$\begin{aligned} \forall t = L_S &:= \{t \in L(G) \mid \xi_G(q^{init}, t) \in Q_{sec}\}, \\ \exists t' \in L_{NS} &:= \{t \in L(G) \mid \xi_G(q^{init}, t) \in (Q \setminus Q_{sec})\}, \\ P(t) &= P(t'). \end{aligned} \quad (3)$$

3.2 命令执行模块 CE

命令执行模块是用于描述系统从接收控制命令到执行事件之间的转换过程. 将命令执行模块建模为一个有限状态自动机(如图2所示):

$$CE = (Q_{ce}, \Sigma_{ce}, \xi_{ce}, q_{ce}^{init}, Q_{ce,m}), \quad (4)$$

其中,

$$Q_{ce} = \{q^\gamma \mid \gamma \in \Gamma\} \cup \{q_{ce}^{init}\},$$

$$\Sigma_{ce} = \Sigma \cup \Gamma,$$

$$\xi_{ce}: Q_{ce} \times \Sigma_{ce} \rightarrow Q_{ce},$$

$$Q_{ce,m} = \{q_{ce}^{init}\}.$$

转移函数 ξ_{ce} 定义如下:

$$\begin{aligned} (\forall \gamma \in \Gamma) \xi_{ce}(q_{ce}^{init}, \gamma) &= q^\gamma, \\ (\forall q \in \{q^\gamma \mid \gamma \in \Gamma\}) (\forall \sigma \in (\gamma \cup \Sigma_{uc}) \cap \Sigma_{uo}) & \\ \xi_{ce}(q, \sigma) &= q, \\ (\forall q \in \{q^\gamma \mid \gamma \in \Gamma\}) (\forall \sigma \in (\gamma \cup \Sigma_{uc}) \cap \Sigma_o) & \\ \xi_{ce}(q, \sigma) &= q_{ce}^{init}, \\ (\forall \sigma \in \Sigma_{uc}) \xi_{ce}(q_{ce}^{init}, \sigma) &= q_{ce}^{init}. \end{aligned} \quad (5)$$

下面, 我们将简要解释以上构建的模型 CE . 对于状态集合 Q_{ce} , 任意的状态 $q^\gamma (\gamma \in \Gamma)$ 表示命令执行模块正在使用控制命令 γ . 对于转移函数 ξ_{ce} :

- (1) 当命令执行模块接收到控制命令 γ 时, 将会开始使用 γ 并转移到状态 q^γ ;
- (2) 当命令执行模块正在使用控制命令 γ 时, 如果

图 2 命令执行模块 CE 模型示意图

Figure 2 The schematic diagram of the command execution automaton CE .

不可观测事件 $\sigma \in (\gamma \cup \Sigma_{uc}) \cap \Sigma_{uo}$ 被执行, 那么命令执行模块将会重复使用控制命令 γ , 且状态保持不变;

(3) 当命令执行模块正在使用控制命令 γ 时, 如果可观测事件 $\sigma \in (\gamma \cup \Sigma_{uc}) \cap \Sigma_o$ 被执行, 那么命令执行模块的状态将会转移到初始状态 q_{ce}^{init} , 且命令执行模块将等待监督器发送的新控制命令;

(4) 在初始状态 q_{ce}^{init} , 任意的不可控事件 $\sigma \in \Sigma_{uc}$ 都有定义(自循环), 因为不可控事件总是可以发生。

基于以上构造的命令执行模块 CE , 得出如下结论: $|Q_{ce}| = 2^{\mathbb{F}_d}$.

3.3 观测信道 OC

如图1所示, 观测信道是一个模块, 代表着系统 G 发出的事件数据传递到监督器的通信路径. 在观测信道中, 时延不可避免地存在, 我们假设这种时延总是存在上限并将时延上限记作 $\Delta_o \in \mathbb{N}$. 而后把观测信道容量记作 $C_{oc} \in \mathbb{N}^+$, 并将观测信道建模为一个有限状态自动机:

$$OC = (Q_{oc}, \Sigma_{oc}, \xi_{oc}, q_{oc}^{init}), \quad (6)$$

其中,

$$Q_{oc} = \mathbb{N}^{\Sigma_o \times [0: \Delta_o]}(C_{oc}),$$

$$\Sigma_{oc} = \Sigma_o \cup \Sigma_o^{out} \cup \{tick\},$$

$$\xi_{oc} \subseteq Q_{oc} \times \Sigma_{oc} \times Q_{oc},$$

$$q_{oc}^{init} = \emptyset.$$

在给出转移关系 ξ_{oc} 的定义前, 我们首先定义以下关于任意两个多重集合 $q, \bar{q} \in Q_{oc}$ 的相关操作:

$$T^{-0}(q) := \{(\sigma, t)^m \in q \mid t = 0\},$$

$$Tick(q) := \{(\sigma, t-1)^m \mid (\sigma, t)^m \in q\},$$

$\bar{q}$ 包含于 q , 记作 $\bar{q} \subseteq q$, 如果

$$(\forall (\sigma, t)^m \in \bar{q})(\exists n \in \mathbb{N}^+)(\sigma, t)^n \in q \wedge n \geq m,$$

$\boxplus$ 是多重集合的求和操作, 定义为

$$q \boxplus \bar{q} := \{(\sigma, t)^{m+\bar{m}} \mid (\sigma, t)^m \in q \wedge (\sigma, t)^{\bar{m}} \in \bar{q}\}$$

$$\cup \{(\sigma, t)^m \in q \mid (\forall \bar{m} \in \mathbb{N}^+)(\sigma, t)^{\bar{m}} \notin \bar{q}\}$$

$$\cup \{(\sigma, t)^{\bar{m}} \in \bar{q} \mid (\forall m \in \mathbb{N}^+)(\sigma, t)^m \notin q\},$$

