

拜占庭攻击下基于以太坊的智慧工厂安全防御策略

杨宪棋^{1,2}, 高庆^{1,2†}, 吕金虎^{1,2}, 张春阳^{1,2}, 王爽³

(1. 北京航空航天大学 自动化科学与电气工程学院, 北京 100083;

2. 中关村实验室, 北京 100080; 3. 东南大学 计算机科学与工程学院, 江苏 南京 211189)

摘要: 针对拜占庭攻击下智慧工厂的防御问题, 本文提出了一种基于以太坊的安全机制. 首先, 本文提出一种基于参考圆的层级信息流动机制. 在该机制下, 系统将节点分为普通节点与关键节点, 占多数的普通节点通过对占少数的关键节点的作恶概率进行评估, 降低恶意关键节点作恶的成功率. 其次, 本文提出信誉驱动的贝叶斯推理方法作为普通节点评估恶意关键节点行为的凭据, 通过综合考虑节点的信誉与节点间的物理距离等因素, 提高评估准确率. 最后, 本文使用协作机械臂作为仿真场景验证所提出方法的可用性与有效性.

关键词: 智慧工厂; 拜占庭攻击; 以太坊; 贝叶斯推理; 工业安全

引用格式: 杨宪棋, 高庆, 吕金虎, 等. 拜占庭攻击下基于以太坊的智慧工厂安全防御策略. 控制理论与应用, 2024, 41(8): 1325 – 1334

DOI: 10.7641/CTA.2023.20673

An Ethereum-based security defense strategy against the Byzantine attack in smart factories

YANG Xian-qi^{1,2}, GAO Qing^{1,2†}, LÜ Jin-hu^{1,2}, ZHANG Chun-yang^{1,2}, WANG Shuang³

(1. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China;

2. China Village National Laboratory, Beijing 100080, China;

3. School of Computer Science and Engineering, Southeast University, Nanjing Jiangsu 211189, China)

Abstract: This paper proposes an Ethereum-based security strategy to defend the Byzantine attack in the smart factories. Firstly, a reference circle-based hierarchical information flow mechanism is proposed, where the nodes in the smart factory are divided into the many common nodes and the few key nodes. The common nodes evaluate the possibility of the key nodes to behave maliciously, by which attacks from the key nodes are more likely to be unsuccessful. Secondly, a reputation-driven Bayesian inference method where the reputation of nodes, the physical distance between nodes are considered to improve the efficiency of evaluation of malicious behaviors. Finally, a series of simulation experiments are carried out using robotic arms and the simulation results are presented to verify the availability and effectiveness of the proposed method.

Key words: smart factories; Byzantine attack; Ethereum; Bayesian inference; industrial security

Citation: YANG Xianqi, GAO Qing, LÜ Jinhu, et al. An Ethereum-based security defense strategy against the Byzantine attack in smart factories. *Control Theory & Applications*, 2024, 41(8): 1325 – 1334

1 引言

随着人工智能、区块链、大数据等新兴技术的蓬勃发展, 传统工业逐步迈向工业4.0时代^[1-3]. 智慧工厂是传统制造业在工业4.0时代升级后的产物^[4], 其意义在于, 通过将上述新兴技术应用在传统工厂车间当中, 使得工业设备之间实现信息交互, 数据之间实现碰撞与融合, 传统制造业实现升级与转型, 从而达到减少开支, 提升生产效率的目的^[5]. 然而, 近年来随着工厂智能化的不断推进, 工业系统当中暴露的安全漏

洞也越来越多, 安全事件发生的频率也变得越来越高. 2020年3月, 钢铁制造商EVRAZ公司在北美分支机构, 包括加拿大和美国的钢铁生产厂均遭受了勒索软件Ryuk攻击, 导致其在北美的分支机构瘫痪, 大多数工厂停止生产; 5月, 瑞士铁路机车制造商Stadler披露公司遭遇黑客窃取公司数据, 导致公司遭受黑客勒索; 2021年11月初, 美国大型乳制品供应商Schreiber Foods遭到勒索软件攻击, 导致系统宕机. 这些安全威胁影响了工业生产的正常运行, 给企业带来了巨大的损

收稿日期: 2022-07-28; 录用日期: 2023-03-30.

†通信作者. E-mail: qing.gao.chance@gmail.com; Tel.: +86 15710016942.

本文责任编辑: 崔巍.

国家自然科学基金项目(61903016)资助.

Supported by the National Natural Science Foundation of China (61903016).

失^[6].

区块链是近年来兴起的一项技术,是密码学、共识算法、分布式数据库等一些传统理论的有机结合^[7].作为一种分布式账本,区块链拥有透明性、不可篡改性及可追溯性等优良性质,能够有效地对抗数据泄露、数据篡改、恶意数据注入等攻击^[8].然而,诸如比特币与以太坊等区块链难以有效地应对拜占庭攻击^[9].拜占庭攻击指的是系统当中存在若干个节点被攻击者占据,通过发送虚假消息来扰乱系统的正常运行^[10].为了解决以区块链作为架构的智慧工厂中存在的拜占庭攻击问题,学术界与工业界做出了许多努力^[11-15].某些区块链在设计时就考虑到了拜占庭容错问题,在此基础上诞生了以实用拜占庭容错算法(practical Byzantine fault tolerance, PBFT)算法为共识算法的区块链^[11].典型的以PBFT为共识算法的区块链有Hyperledger Fabric与Hyperledger Indy^[12].基于该共识,学术界也做了相当多的工作去提升区块链的拜占庭容错能力与吞吐量.文献[13]提出了一种双链结构的区块链,该区块链包含了事务链与信誉链.信誉链采用结合集体签名的同步拜占庭容错,以对抗拜占庭攻击.文献[14]对PBFT算法进行改造,使得该算法在让系统具备拜占庭容错能力的同时,又具备了高吞吐量.在该算法的Request阶段,区块链网络中的控制器会对所有的节点发送验证区块.这些区块链在主节点进行验证,并由节点充当的代理记录主节点验证成功率.该成功率会作为主节点选取阶段的参考.上述的几项工作虽然采取了PBFT共识算法使得网络具备了拜占庭容错,但也使得网络的通讯复杂度达到了 $O(n^2)$ ^[11].因此,此类算法只能用在小规模的网络当中.诚然,有像文献[15]的工作考虑到了大规模节点的状态,但该文献还是将PBFT作为区块链的共识算法,并在此基础上提升区块链的吞吐量.因此,PBFT算法带来的通讯消耗还是无法避免.为了使得将区块链作为底层的智慧工厂在运行时具有可以承受的通讯消耗,可以将以PoW(proof of work)为共识的以太坊作为智慧工厂的架构. PoW算法具有 $O(n)$ 级别的通讯复杂度,比起PBFT的 $O(n^2)$ 通讯复杂度来说是可以接受的^[11,16].然而,以PoW为共识的以太坊并不具备对抗拜占庭攻击的能力.如何使得将以太坊作为架构的智慧工厂拥有对抗拜占庭攻击的能力,是本文主要研究的问题.

