基于区块链智能合约的物联网恶意节点检测和定位

黄豪杰[1]，吴晓晓[1,2]，李刚强[1]
（1. 深圳大学电子与信息工程学院，广东 深圳 518060；2. 鹏城实验室，广东 深圳 518055）

摘　要：随着物联网中分布式设备数量的爆发式增长，设备之间的协作和优化算法的安全问题成为物联网系统研究的前沿问题。物联网中的分布式算法依赖于单个智能体的本地计算和近邻间通信来迭代地解决一类广泛的、受约束的优化问题，但是容易遭受来自内部恶意节点的数据注入攻击。针对现有的检测方法大部分为本地运行，并且存在数据封闭、单点故障、检测过程不透明等问题，考虑使用区块链技术和智能合约对网络中存在的恶意节点进行检测。所提方法基于区块链技术的去中心化和多地备份特性实现了数据共享，避免了单点故障问题。另外，利用智能合约的合约代码、执行过程及结果公开透明且合约代码与结果不可篡改等特性保证检测过程可追溯和可验证。最后，采用平均共识算法并基于树莓派平台对所提方法进行验证分析。

关键词：物联网；数据注入攻击；区块链；智能合约；恶意节点
中图分类号：TN92
文献标识码：A
doi: 10.11959/j.issn.2096–3750.2020.00172

Anomaly detection and location of malicious node for IoT based on smart contract in blockchain network

HUANG Haojie[1], WU Xiaoxiao[1,2], LI Gangqiang[1]
1. College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China
2. Peng Cheng Laboratory, Shenzhen 518055, China

Abstract: With the explosive growth of the number of distributed devices in the Internet of things (IoT) network, the security of decentralized multi-agent optimization algorithm has become the forefront problem. The distributed algorithms in the IoT network are vulnerable to data injection attacks from internal malicious node because each agent locally estimates its state without any supervision. In general, the detection methods for malicious node run independently in each agent, inducing issues such as closed data, single points of failure, opaque detection processes, and so on. The proposed strategy considered detecting via an aid of blockchain technology and smart contracts in Ethereum to detect malicious node in the network. Based on the decentralized and multiple backup features of blockchain technology, the multi-site backup features of the blockchain technology enabled data sharing and avoided single point failure. In addition, the contract code, execution process and result of the smart contract were open and transparent, and the contract code and result could not be tampered to ensure that the detection process could be traced and verified. Finally, the average consensus algorithm was adopted as an example, and the proposed strategy was verified on a simplified IoT network implemented by Raspberry Pi.

Key words: Internet of things (IoT), data injection attack, blockchain, smart contract, malicious node

收稿日期：2020–03–26；**修回日期：**2020–05–03
通信作者：吴晓晓，xxwu.eesissi@szu.edu.cn
基金项目：国家自然科学基金资助项目（No.61701315）；深圳市科技研发基金基础研究项目（No.JCYJ20170817101149906）
Foundation Items: The National Natural Science Foundation of China (No.61701315), The Basic Research Project of Shenzhen Technology Research and Development Fund (No.JCYJ20170817101149906)

1　引言

物联网技术推动着新一轮的信息化浪潮，对于实现物理设备之间的实时控制、管理和决策具有重要意义[1]。近年来，随着设备成本和通信成本的迅速下降，物联网技术得到快速发展。据 IHS Markit 公司估计，到 2030 年接入物联网的智能设备将超过 1 250 亿台[2]。一方面，大量物联网设备的接入改善了人们的生活，提高了社会生产效率；另一方面，大部分设备的计算能力和处理效率不同，需要依赖分布式优化算法来进行协同合作及计算资源分配。物联网中分布式的优化算法通常称为基于 Gossip 的算法或网络偏移算法，该算法依赖于单个智能体的本地计算和近邻间通信来迭代地解决一类广泛的、受约束的优化问题[3-5]。但是，在物联网中运行分布式算法的潜在危险在于必须确保分布式优化的结果不受到恶意攻击。物联网设备面对的攻击主要来自两方面，一方面是外部的攻击行为，另一方面是内部节点受到入侵后被控制，从而在内部发起数据注入攻击行为[6]。针对外部攻击，可以使用加密算法加密或者使用防火墙等工具应对；而内部攻击一般具有隐匿性和伪装性，所以更难检测。事实上，该情况下的网络非常容易受到攻击，即使只有一个内部节点受到攻击也会影响整个网络的优化结果。对于网络中节点的协同攻击，基于 Gossip 的分布式算法将会偏离恶意节点所期望的状态，从而导致整个网络的资源分配完全失衡[7-10]。

数据注入攻击检测理论早已确立，国内外的学者对此进行了大量研究。Bolouki 等在文献[11]中第一次描述了这种可引导系统偏离正常共识状态的恶意节点。Kailkhura 等[12]提出了一种适用于同步平均共识算法的似然比检验法，用来检测数据注入攻击，但是该方法不适用于广泛使用的异步平均共识算法。Su 等[13]建议丢弃极端的状态值，从而减小恶意节点对整个网络的影响，但是该做法会造成信息丢失。Yan 等在文献[14]中利用物理模型的先验知识去估计系统收敛速度来判断网络是否有恶意节点。检测依据了攻击者的数据注入会让系统收敛速度变慢，通过检测指数收敛趋势是否异常来判断攻击者是否存在。但是此方法具有一定的局限性，必须有先验知识才能估计正常收敛速度。谢晋阳等[15]考虑了无线传感网络中攻击者容易俘获正常节点注入虚假数据的情景，提出了一种对事件源能量感知值相近的特征节点的恶意节点检测机制。通过计算事件源的能量值，建立良性节点的坐标系，根据待检测节点与事件源的距离计算值以及与距离感知值的差异判断节点是否为恶意节点。然而，此类算法对网络的要求较高，所需要的节点较多，任意两个节点的数据已知，恶意节点的数目较少，同时恶意节点不会发生协同攻击。王欣等[16]提出了一种基于自适应度量阈值裁决机制的恶意节点筛选算法，该算法假设网络中存在中央控制节点，通过抽样获取节点的自适应度量阈值，然后与中央控制节点进行阈值对比来判断节点是外来节点还是恶意节点。恶意节点的分类通过聚类方式实现，但是该方法不适用于完全分布式的优化算法。季薇等[17]提出了一种将信誉模型与一致性融合相结合的分布式智能入侵防御方案，每个认知用户作为独立的融合中心，并采用冲突惩罚机制对认知用户的信誉值进行更新计算。其中，诚实用户在数据融合中的占比会越来越大，而恶意用户的占比越来越小，从而促使智能的恶意用户放弃攻击，达到网络收敛的目的，但是该方案要求协同攻击下的恶意节点之间是一跳可达的。Wu 和 Gentz 等[7-9]提出了基于时间和空间差分的策略，应用于恶意节点的检测和定位。这些基于分数设计的方法具有不错的性能，可以更有效、准确地揭示攻击者的行为。该方案基于平均一致性共识法则将局部的通信信息带入概率模型，从而判断邻居节点中是否存在恶意节点，在检测完成后通过定位策略将恶意节点踢出网络。