当 $\bar{q} \subseteq q$, 那么

$$q - \bar{q} := \{(\sigma, t)^{m-\bar{m}} \mid (\sigma, t)^m \in q \wedge (\sigma, t)^{\bar{m}} \in \bar{q}\}$$

$$\cup \{(\sigma, t)^m \in q \mid (\forall \bar{m} \in \mathbb{N}^+)(\sigma, t)^{\bar{m}} \notin \bar{q}\}.$$

基于以上关于多重集合的操作定义, 转移关系 ξ_{oc} 定义如下:

$$(\forall q \in Q_{oc}) T^{-0}(q) = \emptyset \Rightarrow (q, tick, Tick(q)) \in \xi_{oc},$$

$$(\forall q \in Q_{oc})(\forall \sigma \in \Sigma_o)(q, \sigma, q \boxplus \{(\sigma, \Delta_o)\}) \in \xi_{oc},$$

$$(\forall \emptyset \neq q \in Q_{oc})(\sigma \in \Sigma_o)(\exists m \in \mathbb{N}^+)(\exists n \in \mathbb{N})(\sigma, n)^m \in q \Rightarrow (q, \sigma^{out}, q - \{(\sigma, n)\}) \in \xi_{oc}. \quad (7)$$

下面, 我们将简要解释此模型 OC . 对于状态集合, 除了初始状态 $q_{oc}^{init} = \emptyset$, 其余所有状态均是二元组的多重集合, 并且每个二元组包含两个部分: (1) 一个正在观测信道中传输的可观测事件 σ ; (2) 可观测事件 σ 在观测信道中的最大剩余传输时间 t , 即经过 t 个 $tick$ 事件发生后, 观测信道一定会输出事件 σ , 且在这 t 个 $tick$ 事件发生过程中的任意时刻, 观测信道都有可能输出事件 σ . 对于事件集合 Σ_{oc} , 任意事件 $\sigma^{out} \in \Sigma_o^{out}$ 表示观测信道输出事件 $\sigma \in \Sigma_o$. 对于转移关系 ξ_{oc} :

(1) 在任意状态 q , 如果 q 中所有元组的第二个部分都不为0, 那么事件 $tick$ 可以发生. 之后, q 中所有元组的第二个部分都将减1. 对于任意状态 q , 如果 q 中存在元组的第二个部分为0, 那么事件 $tick$ 将不被定义, 因为此时观测信道必须要输出某个最大剩余传输时间为0的事件;

(2) 在任意状态 q , 如果事件 $\sigma \in \Sigma_o$ 发生了, 意味着一个可观测事件被送入了观测信道中, 那么 OC 将会把此事件 σ 加入信道中, 且附上最大时延 Δ_o , 记为 $q \boxplus \{(\sigma, \Delta_o)\}$;

(3) 在任意状态 q , 如果事件 $\sigma^{out} \in \Sigma_o^{out}$ 发生了, 意味着观测信道输出事件 σ , 那么该事件将从信道中移除, 记为 $q - \{(\sigma, n)\}$. 这里, 需要注意的是, 由于观测信道的状态为多重集合, 事件 σ^{out} 的发生将会导致观测信道具有以下两个性质: 1) 非先进先出; 2) 不确定性. 这是由于观测信道可能存在多个相同的正在传输的可观测事件.

基于以上构造的观测信道 OC , 得出如下结论:

$$|Q_{oc}| = \frac{[|\Sigma_o|(\Delta_o + 1)]^{C_{oc} + 1} - 1}{|\Sigma_o|(\Delta_o + 1) - 1}. \quad (8)$$

3.4 控制信道CC

如图1所示, 控制信道是一个模块, 代表着网络化监督器NS发送的控制命令数据传递到命令执行模块CE的通信路径. 在控制信道中, 时延不可避免地存在. 我们将控制信道中的时延上限记作 $\Delta_c \in \mathbb{N}$. 而后我们将观测信道容量记作 $C_{cc} \in \mathbb{N}^+$, 并可以将控制信道建模为一个有限状态自动机:

$$CC = (Q_{cc}, \Sigma_{cc}, \xi_{cc}, q_{cc}^{init}), \quad (9)$$

其中,

$$Q_{cc} = \mathbb{N}^{\Gamma \times [0:\Delta_c]}(C_{cc}),$$

$$\Sigma_{cc} = \Gamma \cup \Gamma^{in} \cup \{tick\},$$

$$\xi_{cc} \subseteq Q_{cc} \times \Sigma_{cc} \times Q_{cc},$$

$$q_{cc}^{init} = \emptyset.$$

对于任意两个多重集合 $q, \bar{q} \in Q_{cc}$, $T^0(q)$, $Tick(q)$, $\bar{q} \subseteq q$, $q \uplus \bar{q}$, 以及 $q - \bar{q}$ 的定义与第3.3节的定义相同. 转移关系 ξ_{cc} 定义如下:

$$\begin{aligned} (\forall q \in Q_{cc}) T^0(q) = \emptyset &\Rightarrow (q, tick, Tick(q)) \in \xi_{cc}, \\ (\forall q \in Q_{cc}) (\forall \gamma \in \Gamma) (q, \gamma^{in}, q \uplus \{(\gamma, \Delta_c)\}) &\in \xi_{cc}, \\ (\forall \emptyset \neq q \in Q_{cc}) (\gamma \in \Gamma) (\exists m \in \mathbb{N}^+) (\exists n \in \mathbb{N}) (\gamma, n)^m &\in q \Rightarrow (q, \gamma, q - \{(\gamma, n)\}) \in \xi_{cc}. \end{aligned} \quad (10)$$