具体的,本文提出了基于以太坊的安全防御策略,包含基于参考圆的信息层级流动机制与信誉驱动的贝叶斯推理方法.首先,本文提出了基于参考圆的信息层级流动机制.在该机制下,基于节点与参考圆的相对位置关系,系统将节点分为占多数的普通节点与占少数的关键节点(分别对应第3部分的士兵节点与将军节点).通过信息的层级流动,普通节点对关键节点

的作恶概率进行评估,从而降低恶意关键节点攻击的成功率.其次,本文将信誉机制与贝叶斯推理方法相结合,提出了信誉驱动的贝叶斯推理方法.该方法将节点的信誉值与节点间的物理距离等因素作为先验知识,提高了普通节点对关键节点作恶概率评估的准确性.最后,本文模拟了一个遭受拜占庭攻击的智慧工厂场景,并基于该场景对本文提出的安全防御策略的有效性进行了验证.

本文的主要贡献可以总结为以下两点:

1) 提出了基于以太坊的安全防御策略,在不依赖PBFT算法的同时,使得智慧工厂具备对抗拜占庭攻击的能力;

2) 提出了恶意值约束的信誉更新策略,使得智慧工厂能够去发掘一些以相对保守的方式攻击系统的节点.此类节点会在其信誉值比较高时肆意攻击系统,在其信誉值比较低时又将自身伪装成良性节点去隐藏自身.

本文的其余部分如下:第2部分介绍了一些关于以太坊与拜占庭通过攻击的基本概念;第3部分介绍了所提出的安全策略的具体方法;第4部分描述了实验的具体配置与实验结果;第5部分是对本文的总结.

2 以太坊与拜占庭攻击的基本概念

本节将介绍以太坊的基本架构与运行流程,并对拜占庭攻击的基本概念进行简要阐述.

2.1 以太坊

2.1.1 以太坊基本架构

以太坊是Vitalik Buterin在2013年发布的以太坊黄皮书当中所提出的一种新型区块链^[17].如图1所示,以太坊主要由应用层、合约层、激励层、共识层、网络层、数据层组成.而在这6层中,对于用户比较重要的是合约层中的智能合约与以太坊虚拟机、数据层中存储着区块信息、账户信息与交易信息的世界状态,以及共识层中的共识算法.其中,智能合约是在以太坊中由用户发起创建合约交易进行创建的,被所有用户接受的可执行的代码.智能合约编译后的应用二进制接口(application binary interface, ABI)与字节码被系统存储在相应的合约账户当中,供其他账户去调用^[18-19];以太坊虚拟机(ethereum virtual machine, E-VM)为智能合约的运行提供了类似沙箱的环境,这使得本地环境对代码运行产生的影响可以被忽略,从而达到输入输出的无干扰对应关系^[20];世界状态中存储着所有账户的信息,包括余额、公钥等;共识算法为以太坊账户间的记账权的归属提供了凭据.账户通过共识算法获得记账权,再将自己打包的区块发送至区块链上,从而实现了账本的记录与区块链的延长.在以上基础上,工业设备可以通过连入以太坊平台,获得相应的私钥,并成为以太坊当中的合法节点.以太坊

中的节点可以分为轻节点与全节点. 其中, 轻节点只存储可以去获取完整区块信息的的区块头, 而全节点存储完整的区块信息^[21]. 在资源有限的智慧工厂的场景下, 轻节点一般由资源有限的硬件充当. 相对的, 全节点一般由计算与存储资源充裕的硬件充当. 轻节点负责将来自账户的交易发送至由全节点充当的矿工节点. 全节点作为矿工节点负责将这些交易打包成区块, 并上传至区块链,

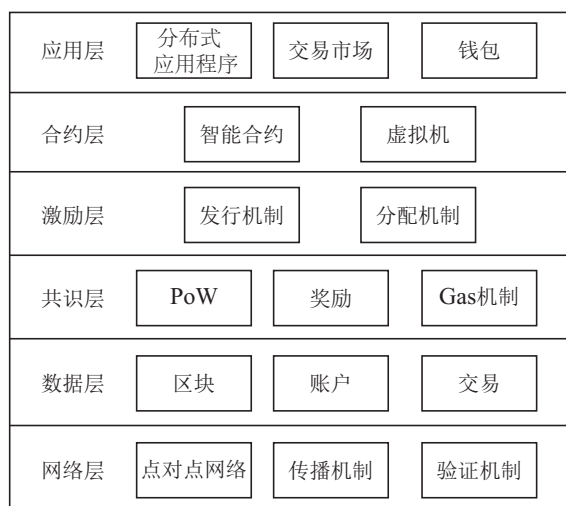


图1 以太坊架构

Fig. 1 The structure of Ethereum

2.1.2 以太坊运行流程

基于交易的以太坊的运行流程可以用交易的生命周期来描述, 如图2所示, 交易的生命周期始于外部账户发起一笔交易. 交易在被创建及签名后被交易发起账户发送至相应节点进行验证, 在验证通过后被节点发送至矿工节点的交易池. 在某个矿工节点执行PoW算法并获得记账权后, 该节点执行其交易池当中的若干个交易, 并将这些交易打包成区块, 上传至区块链中. 在交易执行过程中, 若该交易是创建合约的交易, 则以太坊为该合约创建一个存储着该合约的字节码、ABI等信息的合约账户. 若该交易是普通交易, 则以太坊根据该交易中的信息将交易中包含的数据发送至相应账户. 若该交易是调用合约的交易, 则以太坊根据合约地址找到相应的合约, 根据交易中的输入参数对合约进行调用. 最后, 当交易执行完毕后, 以太坊修改存储着账户信息的世界状态, 并更新链上存储的账户信息.

2.2 拜占庭攻击

拜占庭攻击(Byzantine attack)问题, 最早是在文献[10]当中由Leslie Lamport, Robert Shostak和Marshall Pease共同提出来的. 该问题指的是一群拜占庭军队围绕着敌方的城市驻扎. 各支军队的将军们只能通过信兵互相通信. 当由信兵传递的进攻方案在将军间达成一致时才开始进攻. 在此期间, 信兵被认为是诚实

的, 而将军可能存在叛变行为. 叛变的将军会传递虚假信息迷惑其他将军, 使得进攻方案难以实施. 如何在存在叛变将军的情况下使将军们的意见达成一致, 是拜占庭攻击问题要解决的首要问题. 在工业互联网当中, 关键节点可以被看作是拜占庭将军. 当被控制成为恶意节点后, 该节点就会发动拜占庭攻击去扰乱系统的正常运作, 如通过发送虚假的消息来扰乱系统, 使得系统的功能受到限制^[22-23].