相较而言，Wu 和 Gentz 等[7-9]所提的模型能覆盖较多的场景，且使用限制较少，但是仍然存在一些不足，具体如下。

1)“数据孤岛”问题：网络里的信息分散存储于各节点内部，且各节点存储的数据不尽相同，使得数据未能实现共享，形成了“数据孤岛”，导致数据不能得到充分利用，大幅度降低了数据价值。

2) 数据维护困难：由于数据分散存储且没有副本，同时易发生单点故障，导致部分数据损失，不利于后期的复盘、分析和利用。

3) 检测过程不透明：节点的检测算法都在本地运行，使得外部不知道其具体检测过程，检测结果也未公开。

为了解决上述问题，本文基于文献[8]的平均共识算法，利用区块链和智能合约技术搭建了一个攻击检测系统。对于网络安全领域，区块链技术发挥

着重要作用[18-19]。因此，本文所提方法利用区块链技术实现数据的共享和备份，并保证数据的真实性，再利用智能合约进行攻击检测，使得检测过程透明、公开、可追溯。该检测系统可以为类似于物联网的各种分布式网络提供安全保障，因此，具有广泛的应用意义。

2 基于平均共识的攻击者检测模型

物联网是由智能手机、个人计算机及传感器等终端互相连接成的一个全球型的分布式网络[20]。设备可以通过无线传感器网络等方式进行通信，设备之间的状态通常是某一优化目标的解，如参数估计、源定位等[21-23]。在物联网中，每个接入设备可以被看作一个独立的智能体，物联网可以泛化为一个由多个智能体组成的分布式网络[24-25]。当物联网接入大量智能体时，一般需要对有限的资源进行合理分配，整个分配过程由所有智能体共同协作完成。在资源分配过程中，每个节点都会与邻居节点交换数据。正常节点会在交换数据后更新自身的状态，恶意节点则会发起数据注入攻击，在参与状态交换时仍然保持自身的状态，数据注入攻击下的分布式物联网模型如图 1 所示。一般情况下，恶意节点为了隐匿自身的攻击行为，会在每次交换状态时添加一个衰减的噪声，从而模拟正常节点的收敛状态。针对恶意节点的变化过程，通过邻居节点收集的状态可以对恶意节点进行检测与定位。

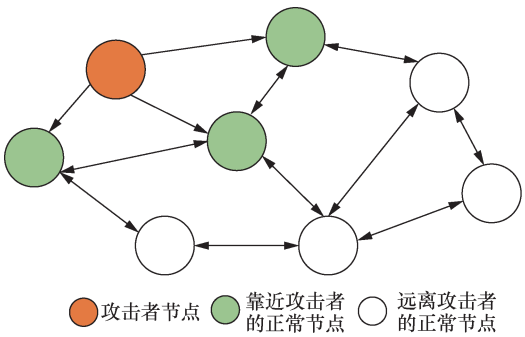


图 1 数据注入攻击下的分布式物联网模型

为了便于研究，将物联网可以抽象为一个无向的连接图 $G=(V,E)$，其中，$V=\{1,\cdots,n\}$ 是网络中所有节点的集合，$E\subseteq V\times V$ 表示所有节点的边，$N_i\in V$ 表示节点 i 的邻居节点的集合。在下述计算式中有如下定义：加粗字母表示向量或矩阵，如向量 $\boldsymbol{x}$，x_i 表示向量 $\boldsymbol{x}$ 中的第 i 个元素。

2.1 基于平均共识的攻击者模型

假设网络中共有 n 个节点，且存在恶意节点的集合 $V_s\subseteq V$，$V_s=\{1,\cdots,n_s\}$，$n_s<n$，则正常节点的集合为 $V_r=V/V_s$。所有节点的状态可以表示为 $\boldsymbol{x}^k(t)=(x_1^k(t),\cdots,x_n^k(t))^{\mathrm{T}}$，其中，$t\in\mathbb{N}$ 表示算法迭代次数，$\mathbb{N}$ 为自然数集，$k\in\{1,\cdots,K\}$ 表示共识过程中的一个实例，$x_i^k(t)$ 表示节点 i 在第 k 个实例中第 t 次迭代时的状态值。假设节点 $i\in V$，初始值为 $x_i^k(0)=\gamma_i^k$，则网络的共识目标为[8]

$$x_{\mathrm{av}}^k=\frac{1}{n}\mathbf{1}^{\mathrm{T}}x^k(0)=\frac{1}{n}\mathbf{1}^{\mathrm{T}}\gamma^k \tag{1}$$