下面, 我们将简要解释此模型CC. 其与模型OC在建模原理上具有相似性, 即基于时间、进入控制信道的信息以及离开控制信道的信息描述控制信道中信息内容状态的变化. 对于状态集合, 除了初始状态 $q_{cc}^{init} = \emptyset$, 其余所有状态均是二元组的多重集合, 并且每个二元组包含两个部分: 一个正在控制信道中传输的控制命令 γ ; 控制命令 γ 在控制信道中的最大剩余传输时间 t . 对于事件集合 Σ_{cc} , 任意事件 $\gamma^{in} \in \Gamma^{in}$ 表示网络化监督器NS发送控制命令 γ 进入控制信道. 转移关系 ξ_{cc} 的定义与第3.3节中 ξ_{oc} 的定义类似. 基于此模型, 控制信道具有以下两个性质: (1) 非先进先出; (2) 不确定性, 这是由于控制信道可能存在多个相同的控制命令. 基于以上构造的控制信道CC, 我们有以下结论:

$$|Q_{cc}| = \frac{[|\Gamma|(\Delta_c + 1)]^{C_{cc}+1} - 1}{|\Gamma|(\Delta_c + 1) - 1}. \quad (11)$$

3.5 网络化监督器NS

为了避免以下不现实的情形, 即网络化监督器NS会发送无穷多个控制命令进入控制信道, 我们假设每次当网络化监督器NS观测到 $tick$ 事件或从观测信道中输出可观事件后, 其能发送的控制命令的数量上限为 $V \in \mathbb{N}^+$. 本节我们将介绍两个模型: (1) 网络化监督器约束SC, 其目的是用于描述网络化监督器的能力; (2) 网络化监督器NS, 也即本文中的目标合成模块. 下面, 我们将介绍如何建模以上两部分.

网络化监督器约束SC: 网络化监督器的工作模式为: 在系统启动时, 其可以发送初始控制命令即使没有接收到任何可观事件; 之后, 只有当接收到可观事件或观测到 $tick$ 事件发生后, 网络化监督器才可以发送控制命令进入控制信道. 值得注意的是, 如图1所示, 由于观测信道OC的存在, 网络化监督器只能观测到 Σ_o^{out} 中的事件而不是 Σ_o 中的事件. 我们假设发送的控制命令总是可以被网络化监督器观测到. 因此, 可以将网络化监督器约束SC建模为一个有限状态自动机(如图3所示):

$$SC = (Q_{sc}, \Sigma_{sc}, \xi_{sc}, q_{sc}^{init}), \quad (12)$$

其中,

$$Q_{sc} = \{q_{sc}^{init}\} \cup \{q_n \mid n \in [1 : V]\},$$

$$\Sigma_{sc} = \Sigma_o^{out} \cup \Gamma^{in} \cup \{tick\},$$

$$\xi_{sc} : Q_{sc} \times \Sigma_{sc} \rightarrow Q_{sc}.$$

图3 网络化监督器约束SC模型示意图

Figure 3 The schematic diagram of the networked supervisor constraints SC.

转移函数 ξ_{sc} 定义如下:

$$\begin{aligned} (\forall \gamma \in \Gamma^{in}) \xi_{sc}(q_{sc}^{init}, \gamma) &= q_1, \\ (n \in [1 : V-1]) (\forall \gamma \in \Gamma^{in}) \xi_{sc}(q_n, \gamma) &= q_{(n+1)}, \\ (\forall q \in Q_{sc}) (\forall \sigma \in \Sigma_o^{out} \cup \{tick\}) \xi_{sc}(q, \sigma) &= q_{sc}^{init}. \end{aligned} \quad (13)$$

下面, 我们将简要解释此模型SC. 对于状态集合, q_{sc}^{init} 为初始状态并表示两种情况: (1) 自从系统启动后, 网络化监督器还未发送控制命令; (2) 自从上一次控制命令发送后, 网络化监督器已经至少观测到一个可观测事件或一个 $tick$ 事件, 且网络化监督器还未发送控制命令. 状态 $q_n (n \in [1:V])$ 表示自从网络化监督器观测到可观测事件或 $tick$ 事件后, 其已发送 n 个控制命令. 对于事件集合, 任意的 $\gamma^{in} \in \Gamma^{in}$ 表示网络化监督器发送控制命令 γ 进入控制信道. 对于转移函数 ξ_{sc} :

(1) 在初始状态 q_{sc}^{init} , 当网络化监督器发送控制命令 γ 后, 状态会转移至 q_1 , 等待下一次事件观测或选择继续发送控制命令;

(2) 情形2描述的是每次当网络化监督器收到观测数据后, 只要当其发送的控制命令未达到最大值 V 时, 其总是可以继续发送控制命令;

(3) 情形3描述的是当网络化监督器收到观测数据时, 其会返回初始状态 q_{sc}^{init} 并可以决定是否发送控制命令.