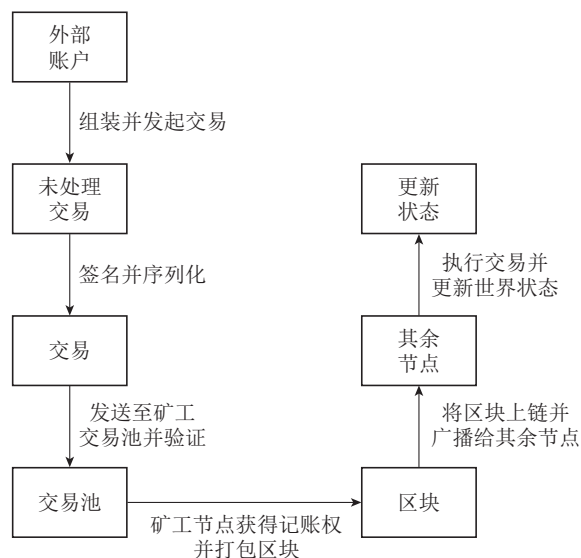


图2 交易的生命周期

Fig. 2 The lifecycle of transactions

3 基于以太坊的安全防御策略

本节详细介绍基于以太坊的安全策略. 首先, 介绍将以以太坊作为架构的智慧工厂应用场景, 如图3所示. 在该智慧工厂场景下, 存在协作机械臂将货物进行组装打包, 并有无车将打包完的货物运送到相应的地点进行装车. 此外, 从区域边缘网关继承算力的本地边缘网关(后文简称为网关)为整个工厂提供算力支持^[24]. 机械臂可将实时要求低且计算复杂度大的任务发送至网关进行处理, 而将计算复杂度低且实时性要求较高的计算任务留在本地处理^[25]. 在以太坊架构下, 为工厂提供算力的网关注册为以太坊平台上的全节点, 而协作机械臂注册为以太坊平台上的轻节点. 当轻节点有处理计算任务或是发送消息的需要时, 将计算任务或是消息作为交易的数据段卸载给全节点进行处理. 全节点作为矿工在执行PoW共识获取记账权后, 执行交易池中的交易并将其打包成区块上传至区块链. 在执行交易的过程中, 若该交易包含计算任务, 则矿工处理计算任务. 当计算任务处理完毕, 矿工将相应的结果返回至轻节点. 然而, 在这样的场景下, 来自轻节点的消息, 在其真实性没有得到确认前, 就被上传到了区块链. 若该消息是虚假信息, 则区块链的不可篡改性将使得该虚假信息难以被修正, 从而为恶意行为提供了生存空间. 本节提出一种安全防御策

略,包括基于参考圆的层级信息流动机制和信誉驱动的贝叶斯推理方法,在虚假消息被上传至区块链前,

实现对其真实性的有效评估,降低虚假消息上链的概率.

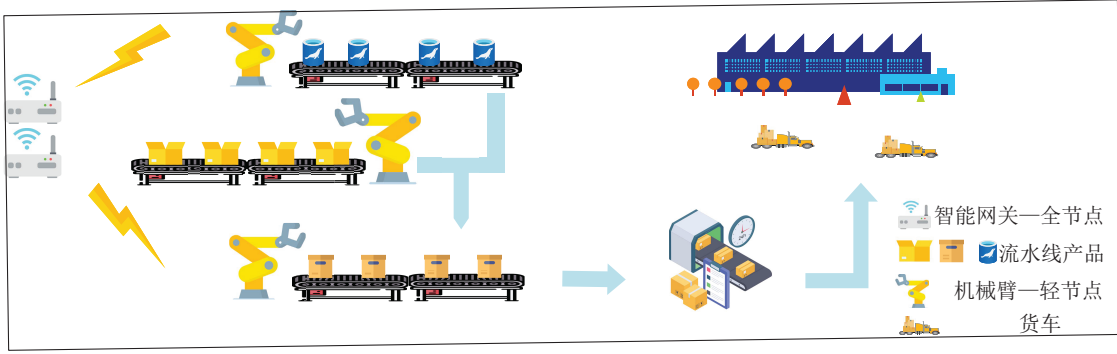


图3 智慧工厂

Fig. 3 The smart factory

3.1 基于参考圆的层级信息流动机制

在基于参考圆的层级信息流动机制中,系统根据轻节点与参考圆的相对位置的关系将轻节点划分为消息发送者、将军节点与士兵节点3种角色.如图4所示,参考圆是以消息发送者为圆心, d_{ra} 为半径的逻辑圆.除消息发送者外,在参考圆内部的其余节点被标记为将军节点.将军节点收到来自消息发送者的消息,并对其真伪进行判断.本文中考虑的拜占庭攻击,是由这类将军节点发起的.将军节点可能会发出虚假判断去混淆系统.在参考圆外部的节点被标记为士兵节点.这些士兵节点从与其余角色的节点间的距离及节点的信誉两个维度通过信誉驱动的贝叶斯推理方法对将军节点的作恶概率进行评估,从而提升了评估的准确性.具体运行过程如下:

步骤1 当网络中的任一节点检测到了某小概率事件的发生(如机械故障),它将会被标记为消息发送者.以该消息发送者为圆心,以预设值 d_{ra} 为半径,生成一个参考圆;

步骤2 在步骤1生成的参考圆内,除消息发送者外的最靠近消息发送者的 n 个节点将被标记为将军节点.生成将军节点的距离向量 $D = [d_1 \ d_2 \ \cdots \ d_n]$,其中 d_i 是将军节点 i 与消息发送者之间的距离;

步骤3 根据极大-极小标准化策略得到将军节点的距离权重向量 $WD = [\omega d_1 \ \omega d_2 \ \cdots \ \omega d_n]$,

$$\omega d_i = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}}, \quad (1)$$

其中: ωd_i 是将军节点 i 的距离权重, $d_{\max}$ 与 $d_{\min}$ 分别为将军节点的距离向量 D 的最大值与最小值;

步骤4 根据式(2)得到将军节点的信誉-距离向量 $RD = [rd_1 \ rd_2 \ \cdots \ rd_n]$,

$$rd_i = r_i + e^{-\alpha d_i}, \quad (2)$$

其中: rd_i 是将军节点 i 的信誉-距离值; r_i 是将军节点

i 的信誉值; α 是距离衰减常数;

步骤5 根据式(3)得到将军节点的信誉-权重向量 $RW = [rw_1 \ rw_2 \ \cdots \ rw_n]$,

$$rw_i = \frac{r_i}{\sum_{k=1}^n r_k}, \quad (3)$$

其中 rw_i 是将军节点 i 的信誉-权重值;