其中，$\mathbf{1}$ 表示元素全为 1 的列向量。在平均共识算法运行中，节点之间的通信基于随机 Gossip 协议的异步模式[26]。在异步模式下，每个时段网络中仅有一个节点 i 被唤醒，节点 i 被唤醒后根据概率 p_{ij} 选择一个邻居节点 $j\in N_i$ 交换并更新自身的状态。其中，p_{ij} 为节点 i 选择节点 j 的概率，每次的交换过程可以表示为

$$x_i(t+1)=x_j(t+1)=\frac{x_i(t)+x_j(t)}{2} \tag{2}$$

经过 $t\to\infty$ 迭代后，网络会收敛至 x_{av}^k，整个共识过程可描述为[8]

$$x^k(t)=\boldsymbol{W}(t-1)x^k(t-1) \tag{3}$$

其中，$\boldsymbol{W}(t)$ 为第 k 个实例中第 t 次迭代时的状态转移矩阵。定义 $[\boldsymbol{P}]_{ij}=p_{ij}$，$[\boldsymbol{\Sigma}]_{ii}=\sum_{ij=1}^{n}(p_{ij}+p_{ji})$，则转移矩阵的期望可以表示为[8]

$$\overline{\boldsymbol{W}}=\mathrm{E}\left[\boldsymbol{W}(t)\right]=\mathbf{I}-\frac{1}{2n}\boldsymbol{\Sigma}+\frac{\boldsymbol{P}+\boldsymbol{P}^{\mathrm{T}}}{2n} \tag{4}$$

其中，$\mathbf{I}$ 为单位矩阵。根据式(3)和式(4)可以得到网络中状态值的期望为[8]

$$\mathrm{E}\left[x^k(t)|x^k(0)\right]=\overline{\boldsymbol{W}}\mathrm{E}\left[x^k(t-1)|x^k(0)\right]=\overline{\boldsymbol{W}}^t x^k(0) \tag{5}$$

如果网络中存在恶意节点，并且恶意节点对网络进行协同攻击，网络的状态均值将跟随恶意节点的状态偏离网络的初始目标。假定恶意节点希望将网络的共识目标引导至 $\alpha^k\neq x_{\mathrm{av}}^k$，则整个网络中节点的共识最终将收敛到式(6)[8]。

$$\lim_{t-\infty}\boldsymbol{x}^k(t)=\alpha^k\mathbf{1} \tag{6}$$

在数据更新过程中，为了防止自身因为状态不变而被轻易地检测到，恶意节点会在交换状态时添加随着时间衰减的噪声，如式(7)所示[8]。

$$x_j^k(t)=\alpha^k+m_j^k(t) \tag{7}$$

其中，$m_j^k(t)$ 是恶意节点为了模仿正常收敛状态而添加的零均值噪声。具体来说，如果恶意节点一直保持状态不变，则邻居节点很容易发现它们的恶意攻击行为。因此，恶意节点会在自身的状态中添加随着时间衰减的噪声，从而增加自身的隐匿性。文献[8]证明了在恶意节点添加零均值衰减噪声后，系统仍然会一致收敛至恶意节点所期望的目标共识状态。针对上述问题，本文应用时间差分和空间差分两种模型来检测并定位恶意节点[8]。

2.2　时间差分检测策略

当网络中存在恶意节点时，恶意节点 $s\in V_s$ 会引导共识结果偏离网络的初始目标，因此，恶意节点的初始状态均值不等于正常节点 $j\in V_r$ 的初始状态均值，即 $\mathrm{E}[x_s^k(0)]=\bar{\alpha}\neq\bar{\gamma}=\mathrm{E}[x_j^k(0)]$。随着算法的不断迭代，当 $t=T$ 时，网络会收敛到最终的共识目标值 $\mathrm{E}[x_s^k(\infty)]$。其中，$T$ 为每次运行实例中节点状态值与最终收敛目标值在满足一定误差条件下算法所需的迭代次数。因此，定义如式(8)所示的度量指标对节点状态的变化差异进行测量[8]。

$$\xi_{ij}:=\frac{1}{K}\sum_{k=1}^{K}\left(x_j^k(T)-x_j^k(0)\right),i\in V_r,j\in N_i \tag{8}$$

当 T 足够大时，网络中所有节点的状态值与最终收敛目标值的误差将趋于零，此时认为网络已达成共识，$x_j^k(T)$ 和 $x_j^k(0)$ 分别为节点 j 的最终收敛结果和初始状态值。在式(8)中，$k\in\{1,\cdots,K\}$ 为算法运行的一个实例，观察更多的实例可以帮助提高检测的精度，具体的检测指标如式(9)所示[8]。

$$D_1^i:=\sum_{j\in N_i}\left|\xi_{ij}-\bar{\xi}\right|\underset{H_0^i}{\overset{H_1^i}{\gtrless}}\delta_1,\bar{\xi}=\frac{1}{|N_i|}\sum_{j\in N_i}\xi_{ij} \tag{9}$$

其中，$\delta_1>0$ 为预先设置的检测阈值，H_0^i 表示节点 i 无恶意的邻居节点，H_1^i 表示节点 i 有恶意的邻居节点。正常节点检测到网络存在恶意攻击行为后开始邻域定位任务，邻域定位任务通过如式(10)所示的指标完成[8]。

$$L_1^{ij}:=\left|\xi_{ij}\right|\underset{H_1^{ij}}{\overset{H_0^{ij}}{\gtrless}}\epsilon_1,\forall j\in N_i \tag{10}$$

其中，H_0^{ij} 表示节点 i 的邻居节点 j 不是恶意节点，H_1^{ij} 表示节点 i 的邻居节点 j 是恶意节点，$\epsilon_1>0$ 为预设的定位任务阈值。