基于以上构建的网络化监督器约束模型SC, 我们有如下结论:

$$|Q_{sc}| = V + 1. \quad (14)$$

网络化监督器NS: 我们将网络化监督器NS建模为一个有限状态自动机:

$$NS = (Q_{ns}, \Sigma_{ns}, \xi_{ns}, q_{ns}^{init}), \quad (15)$$

在NS中, $\Sigma_{ns} = \Sigma \cup \Sigma_o^{out} \cup \Gamma^{in} \cup \Gamma \cup \{tick\}$. 除此之外, 网络化监督器NS还应满足以下两个条件:

(1) (网络化可控性)

$$(\forall q \in Q_{ns}) (\forall \sigma \in \Sigma_{ns} - \Gamma^{in}) \xi_{ns}(q, \sigma) \neq \emptyset;$$

(2) (网络化可观性)

$$\begin{aligned} (\forall q \in Q_{ns}) (\forall \sigma \in \Sigma_{ns} - \Sigma_o^{out} \cup \Gamma^{in} \cup \{tick\}) \xi_{ns}(q, \sigma) \neq \emptyset \\ \Rightarrow \xi_{ns}(q, \sigma) = q. \end{aligned}$$

网络化可控性是指NS只能控制 Γ^{in} 中的事件, 即发

送控制命令进入控制信道; 网络化可观性是指只有 $\Sigma_o^{out} \cup \Gamma^{in} \cup \{tick\}$ 中的事件会导致NS状态的变化, 即其只能观测到从观测信道中输出的可观测事件、 $tick$ 事件、以及发送控制命令进入控制信道所对应的事件. 在下文中, 我们把 $C_{ns} = (\Gamma^{in}, \Sigma_o^{out} \cup \Gamma^{in} \cup \{tick\})$ 称作网络化监督器的控制约束.

3.6 窃听器I

如图1所示, 窃听器是一个试图通过观测到的数据来推测系统秘密的恶意窃听器. 关于窃听者的假设总结如下: (1) 窃听者的可观测事件集合为 Σ_o ; (2) 窃听者的先验知识有且仅有系统 G 的模型. 其并不知晓网络化监督器NS的结构以及控制目标. 下面, 我们介绍如何建模此窃听器, 包括两个构建子步骤.

步骤1 因为窃听者的先验知识包括系统 G 的模型, 且只能观测到事件集合 Σ_o , 因此, 我们通过如下子集构建操作来建模窃听者的在线观测先验知识, 具体构建模型如下:

$$P_{\Sigma_o}(G) = (Q_{temp}, \Sigma_{temp}, \xi_{temp}, q_{temp}^{init}), \quad (16)$$

其中,

$$Q_{temp} = 2^Q,$$

$$\Sigma_{temp} = \Sigma,$$

$$\xi_{temp} : Q_{temp} \times \Sigma_{temp} \rightarrow Q_{temp},$$

$$q_{temp}^{init} = UR_{G, \Sigma - \Sigma_o}(q^{init}).$$

实际上, $P_{\Sigma_o}(G)$ 是一个状态估计器, 且当 $P_{\Sigma_o}(G)$ 的状态到达集合 $2^{Q_{sec}} - \{\emptyset\}$ 中的任一状态时, 窃听器已经成功推测出系统的秘密, 即系统秘密已泄露.

步骤2 基于以上构造的 $P_{\Sigma_o}(G)$, 我们将对其进行轻微的结构变化以此来体现系统秘密泄露的情形, 变化后的模型如下:

$$I = (Q_i, \Sigma_i, \xi_i, q_i^{init}). \quad (17)$$

模型I的具体构造步骤如下:

$$(1) Q_i = Q_{temp} \cup \{q^{unsafe}\},$$

$$(2) \Sigma_{temp} = \Sigma \cup \{decode\},$$

$$(3) (\forall q \in Q_i) q \in 2^{Q_{sec}} - \{\emptyset\} \Rightarrow \xi_i(q, decode) = q^{unsafe},$$

$$(4) (\forall q, q' \in Q_i)(\forall \sigma \in \Sigma) \xi_{temp}(q, \sigma) = q' \Rightarrow \xi_i(q, \sigma) = q',$$

$$(5) q_i^{init} = q_{temp}^{init}.$$

在以上构建的模型 I 中, 对于状态集合和事件集合, 我们分别加入了一个新状态 q^{unsafe} 和一个新事件 $decode$ 来表示系统秘密已泄露(步骤1和2). 对于转移函数 ξ_i , 所有定义于 ξ_{temp} 中的转移维持不变(步骤4), 唯一新定义的转移为 $\xi_i(q, decode) = q^{unsafe}$ (步骤3), 用于体现系统秘密已经泄露. 由于事件 $decode$ 对于网络化监督器是不可控的, 因此, 为了强制不透明性并防止系统秘密的泄露, 网络化监督器应防止窃听器转移至状态 q . 基于此构建的窃听器模型 I , 我们有如下结论:

$$|Q| \leq 2^{|Q|} + 1. \quad (18)$$

4 用于实现隐私保护的网络化监督器合成

基于第三章构造的系统模型, 我们可以生成系统的闭环行为. 基于此闭环行为, 我们可以将实现隐私保护的网络化监督器合成问题建模为经典的 Ramadge-Wonham 监督控制问题^[34]. 下面, 我们将介绍本文设计的算法.