步骤6 步骤2中被选择的 n 个将军节点在接收到来自消息发送者的事件报告后,对于该事件的真实性做出判断.根据这 n 个将军节点的判断,生成将军节点的决策向量 $M = [m_1 \ m_2 \ \cdots \ m_n]$,其中, m_i 是将军节点 i 对于事件真实性的判断,其取值集合为 $\{-1, 1\}$.特别的, $m_i = 1$ 表示将军节点 i 认为该事件已经发生;反之, $m_i = -1$ 表示将军节点 i 认为该事件没有发生;

步骤7 在参考圆范围外的 m 个最靠近参考圆圆心的节点被标记为士兵节点.创建士兵节点的信誉阈值向量 $R_{th} = [r_{th_1} \ r_{th_2} \ \cdots \ r_{th_m}]$ 、信誉向量 $R_s = [r_{s_1} \ r_{s_2} \ \cdots \ r_{s_m}]$ 、以及距离向量 $D_s = [d_{s_1} \ d_{s_2} \ \cdots \ d_{s_m}]$,其中: r_{th_i} , r_{s_i} , d_{s_i} 分别为士兵节点 i 的信誉阈值、信誉值与到参考圆圆心的距离;

步骤8 根据式(4)创建士兵节点距离权重矩阵 $W_s = [w_{s_1} \ w_{s_2} \ \cdots \ w_{s_m}]$,

$$w_{s_i} = \frac{d_{s_i} - d_{s_{\min}}}{d_{s_{\max}} - d_{s_{\min}}}, \quad (4)$$

其中 $d_{s_{\min}}$ 与 $d_{s_{\max}}$ 分别为距离向量 D_s 中元素的最小值与最大值;

步骤9 创建士兵节点的混合距离矩阵 $D_{\text{mix}} = (d_{\text{mix}_{ij}})_{(m \times n)}$.其中 $d_{\text{mix}_{ij}}$ 是士兵节点 i 到将军节点 j 的距离;

步骤10 士兵节点向全节点发起评估请求,该请求内容作为交易的data字段被发送至全节点的交易池中;

步骤11 全节点收取到所有士兵节点的请求后,

抽取其交易池当中的计算任务, 再将这些计算任务打包为区块。在区块产生后, 全节点计算PoW共识当中的哈希难题来争夺记账权。在最早解出哈希难题的全节点获得记账权后, 该节点上传打包的区块至区块链, 并调用被编译成智能合约的信誉驱动的贝叶斯推理方法来获得各个士兵节点对将军节点的作恶概率的评估。该贝叶斯推理方法将在下一节中详细地介绍;

步骤 12 根据信誉驱动的贝叶斯推理方法获得贝叶斯推理结果向量 PF , 并根据式(5)得到最终决策向量 $FD = [fd_1 \ fd_2 \ \cdots \ fd_m]$,

$$\begin{cases} fd_i = 1, & pf_i \geq r_{th_i}, \\ fd_i = -1, & \text{其他}, \end{cases} \quad (5)$$

其中: fd_i 是士兵节点 i 的最终判断, pf_i 是士兵节点 i 的贝叶斯推理结果;

步骤 13 根据公(6)得到最终的推理结果 F , 即

$$F = \sum_{k=1}^m \omega_{s_k} fd_k, \quad (6)$$

若 F 大于等于系统的事件接受阈值 F_{th} , 系统认为消息发送者报告为真, 并将最终的结果 z 设为 1; 否则, 系统认为消息发送者报告为假, 并将最终的结果 z 设为 -1;

步骤 14 根据式(7)更新存储在以太坊世界状态中的所有节点的信息, 即

$$\begin{cases} r_i \leftarrow r_i + \Delta C, & m_i = z, \\ r_{s_i} \leftarrow r_{s_i} + \Delta C, \ r_{th_i} \leftarrow r_{th_i} - \Delta T, \ fd_i = z, \\ r_i \leftarrow r_i - \Delta C, & m_i \neq z, \\ r_{s_i} \leftarrow r_{s_i} - \Delta C, \ r_{th_i} \leftarrow r_{th_i} + \Delta T, \ fd_i \neq z, \end{cases} \quad (7)$$

其中 ΔC 和 ΔT 分别为信誉更新项与阈值更新项。本文称该步骤中更新状态的方式为无恶意值约束的信誉更新策略。

注意到, 以上的设计中没有考虑到系统当中存在狡猾攻击者的情况。由狡猾攻击者操纵的节点被认为是狡猾攻击者节点。这种恶意节点在其信誉值比较高时肆意攻击系统, 在其信誉值比较低时又将自身伪装成良性节点去混淆系统决策。此外, 狡猾攻击者节点会不断的积累自己的信誉, 提高自己对系统的影响力, 并联合其他恶意节点一起对系统展开攻击。若攻击失败, 狡猾攻击者节点会潜藏自己, 在不断积累信誉的同时伺机发动下一次的攻击。当系统中狡猾攻击者节点数目达到一定规模, 系统将被认定为崩坏。比如, 在以 PBFT 为共识的区块链中, 当恶意节点数目达到节点总数的 $\frac{1}{3}$ 时, 全链系统将被认定为安全失效^[11]。

为了使系统具备对抗狡猾攻击者的能力, 并将狡猾攻击者节点从系统中排除, 本文提出了恶意值约束的信誉更新策略。具体的是对前述基于参考圆的层级信息流动机制中的步骤15做如下修正:

1) 根据式(8)更新存储在以太坊世界状态中的将军节点的信息, 即

$$\begin{cases} r_i \leftarrow r_i + (1 - ml_i)S(t), \\ n_i^+ \leftarrow n_i^+ + 1, \ m_i = z, \\ r_i \leftarrow r_i - ml_iT(t), \\ n_i^- \leftarrow n_i^- + 1, \ m_i \neq z, \\ ml_i = \frac{n_i^-}{n_i^+ + n_i^-}, \end{cases} \quad (8)$$

其中: n_i^+ 是将军节点判断正确的次数, n_i^- 是将军节点 i 判断错误的次数, n_i^- 与总判断数之比为该将军节点 i 的恶意值 ml_i 。另外, $S(t)$ 与 $T(t)$ 分别是系统对将军节点的状态激励函数与状态惩罚函数。在 $S(t)$ 与 $T(t)$ 中, 参数 t 为贝叶斯判断的当前轮数;

2) 根据式(9)更新存储在以太坊世界状态中的士兵节点的信息, 即

$$\begin{cases} r_{s_i} \leftarrow r_{s_i} + \Delta C, \ r_{th_i} \leftarrow r_{th_i} - \Delta T, \ fd_i = z, \\ r_{s_i} \leftarrow r_{s_i} - \Delta C, \ r_{th_i} \leftarrow r_{th_i} + \Delta T, \ fd_i \neq z. \end{cases} \quad (9)$$

另外, 引入如下两条规则进一步约束恶意节点的状态更新:

1) 当 ml_i 小于恶意值阈值 MT 时, 将军节点 i 标记为恶意节点;