2.3　空间差分检测策略

时间差分只考虑了节点的收敛状态值和初始值，没有利用共识过程中节点的中间状态。为了挖掘中间状态的信息，从而更好地检测和定位恶意节点，文献[8]引入了空间差分方法。该方法假设在 $0<t<\infty$ 时，有 $\mathrm{E}[x_i^k(t)-x_j^k(t)\,|\,H_0]=0$，$\mathrm{E}[x_i^k(t)-x_j^k(t)\,|\,H_1]\neq 0$，即网络中存在攻击者时，节点与邻居节点的差异的期望不为 0。基于此，检测策略引入指标如式(11)和式(12)所示[8]。

$$\varphi_{im}^k:=\sum_{t=0}^{T}\left(x_m^k(t)-\bar{x}_i^k(t)\right),\bar{x}_i^k(t)=\frac{1}{|N_i|}\sum_{j\in N_i}x_j^k(t) \tag{11}$$

$$D_2^i:=\frac{1}{|N_i|}\sum_{j\in N_i}\left(\frac{1}{K}\sum_{k=1}^{K}\varphi_{ij}^k\right)^2\underset{H_0^i}{\overset{H_1^i}{\gtrless}}\delta_{\mathrm{II}} \tag{12}$$

在式(11)中，$i\in V_r$，$m\in N_i\cup\{i\}$；在式(12)中，$\delta_{\mathrm{II}}>0$ 为空间差分策略攻击检测方法的预设阈值。为了进行恶意节点定位任务，定义了指标如式(13)和式(14)所示[8]。

$$\tilde{\varphi}_{ij}^k:=\sum_{t=0}^{T}\left(x_j^k(t)-x_i^k(t)\right)-\varphi_{ii}^k \tag{13}$$

$$L_2^{ij}:=\left(\frac{1}{K}\sum_{k=1}^{K}\tilde{\varphi}_{ij}^k\right)^2\underset{H_0^{ij}}{\overset{H_1^{ij}}{\gtrless}}\epsilon_{\mathrm{II}},\forall j\in N_i \tag{14}$$

在式(13)中，φ_{ii}^k 可以通过节点 i 由式(11)计算得到，$\tilde{\varphi}_{ij}^k$ 用来计算邻居节点 j 与节点 i 自身的差异程度；在式(14)中，$\epsilon_{\mathrm{II}}>0$ 为预设的定位任务阈值。

3　区块链和智能合约

3.1　区块链底层数据结构

区块链是一种由多方共同维护，并且利用密码学保证传输和访问安全，能够实现数据一致存储、难以被篡改、防止抵赖的记账技术，也称为分布式账本技术[27]。区块链技术最早于 2008 年在文献[28]中被提出，经过多年的发展，现在已衍生出多种不同的版本[29-30]。但是，其底层的数据结构和核心的交易模型没有改变。区块链是由多个区块依次链式连接而成，每个区块又可以分为区块头和区块体两部分，区块结构示意图如图 2 所示。

由图 2 的区块结构可知，区块头存储了版本号、时间戳、随机数、前一区块头哈希值、目标难度值

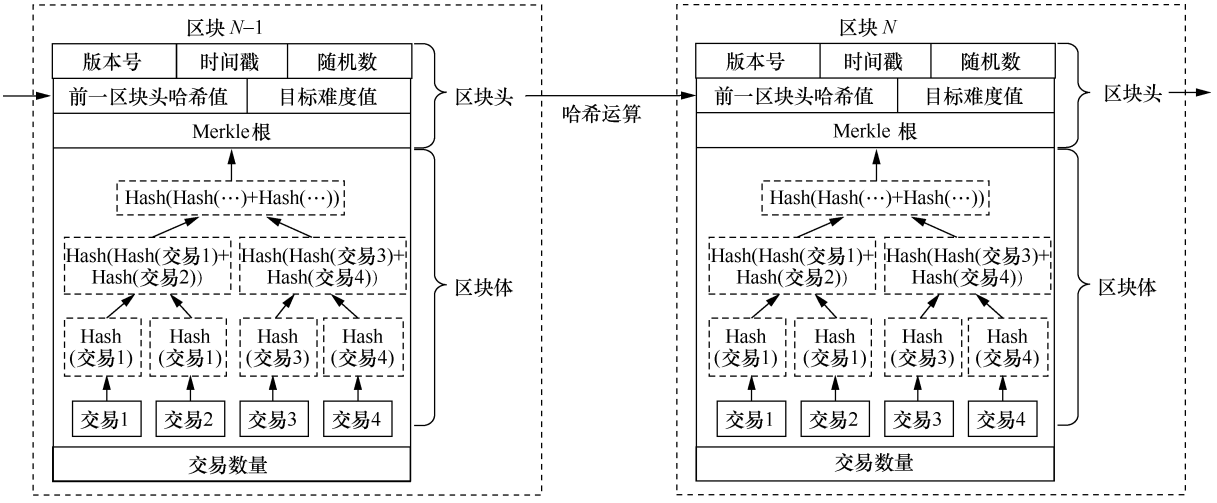


图 2 区块结构示意图

和 Merkle 根等信息；区块体存储了多笔交易信息及交易数量。区块链中的所有交易信息都存储在区块体中，这些交易通过反复迭代进行哈希计算，最终得到一个 Merkle 根。

区块中的各部分含义分别如下。

1) 版本号：验证该区块时使用的协议版本。

2) 时间戳：生成该区块时对应的时间，精确到秒，该时间采用统一的 Unix 时间戳。

3) 随机数：用于证明工作量的随机参数。

4) 前一区块头哈希值：前一区块的区块头的哈希值。

5) 目标难度值：生成该区块时要达成的难度目标。

6) Merkle 根：区块体中的交易通过反复哈希迭代后生成的树形结构的根节点值。

7) 交易：区块链网络中账户之间的转账信息。

8) 交易数量：存储在区块中的交易数量总数。

综上所述，区块链的数据结构运用了大量哈希运算，而哈希运算具有抗碰撞特性，从而保证了数据唯一和不可篡改。在区块体中，每一笔交易都附带发送方的数字签名，以保证交易数据的真实可靠。因此，在数据上链的过程中，依靠交易发送数据可以确认数据的来源，进而防止数据发送方抵赖，保证了数据可追溯。