4.1 算法设计

基于图1所示的系统架构, 给定系统 G , 命令执行模块 CE , 观测信道 OC , 控制信道 CC , 网络化监督器约束 SC , 网络化监督器 NS , 以及窃听器 I , 我们可以生成系统的闭环行为, 定义为如下的同步积:

$$H = G \parallel CE \parallel OC \parallel CC \parallel SC \parallel I \parallel NS \\ = (Q_b, \Sigma_b, \xi_b, q_b^{init}, Q_{b,m}), \quad (19)$$

其中,

$$Q_b = Q \times Q_{ce} \times Q_{oc} \times Q_{cc} \times Q_{sc} \times Q_i \times Q_{ns},$$

$$\Sigma_b = \Sigma \cup \Sigma_o^{out} \cup \Gamma^{in} \cup \Gamma \cup \{tick, decode\},$$

$$\xi_b \subseteq Q_b \times \Sigma_b \times Q_b,$$

$$q_b^{init} = (q^{init}, q_{ce}^{init}, q_{oc}^{init}, q_{cc}^{init}, q_{sc}^{init}, q_i^{init}, q_{ns}^{init}),$$

$$Q_{b,m} = Q_m \times Q_{ce,m} \times Q_{oc} \times Q_{cc} \times Q_{sc} \times Q_i \times Q_{ns}.$$

基于生成的闭环行为 H , 我们有如下定义:

定义4.1 (网络化不透明性) 给定系统 G , 命令执行模块 CE , 观测信道 OC , 控制信道 CC , 网络化监督器约束 SC , 以及窃听器 I , 对于控制约束 C_{ns} , 网络化监督器 NS 是不透明的, 如果 $Q_{bad} = \{(q, q_{ce}, q_{oc}, q_{cc}, q_{sc}, q_i, q_{ns}) \in Q_b | q_i = q^{unsafe}\}$ 中的任意状态在 H 中均为不可达.

下面, 我们将介绍如何合成实现隐私保护的网络化监督器.

首先, 我们简要解释算法1. 在第1步, 我们生成一个新的系统 P . 在第2步, 基于新的系统 P , 只需要把 $\{(q, q_{ce}, q_{oc}, q_{cc}, q_{sc}, q_i) \in Q_p | q_i = q^{unsafe}\}$ 中的状态(表示系统秘密泄露)从 P 中移除即可得到控制目标 P_r . 在第3步, 我们把 P 和 P_r 分别视作新系统及其控制目标即可得到满足不透明性和非阻塞性的网络化监督器 NS . 基于算法1, 我们有如下结论.

定理4.1 给定系统 G , 命令执行模块 CE , 观测信道 OC , 控制信道 CC , 网络化监督器约束 SC , 以及窃听器 I , 对于控制约束 C_{ns} 存在一个不透明且非阻塞的网络化监督器 NS 当且仅当对于约束 C_{ns} 存在一个监督器 S' (状态集合为 Q'_s) 使得:

$$(1) \{(q, q_{ce}, q_{oc}, q_{cc}, q_{sc}, q_i, q'_s) \in Q_p \times Q'_s \mid q_i = q^{unsafe}\}$$

中的任意状态在 $P \parallel S'$ 中均为不可达;

$$(2) P \parallel S' \text{ 是非阻塞的.}$$

证明: 基于不透明性和非阻塞性的定义, NS 对于控制约束 C_{ns} 是一个不透明且非阻塞的监督器当且仅当: 1) $Q_{bad} = \{(q, q_{ce}, q_{oc}, q_{cc}, q_{sc}, q_i, q_{ns}) \in Q_b | q_i = q^{unsafe}\}$ 中的任意状态在 $P \parallel NS$ 中均为不可达; 2) $P \parallel NS$ 是非阻塞的. 因此, 我们可以将 P 和 NS 分别视作新系统和监督器. 证毕.

定理4.2 给定系统 G , 命令执行模块 CE , 观测信道 OC , 控制信道 CC , 网络化监督器约束 SC , 以及窃听器 I , 算法1生成的监督器 NS (如果非空) 对于控制约束 C_{ns} 是不透明且非阻塞的.

证明: 基于算法1的第3步, 此结论成立.

定理4.3 给定系统 G , 命令执行模块 CE , 观测信道 OC , 控制信道 CC , 网络化监督器约束 SC , 以及窃听器 I , 对于控制约束 C_{ns} 存在一个最大的不透明且非阻塞的网络化监督器 NS .

证明: 因为 $C_{ns} = (\Gamma^{in}, \Sigma_o^{out} \cup \Gamma^{in} \cup \{tick\})$, 且 $\Gamma^{in} \subseteq \Sigma_o^{out} \cup \Gamma^{in} \cup \{tick\}$, 即可控事件集合是可观测事件集合的子集, 因此 normality 等价于可观性^[35]. 又因为

算法1

1. 生成 $P = G \parallel CE \parallel OC \parallel CC \parallel SC \parallel I = (Q_p, \Sigma_p, \xi_p, q_p^{init}, Q_{p,m})$;
2. 计算 $P_r = (Q_{p_r}, \Sigma_{p_r}, \xi_{p_r}, q_{p_r}^{init}, Q_{p_r,m})$:
 - a. $Q_{p_r} = Q_p - \left\{ (q, q_{ce}, q_{oc}, q_{cc}, q_{sc}, q_i) \in Q_p \mid q_i = q^{unsafe} \right\}$;
 - b. $\Sigma_{p_r} = \Sigma_p$;
 - c. $(\forall q, q' \in Q_{p_r})(\forall \sigma \in \Sigma_{p_r}) \xi_p(q, \sigma) = q' \Leftrightarrow \xi_{p_r}(q, \sigma) = q'$;
 - d. $q_{p_r}^{init} = q_p^{init}$;
 - e. $Q_{p_r,m} = Q_{p,m} - \left\{ (q, q_{ce}, q_{oc}, q_{cc}, q_{sc}, q_i) \in Q_p \mid q_i = q^{unsafe} \right\}$;
3. 把 P 和 P_r 分别视作新系统和控制目标, 对于控制约束 C_{ns} 合成监督器 $NS = (Q_{ns}, \Sigma_{ns}, \xi_{ns}, q_{ns}^{init})$ 使得: 1) $P \parallel NS$ 是非阻塞的; 2) $P \parallel NS$ 对于 P_r 是安全的。