2) 只有良性的节点可以被允许去增加 n_i^+ 。

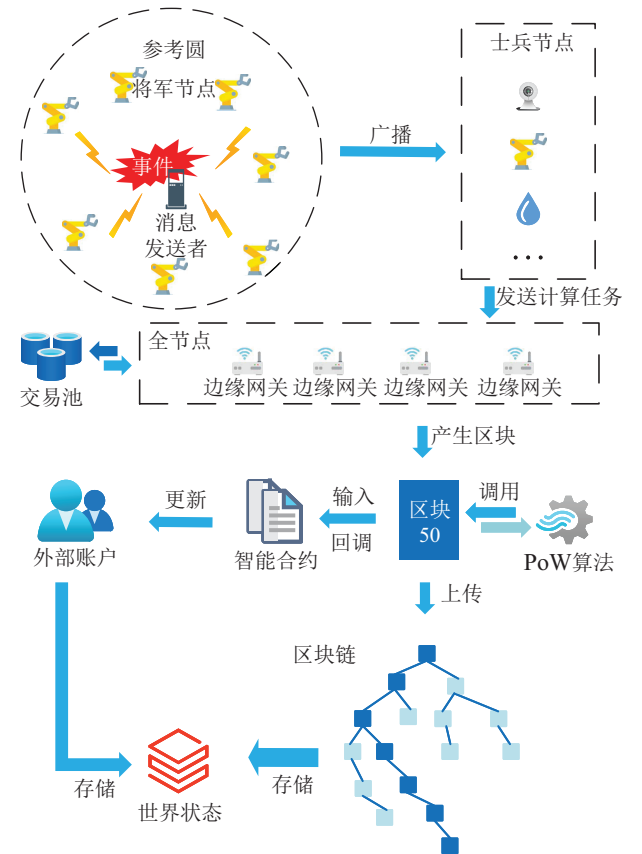


图4 策略运行流程

Fig. 4 The running flow of the strategy

3.2 信誉驱动的贝叶斯推理方法

根据文献[26–28], 贝叶斯推理能够有效地去对抗 Byzantine 攻击, 它在不以 BFT 算法为共识的以太坊系统对抗拜占庭攻击提供了理论支持. 本节提出一种信誉驱动的贝叶斯推理方法, 承接上一节的基于参考圆的信息流动机制, 从士兵节点包括各节点的信誉、士兵节点到参考圆圆心的距离及到将军节点的距离在内的物理属性出发, 对将军节点的作恶概率进行评估. 该机制利用协同决策的思想, 去降低拜占庭攻击的发动的成功率. 该方法运行过程具体如下, 对应的伪代码可见表1.

步骤 1 根据式(10)计算士兵节点的距离和向量 $H = [h_1 \ h_2 \ \cdots \ h_m]$,

$$h_i = \sum_{j=1}^n d_{\text{mix}_{ij}}, \quad (10)$$

其中: h_i 是士兵节点 i 到所有将军节点的距离之和; $d_{\text{mix}_{ij}}$ 是士兵节点 i 到将军节点 j 的距离;

步骤 2 根据式(11)计算士兵节点的中间决策矩阵 $T = (t_{ij})_{(m \times n)}$, 即

$$t_{ij} = m_j \times (\omega_j + wr_j) \times (1 - \frac{d_{\text{mix}_{ij}}}{h_i}), \quad (11)$$

其中: t_{ij} 代表着士兵节点 i 对将军节点 j 的评估水平; T 是一个 $m \times n$ 维的矩阵, 代表着 m 个士兵节点对 n 个将军节点的判断的评估; m_j 代表着将军节点 j 的决策; wr_j 代表着将军节点 j 的信任-权重值, ω_j 代表着将军节点 j 的距离-权重值, $d_{\text{mix}_{ij}}$ 代表着将军节点 j 到士兵节点 i 的距离. 若 $t_{ij} > 0$, 则 t_{ij} 表示在事件确实发生的情况下, 士兵节点 i 认为将军节点 j 没有作恶的概率; 反之, 若 $t_{ij} < 0$, 则 t_{ij} 表示在事件没有发生的状况下, 将军节点作恶的概率;

步骤 3 根据式(12)获得贝叶斯推理结果 pf_i , 即

$$pf_i = P(po_i|e) = \frac{P_e \times \prod_{j=1}^n P(po_{ij}|e)}{P_e \times \prod_{j=1}^n P(po_{ij}|e) + (1 - P_e) \times \prod_{j=1}^n P(po_{ij}|\bar{e})}, \quad (12)$$

其中: e 代表的是事件确实发生, $\bar{e}$ 代表的是事件没有发生, P_e 代表的是事件发生的概率, $P(po_i|e)$ 代表的是在事件确实发生的情况下, 士兵节点 i 认为事件发生的概率, $P(po_{ij}|e)$ 代表的是在事件确实发生的情况下, 士兵节点 i 认为将军节点 j 没有作恶的概率.

通过将信誉驱动的贝叶斯推理方法与基于参考圆的信息层级流动机制相结合, 本文最终得到了基于以太坊的安全防御策略. 该策略能够充分利用系统中的有限信息, 降低来自恶意节点的虚假信息对网络的影响力, 使系统具备对抗拜占庭攻击的能力.

表 1 信誉驱动的贝叶斯推理方法

Table 1 Reputation-driven Bayesian inference method

Require: 将军节点的信誉-权重向量 WR , 将军节点距离-权重向量 W , 将军节点的决策向量, M , 士兵节点的距离矩阵, D_{mix} .

Resure: 士兵节点的贝叶斯推理结果向量, $PF = [pf_1 \ pf_2 \ \cdots \ pf_m]$.

1: **for** $i \in \{1, 2, \cdots, m\}$ **do**

2: **for** $j \in \{1, 2, \cdots, n\}$ **do**

3: 根据下式, 计算士兵节点 i 的距离和:

$$h_i = \sum_{j=1}^n d_{\text{mix}_{ij}},$$

其中 h_i 是士兵节点 i 到所有将军节点的距离之和;

4: **end for**

5: **for** $j \in \{1, 2, \cdots, n\}$ **do**

6: 根据下式, 计算士兵节点 i 的中间决策:

$$t_{ij} = m_j \times (\omega_j + wr_j) \times (1 - \frac{d_{\text{mix}_{ij}}}{h_i});$$

7: **end for**

8: 执行贝叶斯推理, 通过下式, 得到输出结果 pf_i :

$$pf_i = P(po_i|e) = \frac{P_e \times \prod_{j=1}^n P(po_{ij}|e)}{P_e \times \prod_{j=1}^n P(po_{ij}|e) + (1 - P_e) \times \prod_{j=1}^n P(po_{ij}|\bar{e})},$$

其中: $P(po_{ij}|e) = t_{ij} (t_{ij} > 0)$,
 $P(po_{ij}|\bar{e}) = |t_{ij}| (t_{ij} < 0)$.