3.2 区块链的运行流程

区块链系统实质上是一个由交易驱动的状态机，所有状态的改变都是由交易触发的。区块链系统的运行可以分为以下 5 个步骤：1) 区块链中的用户发起转账信息，生成交易数据，然后通过点对点网络将交易信息进行全网广播；2) 网络上的节点通过共识机制产生授权节点；3) 授权节点将网络上未处理的交易信息打包生成新区块；4) 授权节点发起点对点广播，将区块广播给全网节点；5) 其他节点收到区块后进行验证，验证成功后将区块加入区块链末尾，然后开始下一轮运行。

在区块链运行过程中，所有交易信息和区块都经过点对点网络进行广播传输，因此，所有节点都有一个完整的交易账本和状态账本。即使网络中的部分节点下线，但只要有节点正常工作，那么区块链系统就可以正常运行。这不仅实现了数据的共享，而且分布式的数据备份使得数据安全性得到增强。

3.3 智能合约

智能合约最早由 Szabo 于 1996 年提出，其将智能合约定义为数字化的合约条款[31]。智能合约是一段可以自动运行的计算机程序，旨在使用数字化的合约取代现实世界中复杂的交易关系。智能合约的特点是经部署后可自动运行且不可被更改，由于之前缺乏可信的平台，导致智能合约的概念虽然被提出了很多年，却一直没有得到大规模应用。直到比特币出现后，智能合约才开始进入人们的视野。

比特币采用一种称为比特币脚本的代码块实现智能化的转账等简单合同，但是其缺乏图灵完备性，因此，限制了它的推广应用。现在一般提到的智能合约大多是指基于以太坊的智能合约，该智能合约使用图灵完备的 Solidity 语言编写，这种图灵完备的特性进一步扩大了智能合约的应用场景，现阶段已有许多优质的智能合约应用[32-33]。

智能合约完整的生命周期包括 3 个阶段，即合

约生成、合约发布和合约执行。由于合约一经发布则无法篡改，因此，合约的生成必须由多方协定，最终生成的合约将转为程序字节码文件。合约发布过程与交易发布过程类似，合约会随着一笔转账交易广播到全网节点，但是交易的接收方地址为空白，区块链系统根据此空白接收地址判定该交易为合约部署指令，待交易验证后则将合约存储在区块链中。合约发布后会有一个特定的合约地址，该合约地址下存储着合约的字节码文件和状态数据，外部程序可以通过该合约地址与合约进行交互。所有合约都运行在被称为虚拟机的沙盒环境中，虚拟机与外界隔绝，保证了合约完全按照编写的逻辑执行，不受外部环境影响。在合约执行过程中，改变的状态都会进入区块链的状态数据库，供各方查询和验证。

考虑合约开发和部署问题，本系统采用基于以太坊的智能合约。在以太坊中，可以直接使用命令行客户端 Geth 访问虚拟机和数据库，但是能在命令行执行的操作有限，因此，本系统选用 Web3.js 通过 Json-rpc 调用 Geth 提供的接口，进而与虚拟机和数据库进行交互。外部程序与智能合约交互示意图如图 3 所示，其中，State tries 存储各账户的数据；Storage tries 存储智能合约的数据；Transaction tries 存储交易数据。

4　基于区块链的检测系统架构

本系统基于区块链和智能合约构建，按照功能可以划分为物联网模块、共识模块、智能合约模块、区块链模块、检测模块等 5 个模块。1) 物联网模块提供通信网络，供节点之间交换数据；2) 共识模块负责运行平均一致性共识算法，此算法是物联网网络资源分配的共识过程，区别于区块链的用户共识机制，并且在共识完成后通过 Web3.js 接口调用智能合约模块将数据上传到区块链模块；3) 智能合约模块分为存储、查询合约和检测合约，负责数据的存储、查询和计算分析；4) 区块链模块中存储所有数据，实现全网数据共享；5) 检测模块以一定的周期调用检测合约运行恶意节点检测算法，然后根据合约的返回结果动态调整物联网网络拓扑。系统架构如图 4 所示。

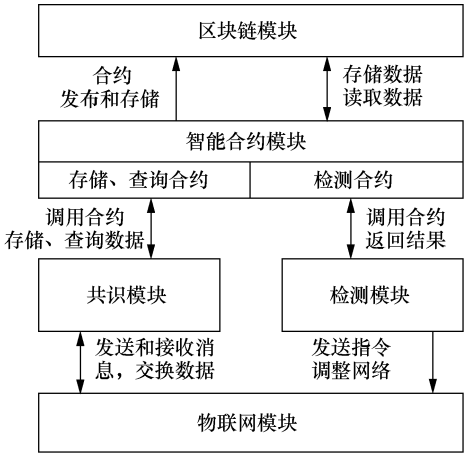


图 4　系统架构

4.1　物联网模块

物联网模块负责通信网络的构建，实现节点之间的连接和通信。物联网中的通信技术根据传输距离可以分为两大类：一类是低功率广域网络（LPWAN, low-power wide-area network）技术，其典型技术包括 Lora（long range）、NB-IoT（narrowband Internet of things）、Sigfox、5G 等；另一类是近距离通信技术，其常见的技术包括 RFID（radio frequency identification）、NFC（near field communication）、Wi-Fi、BLE（bluetooth low energy）、ZigBee、Z-Wave、6LowPAN（IPv6 over low-power wireless personal