可控性和normality在合并操作下是封闭的^[35], 因此, 基于监督控制理论^[35], 最大的不透明且非阻塞的网络化监督器 NS 存在。证毕。

4.2 计算复杂度分析

在本节, 我们将分析算法1的计算复杂度。基于 P 的构造形式, 新系统 P 的状态集合基数为 $|Q_p| = |Q| \times |Q_{ce}| \times |Q_{oc}| \times |Q_{cc}| \times |Q_{sc}| \times |Q_i|$ (每部分具体取值详见3.2~3.6节分析), 新系统 P 的事件集合基数为 $|\Sigma_p| = |\Sigma| + |\Sigma_o| + 2|\Gamma| + 2$ 。基于 C_{ns} 的形式, 可知normality成立, 因此我们可以采用基于normality的合成方法^[34], 从而得到算法1的计算复杂度不超过 $O(|\Sigma_p||Q_p|^2 4^{|Q_p|})$ 。

5 仿真实验

本章将通过一个例子来展示所设计算法的有效

性。对于工业互联网供应链中极为关键的原材料调度场景(如图4所示), 自动导引车(automated guided vehicle, AGV)是连接人-机-物的重要环节。本仿真旨在展示针对工业互联网平台中的AGV调度过程, 本文所提出的隐私保护控制策略在成功实现控制目标的同时, 也能有效防止AGV调度过程中的隐私数据泄露。

在本例中, 两辆AGV需要沿着既定路线搬运物料依次到达站点1, 2, 3。如图所示, 部分站点安装了摄像头用于观测AGV是否已到达该站点。摄像头采集到的传感器数据将通过通讯网络传递给中央控制器, 之后中央控制器会根据收到的传感器数据进行控制决策, 并通过通讯网络将控制命令下发给两辆AGV。在运行过程中, 两辆AGV需要避免同时占用同一站点。

在本例中, 事件集合 $\Sigma = \Sigma_c = \{a_1, a_2, a_3, b_1, b_2, b_3\}$ 。 $a_i(b_i)$ 表示AGV 1 (AGV 2) 进入第 i 个站点。 $\Sigma_{uo} = \{a_2, b_2\}$ 。 控制命令集合 $\Gamma = \{v_1, v_2, v_3\}$ 。 v_1 只允许AGV 1前进, v_2 只允许

图4 (网络版彩图)工业互联网供应链中AGV调度场景示意图

Figure 4 (Color online) The schematic diagram of the scheduling scenario in the supply chain of the industrial Internet.

AGV 2前进, v_3 允许AGV 1和AGV 2中的任意一辆前进. $\Delta_o=1$, $\Delta_c=0$, $V=1$. 图5~7分别展示了系统 G ($Q_{sec}=\{3\}$)、命令执行模块 CE 以及网络化监督器约束 SC . 在图5所示的系统 G 中, 状态3为秘密状态(表示AGV 1先到达3号站点), 状态15为标记状态, 即监督器需要发送控制命令使得系统 G 到达状态15. 基于算法1, 我们可以合成网络化监督器 NS , 如图8所示(Σ_{ns} 为 NS 的事件集合).

对于合成的网络化监督器 NS , 其工作模式为: 在初始状态, NS 发送控制命令 v_2 使得系统 G 执行事件 b_1 , 即AGV 2进入站点1, 之后系统 G 的状态变为状态4. 当事件 b_1 被送入观测信道后, 可能会经历一个 $tick$ 的时延,

图 5 (网络版彩图)系统 G
Figure 5 (Color online) Plant G .

图 6 命令执行模块 CE
Figure 6 Command execution automaton CE .

图 7 网络化监督器约束 SC
Figure 7 Networked supervisor constraints SC .

图 8 网络化监督器 NS
Figure 8 Networked supervisor.

也有可能无时延发生. 当 NS 观测到事件 b_1 时, 监督器 NS 会继续发送控制命令 v_2 使得系统 G 执行事件 b_2 , 即AGV 2进入站点2, 之后系统 G 的状态变为状态8. 由于 b_2 不可观测, 命令执行模块 CE 会继续使用控制命令 v_2 , 使得系统 G 执行事件 b_3 , 之后系统 G 的状态变为状态12, 即AGV 2进入站点3. 类似地, 当事件 b_3 被送入观测信道后, 可能会经历一个 $tick$ 的时延, 也有可能无时延发生. 当 NS 观测到事件 b_3 时, 监督器 NS 会发送控制命令 v_1 使得系统 G 执行事件 a_1 , 即AGV 1进入站点1, 之后系统 G 的状态变为状态13. 当 NS 观测到事件 a_1 时, 监督器 NS 会发送控制命令 v_1 使得系统 G 依次执行事件 a_2 与 a_3 , 即AGV 1进入站点2和站点3, 之后系统 G 的状态变为状态15, 即表示控制目标已达成. 同时, 基于当前状态不透明性的定义, 会暴露系统秘密状态的语言为 $a_1a_2a_3$, 而在网络化监督器 NS 的控制下, 系统 G 的行为是 $b_1b_2b_3a_1a_2a_3$, 因此, 系统的秘密状态不会暴露, 即隐私保护任务也得以完成. 因此, 通过本例, 我们可以看出, 基于本文所提出的隐私保护控制算法合成的网络化监督器一方面完成了给定的控制目标, 另一方面也

成功防止了窃听者推测出系统已到达秘密状态, 因此, 所提算法的有效性得以验证.