9: **end for**

10: **return** 士兵节点的贝叶斯推理结果向量,

$$PF = [pf_1 \ pf_2 \ \cdots \ pf_m].$$

4 仿真实验

为了验证本文提出基于以太坊的安全防御策略的有效性, 本文采取了如图3所示的智慧工厂来作为仿真场景. 该场景是一个无人化的流水车间, 其中各个机械臂的任务是将流水线上的各个半成品进行加工组装为成品, 并最后通过无人车的运输, 将产品运到相应的位置. 本实验只考虑机械臂之间的协作, 这些机械臂注册为以太坊中的轻节点, 通过将消息作为交易的字段发出至作为全节点的网关进行处理, 在全节点获得记账权后, 再将消息在全网进行广播. 基于该仿真场景, 本实验在系统初始未崩坏的假设下, 在 MATLAB 平台上进行了一系列仿真实验.

4.1 实验配置

在上述实验场景下, 本文考虑的拜占庭攻击具体发生在消息发送者发出警告消息后将军节点对该消息的真实性进行判断的阶段. 在系统中, 恶意节点通过对警告消息的真实性做出虚假判断, 扰乱系统的正常运行. 为了验证本文提出方法的有效性, 本文对不采取恶意值约束与采取恶意值约束两种信誉更新策略做了对比实验. 其中, 推理的总轮数设为500, 将军

节点与士兵节点的数量分别设为10与100. 在10个将军节点中, 存在1个恶意节点会对网络发起拜占庭攻击. 该恶意节点的攻击程度被分为3个等级, 分别对应着虚假判断数占总判断数的30%、60%与90%的情形. 为了使仿真场景更加贴近真实场景, 在仿真中, 良性节点有1%的概率发生误判的情形, 以描述环境噪声等影响.

4.2 实验结果结果及分析

4.2.1 无恶意值约束的信誉更新策略

无恶意值约束的信誉更新策略下的实验结果如图5~7所示.

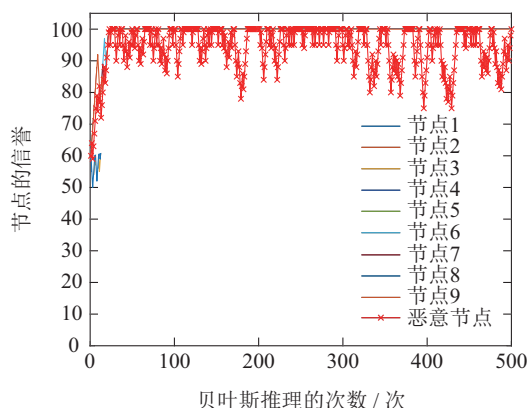


图5 来自恶意节点的虚假判断数占判断总数的30%的情况下, 使用无恶意值约束策略的将军节点信誉值的变化

Fig. 5 The change of reputation of the general nodes by using the non-malicious value constrain s-strategy, when the number of false judgments from the malicious nodes accounts for 30% of the number of total judgments

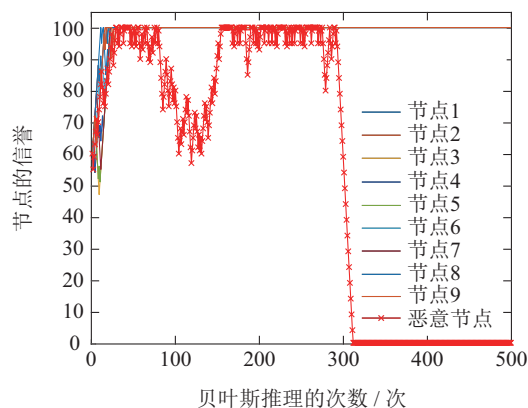


图6 来自恶意节点的恶意节点虚假判断数占判断总数的60%的情况下, 使用无恶意值约束策略的将军节点信誉值的变化

Fig. 6 The change of reputation of the general nodes by using the non-malicious value constrain s-strategy, when the number of false judgments from the malicious nodes accounts for 60% of the number of total judgments

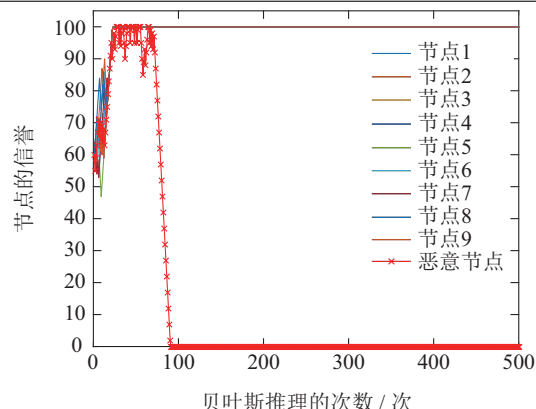


图7 来自恶意节点的恶意节点虚假判断数占判断总数的90%的情况下, 使用无恶意值约束策略的将军节点信誉值的变化

Fig. 7 The change of reputation of the general nodes by using the non-malicious value constrain s-strategy, when the number of false judgments from the malicious nodes accounts for 90% of the number of total judgments

可以看出, 在恶意节点的攻击程度相对较高时, 所提出的信誉更新策略能够较好地发现恶意节点. 在信誉驱动的贝叶斯推理中, 将军节点的信誉值越高, 来自于将军节点的决策更容易被士兵节点所接受. 从这个角度来说, 当恶意将军节点被发现且信誉值被归零后, 该恶意节点将失去活性. 然而, 若恶意节点以一种相对保守的方式去攻击网络, 如在信誉值比较高时肆意攻击系统, 在信誉值比较低时又将自身伪装成良性节点去混淆系统时, 系统将不易发现这类节点. 如图5所示, 当恶意节点采取保守的攻击方式时, 其信誉值一直在一个比较高的范围内波动. 这种情况下, 考虑到环境噪声等影响, 良性节点与恶意节点的区分将十分困难.

4.2.2 恶意值约束的信誉更新策略

为了解决前述恶意节点通过保守攻击方式获取生存空间的问题, 本文采取了恶意值约束的信誉更新策略来作为信誉更新策略. 最终的实验结果如图8~10所示.

由图8可以看出, 在恶意节点的虚假判断数占总判断数的30%的情况下, 该恶意节点的信誉震荡情况得到了明显的改善. 在第82轮推理时, 恶意节点的信誉值归零, 代表恶意节点已被成功发现. 图9~10显示, 所提出的信誉更新策略在恶意节点的攻击水平较高的情形下也具有较好效果, 在判断的初始阶段, 就能够将恶意节点的信誉值快速归零.

进一步地, 为了验证在恶意值约束的信誉更新策略下的安全防御策略在不同的事件发生概率下的推理精度, 本文在不同的事件发生概率下做了仿真实验. 实验结果如图11~13所示.