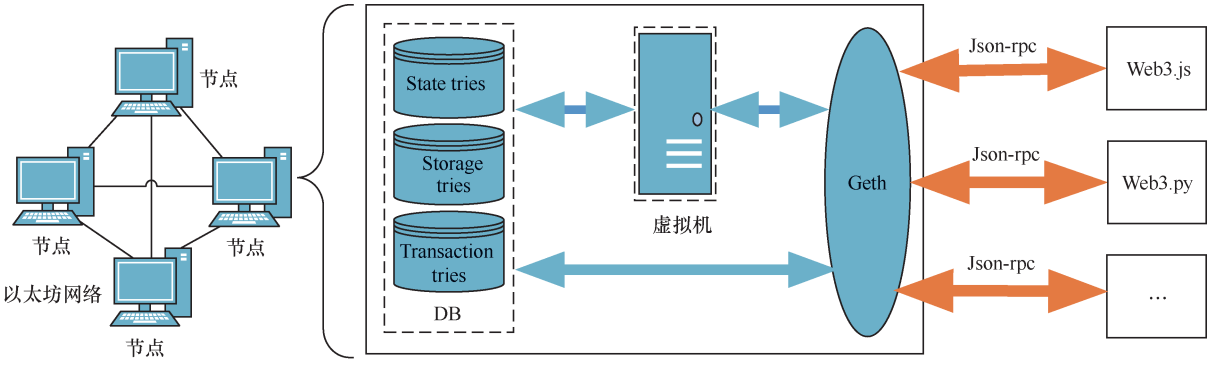


图 3　外部程序与智能合约交互示意图

area networks）等。广域网通信技术的覆盖范围大，但是传输速度普遍较慢，而在近距离通信技术中，大部分技术都专注于低功耗场景，限制了数据交换速率，考虑共识过程中涉及大量的数据交换，本系统采用吞吐量较大的 Wi-Fi 协议搭建底层网络。

4.2　共识模块

共识模块负责在物联网网络内上传节点的本地状态数据并运行共识算法，是系统的数据来源。共识模块工作流程如图 5 所示，每轮共识过程执行的具体步骤如下：1) 在每个周期内，根据泊松过程唤醒节点 i；2) 节点 i 均匀随机地选择一个邻居节点 j，两者建立通信链路；3) 节点 i 和节点 j 互相交换状态，更新自身状态；4) 节点 i 和节点 j 判断是否达成共识，如果达成共识则通过 Web3.js 接口调用存储、查询合约的数据上传函数，将所收集的数据上传；如果未达成共识，则跳至步骤 1 继续循环整个过程。

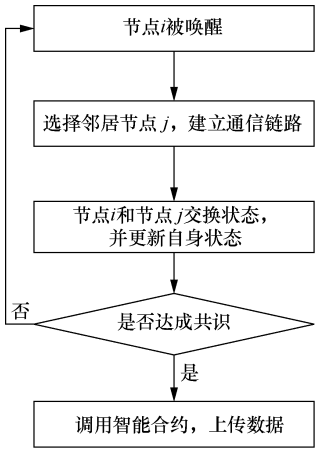


图 5　共识模块工作流程

4.3　智能合约模块

智能合约模块分为存储、查询合约与检测合约。存储、查询合约定义了数据的结构及供共识模块调用的数据存储函数和数据查询函数，存储函数负责将共识模块上传的数据按照定义的数据结构存储至区块链模块，查询函数则按照接收的查询指令查询区块链模块中的数据并将结果返回。检测合约实现了时间差分检测算法和空间差分检测算法，并且提供函数给检测模块进行调用。在调用结束后，返回对应的检测结果。检测算法在运行过程中会访问区块链模块中存储的共识数据，并依赖于共识数据进行恶意节点的检测与定位。

智能合约的执行分为两个过程，包括合约部署和合约执行。合约部署流程为：1) 合约制定后由合约发布者发起一笔对空转账，并将合约字节码附在转账交易上进行全网广播；2) 接收交易的节点会确认交易的合法性，确认合法后区块链模块会生成一个合约地址，合约的内容及数据会存储在该地址字段下；3) 在合约部署完成后，等待被调用。

合约执行流程分为如下 3 个步骤：1) 节点 i 通过 Web3.js 接口发起一笔交易，交易接收方为智能合约的合约存储地址，交易数据字段为被调用函数的 ABI（application binary interface）码和十六进制参数；2) 交易全网广播，接收到交易的节点确认交易合法后在以太坊虚拟机中执行智能合约代码块，在代码执行过程中，所有的状态改变和执行结果都将被永久地记录在区块链中；3) 节点 i 对应的虚拟机通过 Json-rpc 接口返回合约执行结果。

4.4　区块链模块

区块链模块负责存储和读取数据。如图 3 所示，与传统区块链存储模型相比，以太坊新增了一个 Merkle 树类型的 Storage 数据库，该数据库负责存储智能合约上传的数据及状态，通过合约地址可以访问合约的数据空间。在本系统中，共识数据的逻辑存储结构如图 6 所示，该结构采用 Solidity 内置的哈希表存储。

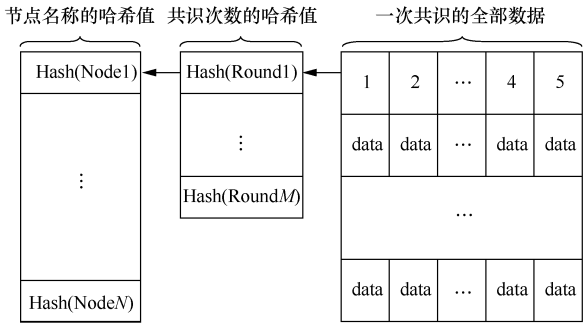


图 6　共识数据的逻辑存储结构

4.5　检测模块

检测模块负责调用智能合约模块中的检测合约，按照检测合约的返回结果动态地调整物联网网络拓扑。检测模块的运行流程为：1) 在一个检测周期中，如果共识次数达到检测条件，则通过 Web3.js 接口发送一笔交易调用检测合约；2) 将检测合约调用区块链模块的数据代入检测函数中，并把检测结果返回给调用者；3) 检测模块根据检测结果执行操作，如果邻居节点中有恶意节点，则断开与恶意节点的连接。