6 结论与未来工作

本文针对面向工业互联网中数据不透明性的隐私保护控制问题提出了一套新的基于离散事件系统的建模框架. 该框架显示地建模了工业互联网中不可避免的网络时延现象, 并考虑了一类旨在推测工业系统秘密信息的窃听者. 通过将每个子模块建模为有限状态自动机来模拟其工作模式, 原问题可以被转化为经典的Ramadge-Wonham监督控制问题, 这使得许多现成的合成工具包(例如SuSyNA、Supremica^[39]以及

TCT^[40])可以直接求解本文中用于抵御外界恶意窃听者的网络化监督器. 因此, 本文提出的建模框架与合成算法不再需要开发新的合成工具并为实际工业系统设计者提供了便利.

在未来工作中, 一方面, 我们将进一步研究如何设计高效的网络化监督器合成算法以减小计算复杂度, 包括利用基于抽象的监督器合成方法; 另一方面, 本文仅考虑了通过设计网络化监督器来抵御一种旨在推测系统秘密的窃听者, 在实际工业互联网系统中, 还存在通过篡改传感器与驱动器读数以对系统造成损害的窃听者, 因此, 设计更加鲁棒的网络化监督器以及增加功能更丰富的隐私保护元件(例如隐私数据加密模块)将对于实现提供完备的工业网络安全防御体系具有重要意义.

参考文献

- 1 China Industrial Control Systems Cyber Emergency Response Team, National Industrial Security Industry Alliance. Industrial Internet platform security white paper (in Chinese). 2020 [国家工业信息安全发展研究中心, 工业信息安全产业发展联盟. 工业互联网平台安全白皮书. 2020]
- 2 Yuan Y, Zhang Y, Ding H. Research on key technology of industrial artificial intelligence and its application in predictive maintenance (in Chinese). *Acta Autom Sinica*, 2020, 46: 13–30 [袁烨, 张永, 丁汉. 工业人工智能的关键技术及其在预测性维护中的应用现状. *自动化学报*, 2020, 46: 13–30]
- 3 Yuan Y, Tang X, Zhou W, et al. Data driven discovery of cyber physical systems. *Nat Commun*, 2019, 10: 1–9
- 4 Yuan Y, Ma G, Cheng C, et al. A general end-to-end diagnosis framework for manufacturing systems. *Natl Sci Rev*, 2020, 7: 418–429
- 5 Mazare L. Using unification for opacity properties. In: *Proceedings of the Workshop on Issues in the Theory of Security (WITS'04)*. 2004. 165–176
- 6 Bérard B, Mullins J, Sassolas M. Quantifying opacity. *Math Struct Comp Sci*, 2015, 25: 361–403
- 7 Bryans J W, Koutny M, Ryan P Y A. Modelling opacity using Petri nets. *Electron Notes Theor Comput Sci*, 2015, 121: 101–115
- 8 Saboori A, Hadjicostis C N. Notions of security and opacity in discrete event systems. In: *Proceedings of 46th IEEE Conference on Decision Control*. New Orleans, LA, USA: IEEE, 2007. 5056–5061
- 9 Badouel E, Bednarczyk M, Borzyszkowski A, et al. Concurrent secrets. *Discrete Event Dyn Syst*, 2007, 17: 425–446
- 10 Dubreil J, Darondeau P, Marchand M. Opacity enforcing control synthesis. In: *Proceedings of 9th International Workshop on Discrete Event Systems*. Goteborg, Sweden, 2008. 28–35
- 11 Wu Y C, Lafortune S. Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dyn Syst*, 2013, 23: 307–339
- 12 Saboori A, Hadjicostis C N. Verification of infinite-step opacity and analysis of its complexity. *IFAC Proc Volumes*, 2009, 42: 46–51
- 13 Takai S, Kumar R. Verification and synthesis for secrecy in discrete-event systems. In: *Proceedings of 2009 American Control Conference*. St. Louis, MO, USA: IEEE, 2009. 4741–4746
- 14 Hou J, Yin X, Li S, et al. Abstraction-based synthesis of opacity-enforcing controllers using alternating simulation relations. In: *Proceedings of 58th IEEE Conference on Decision and Control (CDC)*. Nice, France: IEEE, 2019. 7653–7658
- 15 Saboori A, Hadjicostis C N. Opacity-enforcing supervisory strategies for secure discrete event systems. In: *Proceedings of 47th IEEE Conference on Decision and Control (CDC)*. Cancun, Mexico: IEEE, 2008. 889–894
- 16 Saboori A, Hadjicostis C N. Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Trans Automat Contr*, 2012, 57: 1155–1165
- 17 Darondeau P, Marchand H, Ricker L. Enforcing opacity of regular predicates on modal transition systems. *Discrete Event Dyn Syst*, 2015, 25:

251–270

- 18 Dubreil J, Darondeau P, Marchand H. Supervisory control for opacity. *IEEE Trans Automat Contr*, 2010, 55: 1089–1100
- 19 Dubreil J, Darondeau P, Marchand H. Opacity enforcing control synthesis. In: Proceedings of 9th International Workshop on Discrete Event Systems. Goteborg, Sweden, 2008. 28–35
- 20 Falcone Y, Marchand H. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dyn Syst*, 2015, 25: 531–570
- 21 Wu Y C, Lafortune S. Synthesis of insertion functions for enforcement of opacity security properties. *Automatica*, 2014, 50: 1336–1348
- 22 Barcelos R J, Basilio J C. Enforcing current-state opacity through shuffle in event observations. In: Proceedings of 14th IFAC International Workshop on Discrete Event Systems (WODES). Italy, 2018. 106–111
- 23 Ji Y, Wu Y C, Lafortune S. Enforcement of opacity by public and private insertion functions. *Automatica*, 2018, 93: 369–378
- 24 Ji Y, Lafortune S. Enforcing opacity by publicly known edit functions. In: Proceedings of 56th Annual IEEE Conference on Decision and Control (CDC). Melbourne, Australia: IEEE, 2017. 4866–4871
- 25 Ji Y, Yin X, Lafortune S. Opacity enforcement using nondeterministic publicly known edit functions. *IEEE Trans Automat Contr*, 2019, 64: 4369–4376
- 26 Wu Y, Lafortune S. Synthesis of opacity-enforcing insertion functions that can be publicly known. In: Proceedings of 54th IEEE Conference on Decision and Control (CDC). Osaka, Japan: IEEE, 2015. 3506–3513
- 27 Mohajerani S, Ji Y, Lafortune S. Efficient synthesis of edit functions for opacity enforcement using bisimulation-based abstractions. In: Proceedings of 57th IEEE Conference on Decision and Control (CDC). Miami Beach: IEEE, 2018. 3573–3578
- 28 Mohajerani S, Ji Y, Lafortune S. Compositional and abstraction-based approach for synthesis of edit functions for opacity enforcement. *IEEE Trans Automat Contr*, 2019, 65: 3349–3364
- 29 Falcone Y, Marchand H. Runtime enforcement of K-step opacity. In: Proceedings of 52nd IEEE Annual Conference on Decision and Control (CDC). Florence, Italy: IEEE, 2013. 7271–7278
- 30 Lima P M, Carvalho L K, Moreira M V. Confidentiality of cyber-physical systems using event-based cryptography. *IFAC-PapersOnLine*, 2020, 53: 1735–1740
- 31 Cassez F, Dubreil J, Marchand H. Synthesis of opaque systems with static and dynamic masks. *Form Methods Syst Des*, 2012, 40: 88–115
- 32 Yin X, Li S. Synthesis of dynamic masks for infinite-step opacity. *IEEE Trans Automat Contr*, 2019, 65: 1429–1441
- 33 Zhang B, Shu S, Lin F. Maximum information release while ensuring opacity in discrete event systems. *IEEE Trans Automat Sci Eng*, 2015, 12: 1067–1079
- 34 Wonham W M, Cai K. Supervisory Control of Discrete-event Systems. Cham: Springer, 2019. 1–487
- 35 Cassandras C, Lafortune S. Introduction to Discrete Event Systems. 2nd ed. Boston, MA: Springer, 2008. 1–772
- 36 Hopcroft J E, Ullman J D. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 2001, 32: 60–65
- 37 Brandin B A, Wonham W M. Supervisory control of timed discrete-event systems. *IEEE Trans Automat Contr*, 1994, 39: 329–342
- 38 Lin F. Control of networked discrete event systems: Dealing with communication delays and losses. *SIAM J Control Optim*, 2014, 52: 1276–1298
- 39 Akesson K, Fabian M, Flordal H, Malik R. Supremica—An integrated environment for verification synthesis and simulation of discrete event systems. In: Proceedings of 8th International Workshop on Discrete Event Systems. Ann Arbor, MI, 2006. 384–385
- 40 Feng L, Wonham W M. Tct: A computation tool for supervisory control synthesis. In: Proceedings of 8th International Workshop on Discrete Event Systems. Ann Arbor, MI, 2006. 388–389

Design of privacy-preserving control strategy for data opacity of industrial internet

SHEN Yi¹, ZHOU ChunJie¹, HU XiaoYa¹, CHENG Cheng¹, HE Xin² & TAI RuoChen³

¹ School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China;

² School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China;

³ School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

With the continuous development of information and communication technologies (ICT) as well as their applications in the industrial environment, industrial internet comes into being. In comparison to the relatively closed and secure production environment of traditional industries, the industrial internet faces significant security challenges in terms of data privacy. This paper investigates the privacy-preserving control problem for data opacity of the industrial internet to provide a resilient defending strategy against cyber threats, where the system to be controlled is modeled as a networked discrete-event system (DES) and malicious eavesdroppers exist to infer the system secret based on its observations. We here aim at designing a networked supervisor to achieve the following two objectives: (1) eavesdroppers cannot infer the system secret based on its observations, which is formulated as the requirement of opacity enforcement, (2) under the control of the designed networked supervisor, the system could still achieve the predetermined control requirement. To solve the previously mentioned privacy-preserving supervisory control problem, this paper first proposes a privacy-preserving control architecture for networked DES, in which each component is modeled as a finite-state automaton to simulate its dynamics. We design an algorithm based on the constructed models to synthesize the networked supervisor by transforming the original privacy-preserving control problem into a classical Ramadge-Wonham supervisory control problem. Furthermore, this paper proves (1) the designed synthesis algorithm is sound and complete, and (2) the existence of the supremal networked supervisor for this privacy-preserving supervisory control problem.

industrial internet, networked discrete-event systems, privacy-preserving control, networked supervisor, finite state automata, opacity

doi: [10.1360/SST-2021-0407](https://doi.org/10.1360/SST-2021-0407)