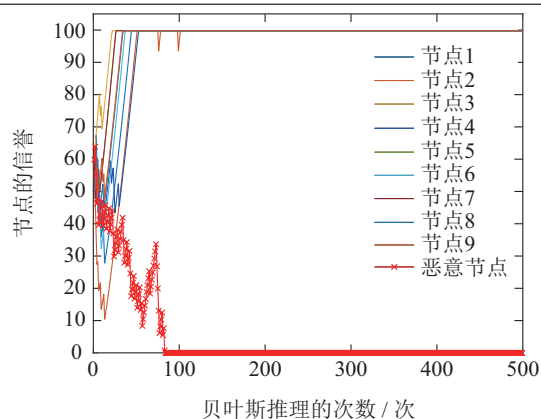


图8 恶意节点虚假判断数占判断总数的30%的情况下,使用恶意值约束策略的将军节点信誉值的变化

Fig. 8 The change of reputation of the general nodes by using the malicious value constrain strategy, when the number of false judgments from the malicious nodes accounts for 30% of the number of total judgments

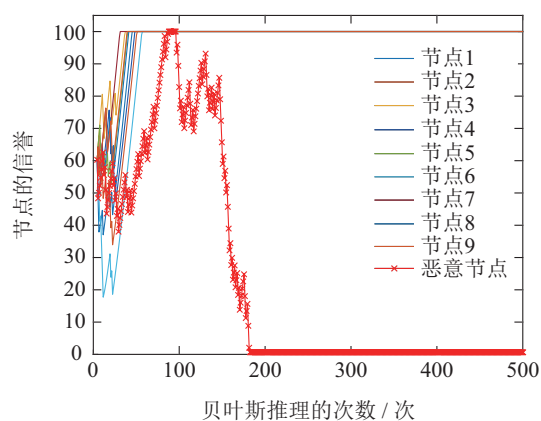


图9 恶意节点虚假判断数占判断总数的60%的情况下,使用恶意值约束策略的将军节点信誉值的变化

Fig. 9 The change of reputation of the general nodes by using the malicious value constrain strategy, when the number of false judgments from the malicious nodes accounts for 60% of the number of total judgments

可以看出,在3种攻击程度与在不同的事件发生概率的情况下,推理的精度在一开始都下降的较快。这一现象的原因在于恶意将军节点在一开始有着比较高的信誉值,对系统的判断的影响比较大,会使得系统对事件的发生情况造成误判。随着推理轮数的增多与恶意节点的信誉值的下降,来自恶意节点的虚假判断难以对系统的最终判断造成影响,这使得推理的精度逐渐收敛。在500轮时,推理的精度基本收敛在了99.8%的水平。从这个角度看,本文提出的安全防御策略在恶意值约束的信誉更新策略下能够应用于不同发生概率的事件。在不同的事件发生概率下,推理

精度能够收敛到比较高的水平。

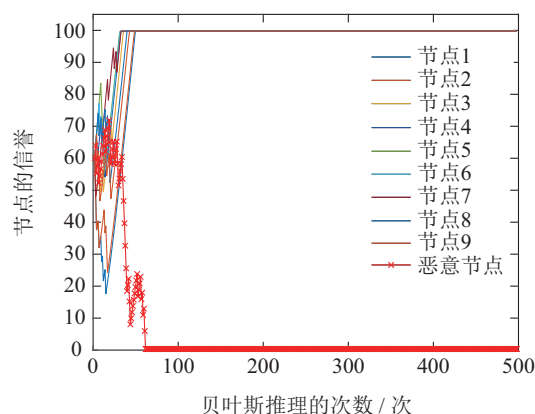


图10 恶意节点虚假判断数占判断总数的90%的情况下,使用无恶意值约束策略的将军节点信誉值的变化

Fig. 10 The change of reputation of the general nodes by using the malicious value constrain strategy, when the number of false judgments from the malicious nodes accounts for 90% of the number of total judgments

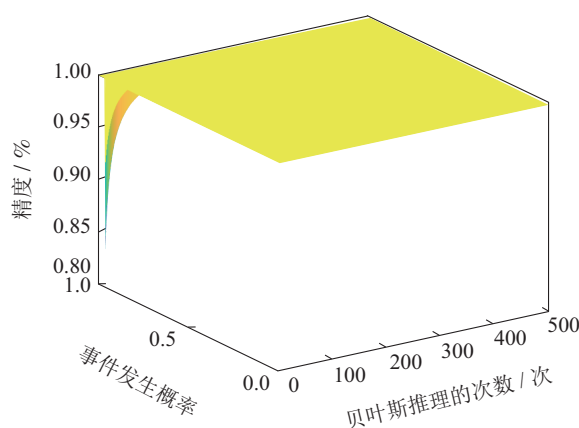


图11 恶意节点虚假判断数占判断总数的30%情况下,执行安全防御策略的推理精度

Fig. 11 The inference accuracy of executing security defense strategy, when the number of false judgements from the malicious nodes accounts for 30% of the number of total judgments

5 结论

本文提出了一种基于以太坊的安全防御策略去对抗智慧工厂中常见的拜占庭攻击。首先,本文设计了一种基于参考圆的信息层级流动机制。在该机制下,占多数的普通节点对占少数的关键节点的作恶概率进行评估,降低了恶意关键节点作恶的概率。其次,为了提高普通节点能够对关键节点的作恶概率评估的准确性,本文设计了一种信誉驱动的贝叶斯推理方法。实验结果表明,本文提出的安全防御策略能够有效地对抗智慧工厂中的拜占庭攻击。未来的研究重点是考虑恶意士兵节点与恶意将军节点相互协作的场景。

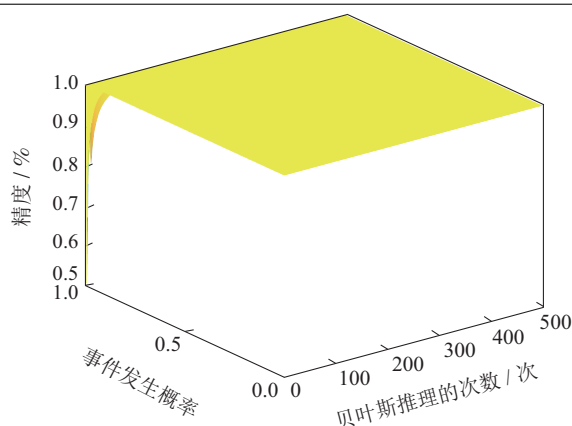


图 12 恶意节点虚假判断数占判断总数的60%情况下, 执行安全防御策略的推理精度

Fig. 12 The inference accuracy of executing security defense strategy, when the number of false judgements from the malicious nodes accounts for 60% of the number of total judgments