在检测过程中，所有正常节点都会在虚拟机中执行检测合约，整个检测过程被记录在区块链模块中，从而保证检测过程和检测结果公开、透明，检测结果不可篡改。

5　仿真分析

本节将给出仿真结果，以验证本模型的有效性，曼哈顿网络如图 7 所示。在仿真过程中，以图 7 中拥有 9 个节点的曼哈顿网络为例来验证本文提出的检测系统的有效性，更复杂的网络模型（如小世界网络模型）将会在后续的工作中进一步论述。节点 2 为正常节点，其邻居节点有节点 1、节点 3、节点 5 和节点 8，其中，节点 1 是普通节点伪装成的恶意节点，下文所有的数据和操作全部来自节点 2。在仿真过程中，参数设置与文献[8]相同。其中，恶意节点的初始值服从均值为 0、方差为 1 的高斯分布 $\alpha^k \sim N[0,1]$，正常节点的初始值服从均匀分布 $\gamma_i^k \sim U[-0.5,1.5]$，恶意节点添加的随机衰减噪声为 $m_j^k(t) \sim U[-(\lambda_2(\bar{\boldsymbol{W}}))^t,(\lambda_2(\bar{\boldsymbol{W}}))^t]$，$\lambda_2(\bar{\boldsymbol{W}})$ 是 $\bar{\boldsymbol{W}}$ 的第二大特征值。在本节，首先给出了测试网络的收敛性以及时间差分、空间差分的性能，并采用以太坊智能合约测试了节点状态上传模块和节点检测模块的性能。具体为将系统部署在由 9 台树莓派组成的集群中，每个树莓派模拟一个独立的物联网节点，其中的恶意节点由树莓派伪装而成。所有节点都接入以太坊的测试网络中，在共识算法收敛后将数据上传至测试网络。网络中每个正常节点都调用检测模块进行恶意节点检测；在收到检测结果后，正常节点与恶意节点之间的通信连接将被断开，恶意节点被踢出网络。

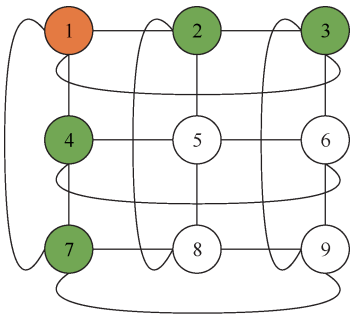


图 7　曼哈顿网络

5.1　分布式网络收敛性分析

网络节点状态收敛变化如图 8 所示，图 8 对比了正常网络与存在恶意节点的网络中节点状态的变化。其中，横轴为迭代次数，纵轴为节点的状态值。图 8(a)的网络中没有恶意节点，所有节点都收敛到网络初始均值 x_{av}^k；图 8(b)的网络中，节点 1 为恶意节点，网络的共识结果被节点 1 引导至 $x_1^k(0)$，偏离了网络初始均值 x_{av}^k。

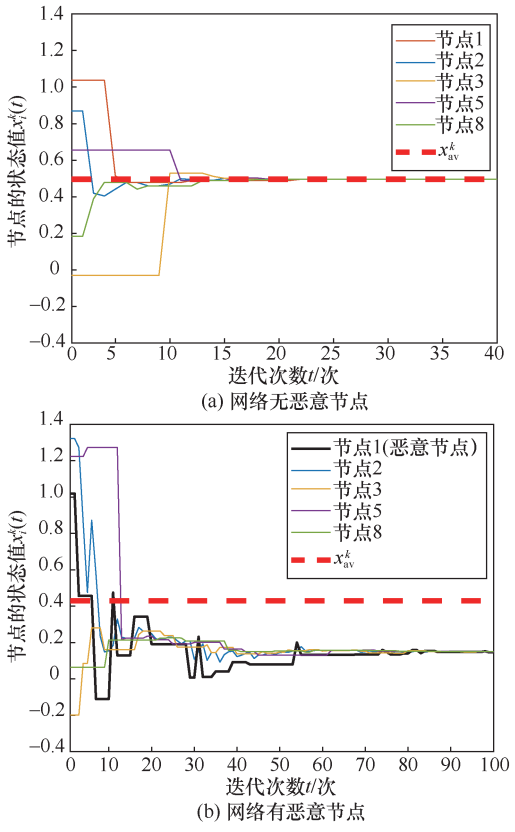


图 8　网络节点状态收敛变化

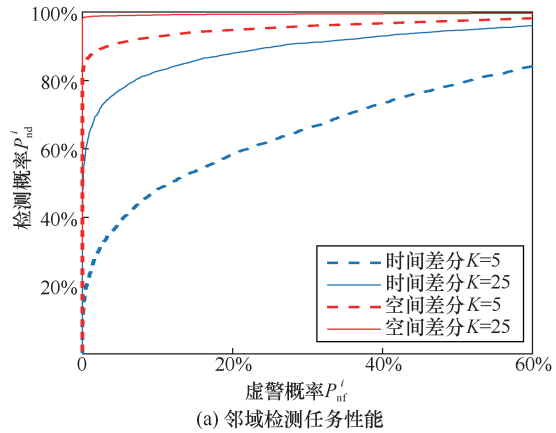
5.2　时间差分和空间差分策略的性能

时间差分和空间差分策略的性能如图 9 所示，在图 9 中，考虑通过受试者工作曲线（ROC, receiver operating characteristics）来对时间差分和空间差分策略的性能进行评估[34]。蒙特卡洛模拟运行试验 2 000 次，采用概率指标对时间差分和空间差分策略进行评估，如式(15)和式(16)所示[9-10]。

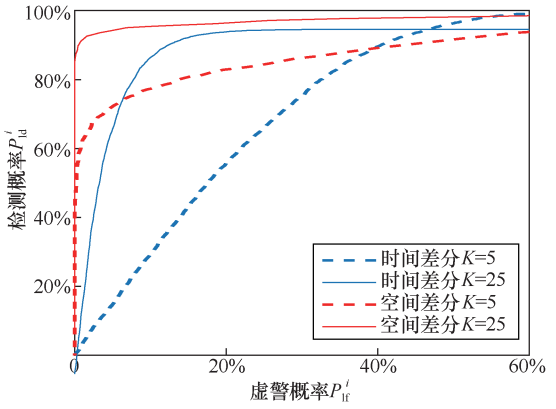
$$P_{\text{nd}}^i := P\left(\hat{H}^i = H_1^i | H_1^i\right), P_{\text{nf}}^i := P\left(\hat{H}^i = H_1^i | H_0^i\right) \tag{15}$$

$$P_{\text{ld}}^i := P\left(\hat{H}^{ij} = H_1^{ij} | H_1^{ij}\right), P_{\text{lf}}^i := P\left(\hat{H}^{ij} = H_1^{ij} | H_0^{ij}\right) \tag{16}$$