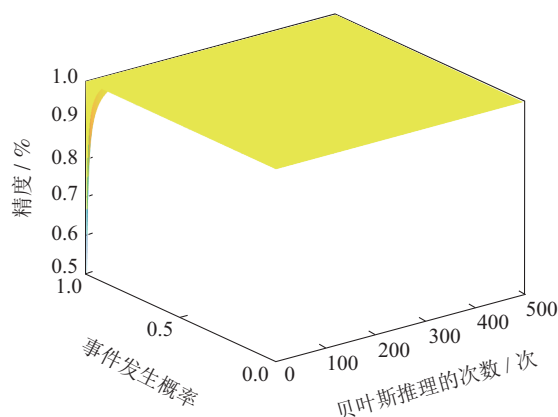


图 13 恶意节点虚假判断数占判断总数的90%情况下, 执行安全防御策略的推理精度

Fig. 13 The inference accuracy of executing security defense strategy, when the number of false judgements from the malicious nodes accounts for 90% of the number of total judgments

参考文献:

- [1] FRANK A G, DALENOGARE L S, AYALA N F. Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, 2019, 210: 15 – 26.
- [2] LASI H, FETTKE P, KEMPER H G, et al. Industry 4.0. *Business & Information Systems Engineering*, 2014, 6(4): 239 – 242.
- [3] MANAVALAN E, JAYAKRISHNA K. A review of internet of things (IoT) embedded sustainable supply chain for industry 4.0 requirements. *Computers & Industrial Engineering*, 2019, 127: 925 – 953.
- [4] WANG S, WAN J, ZHANG D, et al. Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 2016, 101: 158 – 168.
- [5] WAN J, LI J, IMRAN M, et al. A blockchain-based solution for enhancing security and privacy in smart factory. *IEEE Transactions on Industrial Informatics*, 2019, 15(6): 3652 – 3660.
- [6] CHEN Xiayang. *Research on industrial internet information security strategy based on game theory*. Changsha: National University of Defense Technology, 2019.
- [7] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system. *Working Paper*, 2008. <https://bitcoin.org/bitcoin.pdf>.
- [8] BEAANI A, ALCHIERI E, SOUSA J, et al. From Byzantine replication to blockchain: Consensus is only the beginning. *Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. València, Spain: IEEE, 2020: 424 – 436.
- [9] LU Y, HUANG X, DAI Y, et al. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Transactions on Industrial Informatics*, 2019, 16(6): 4177 – 4186.
- [10] LAMPORT L, SHOSTAK R, PEASE M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 1982, 4(3): 382 – 401.
- [11] CASTRO M, LISKOV B. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 2019, 3(1): 1 – 17.
- [12] CACHIN C. Architecture of the Hyperledger blockchain fabric. *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL)*. Chicago, IL, USA, 2016, 310(4): 1 – 4.
- [13] HUANG C, WANG Z, CHEN H, et al. A reputation-based secure, fast, and high incentive blockchain system via sharding. *IEEE Internet of Things Journal*, 2020, 8(6): 4291 – 4304.
- [14] QIU C, YU F R, YAO H, et al. Blockchain-based software-defined industrial Internet of Things: A dueling deep Q-Learning approach. *IEEE Internet of Things Journal*, 2018, 6(3): 4627 – 4639.
- [15] CRAIN T, NATOLI C, GRAMOLI V. Red Belly: A secure, fair and scalable open blockchain. *Proceedings of the 42nd IEEE Symposium on Security and Privacy (SP)*. San Francisco, USA: IEEE, 2021: 466 – 483.
- [16] GERVAIS A, KARAME G O, WÜST K, et al. On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Dallas Texas, USA: Association for Computing Machinery, 2016: 3 – 16.
- [17] WOOD G. Ethereum: A secure decentralised generalised transaction ledger. 2014, <http://gavwood.com/paper.pdf>.
- [18] ZHANG Y, YU W, LI Z, et al. Detecting Ethereum Ponzi schemes based on improved LightGBM algorithm. *IEEE Transactions on Computational Social Systems*, 2021, 9(2): 624 – 637.
- [19] YANG X, GAO Q, LIU K, et al. Smart contracts based supply chain resource management system in the industrial internet. *Proceedings of the 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. Nanchang, China: IEEE, 2021: 31 – 36.
- [20] HILDENBRANDT E, SAXENA M, RODRIGUES N, et al. Kevm: A complete formal semantics of the Ethereum virtual machine. *Proceedings of the 31st IEEE Computer Security Foundations Symposium*. Oxford, United Kingdom: IEEE, 2018: 204 – 217.
- [21] KIM S K, MA Z, MURALI S, et al. Measuring Ethereum network peers. *Proceedings of the Internet Measurement Conference*. Boston, USA: ACM, 2018: 91 – 104.
- [22] DING D, HAN Q, XIANG Y, et al. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 2018, 275: 1674 – 1683.
- [23] MARANO S, MATTA V, TONG L. Distributed detection in the presence of Byzantine attacks. *IEEE Transactions on Signal Processing*, 2008, 57(1): 16 – 29.
- [24] PARK H M, JEON J W. OPC UA based universal edge gateway for legacy equipment. *Proceedings of the 17th International Conference on Industrial Informatics (INDIN)*. Helsinki-Espoo, Finland: IEEE, 2019: 1002 – 1007.

- [25] LIN C C, DENG D J, CHIH Y L, et al. Smart manufacturing scheduling with edge computing using multiclass deep Q network. *IEEE Transactions on Industrial Informatics*, 2019, 15(7): 4276 – 4284.
- [26] KAILKHURA B, HAN Y S, BRAHMA S, et al. Distributed Bayesian detection in the presence of Byzantine data. *IEEE Transactions on Signal Processing*, 2015, 63(19): 5250 – 5263.
- [27] BAI Y, FAN K, ZHANG K, et al. Blockchain-based trust management for agricultural green supply: A game theoretic approach. *Journal of Cleaner Production*, 2021, 310: 127407.
- [28] HASHLAMOUN W, BRAHMA S, VARSHNEY P K. Audit bit based distributed Bayesian detection in the presence of Byzantines. *IEEE Transactions on Signal and Information Processing over Networks*, 2018, 4(4): 643 – 655.

作者简介:

杨宪棋 硕士研究生, 目前研究方向为区块链与工业互联网,

E-mail: xqyang@buaa.edu.cn;

高庆 教授, 目前研究方向为智能系统与控制理论, E-mail:

gaoqing@buaa.edu.cn;

吕金虎 教授, 目前研究方向为智能系统、协同控制理论与技术、复杂网络与大数据, E-mail: jhlu@iss.ac.cn;

张春阳 硕士研究生, 目前研究方向为智能控制理论与深度学习, E-mail: chunyangzhang@buaa.edu.cn;

王爽 讲师, 目前研究方向为任务调度、性能分析、云计算、强化学习, E-mail: shuangwang@seu.edu.cn.