式(15)和式(16)中，P_{nd}^i (P_{ld}^i) 和 P_{nf}^i (P_{lf}^i) 分别为邻域检测（定位）任务中的检测概率和虚警概率，相关指标计算如式(17)和式(18)所示。



(a) 邻域检测任务性能



(b) 邻域定位任务性能

图 9 时间差分和空间差分策略的性能

$$P_{\mathrm{nd}}^{i}=\frac{1}{D_p}\sum_{n=1}^{D_p}I(d^{i,n}=\hat{d}^{i,n}=1),$$
$$P_{\mathrm{nf}}^{i}=\frac{1}{D_n}\sum_{n=1}^{D_n}I(d^{i,n}=0\wedge\hat{d}^{i,n}=1) \tag{17}$$

$$P_{\mathrm{ld}}^{i}=\frac{1}{L_p}\sum_{n=1}^{L_p}I(\ell_j^{i,n}=\hat{\ell}_j^{i,n}=1),$$
$$P_{\mathrm{lf}}^{i}=\frac{1}{L_n}\sum_{n=1}^{L_n}I(\ell_j^{i,n}=0\wedge\hat{\ell}_j^{i,n}=1) \tag{18}$$

式(17)中，$D_p(D_n)$ 为检测任务中的正例（负例）样本个数，$I(\cdot)$ 是一个指示函数。$d^{i,n}$ 和 $\hat{d}^{i,n}$ 分别为检测任务中样本的真实类别和预测类别。在式(18)定位任务中，L_p 和 L_n 分别为正例和负例节点个数，$\ell_j^{i,n}$ 和 $\hat{\ell}_j^{i,n}$ 分别为节点的真实类别和预测类别，定位指标采用同样的方式计算。在图 9 中，纵轴为检测概率，横轴为虚警概率。

在图 9 中，时间差分和空间差分策略的检测和定位性能会随着检测实例 K 的增加得到大幅度提升。在 K=25 时，时间差分策略已经可以提供可靠的攻击检测和恶意节点定位性能。与时间差分策略相比，由于空间差分策略考虑更多的瞬时状态信息，所以更有助于提升检测模型的性能，在 K=5 时就能够提供很可靠的攻击检测和恶意节点定位性能。

5.3 基于区块链的节点状态共享

本节介绍了节点在共识完成后基于区块链的状态共享过程。将状态共享过程分为两部分：第一部分为将存储合约部署在区块链上，第二部分为节点调用数据存储合约上传共识数据。数据存储合约部署交易如表 1 所示，节点状态数据上传交易如表 2 所示。在表 1 中，交易发送方（From）为合约部署者，即节点 2；接收方（To）地址为空地址，表示此交易为合约部署交易，合约的字节码文件被放在 Input 字段。合约部署交易经过网络中节点的验证后，区块链系统会根据特定算法分配一个地址存储合约的字节码文件和数据，该地址即为合约地址。

在表 2 中，交易发送方也是节点 2，接收方地址为数据存储合约地址。与合约部署交易不同，合

表 1 数据存储合约部署交易

键	值
Transaction Hash	0x6fa0a65054d5a87c224bd70f611a39fa62f7341bf72ddf1b2dc6cfaabad4ea38
From	0x1c3edeff3e9967295084690d561bdc31fc0099b5
To	StorageData.(constructor)
Gas	417972
Transaction cost	417972
Hash	0x6fa0a65054d5a87c224bd70f611a39fa62f7341bf72ddf1b2dc6cfaabad4ea38
Input	0x6080···0029
Decoded input	[]
Decoded output	[]

表 2　　**节点状态数据上传交易**

键	值
Transaction Hash	0xf557bbec590e6342bf8845e88cb679690a1adc81251249b5762e582e5fe96e2d
From	0x1c3edeff3e9967295084690d561bdc31fc0099b5
To	StorageData.storageData(bytes32,uint256,uint256[])
	0xf49bdcfa72aba6ed35d05e85f8326509138a1259
Gas	219872
Hash	0xf557bbec590e6342bf8845e88cb679690a1adc81251249b5762e582e5fe96e2d
Input	0xcf77…026b
	“bytes32 nodeName”: “0x6e6f646531”
Decoded input	“uint256 round”: “1”,
	“uint256[] data”: [“5933”,“13002”,“1630”,“11271”,“619”]
Decoded output	[]

约调用交易的 Input 字段是合约被调用函数的 ABI 码与输入参数的组合，而 Decoded input 则是 Input 解码后的值。在交易信息经过节点的认证后，上传的数据会被存储到合约地址下如图 6 所示的哈希表中。

考虑智能合约（Solidity 语言）无法存储浮点数[35]，因此，在上传数据前要先将所有状态放大 10 000 倍以保留 4 位小数的精度。然后再将放大的数值进行向下取整，得到实际上传的数据。实际上，放大数据的操作等同于将随机变量 α^k、γ_i^k、$m_j^k(t)$ 放大相同倍数，即 $\alpha^k = 10\,000 \times \alpha^k$，$\gamma_i^k = 10\,000 \times \gamma_i^k$，$m_j^k(t) = 10\,000 \times m_j^k(t)$，然后在共识过程中更新状态时将状态数值向下取整。

5.4　基于智能合约的恶意节点检测

本节介绍了基于智能合约的恶意节点检测过程，整个检测过程可以分为两部分：第一部分为将检测合约部署在区块链上，第二部分为检测模块调用检测合约，并且根据检测结果动态调整物联网网络拓扑。检测合约部署交易如表 3 所示，检测合约调用交易如表 4 所示。检测合约部署交易与数据存储合约部署交易类似，同样地，表 3 对应的合约部署交易在经过节点验证后，区块链系统会给合约分配一个合约地址，合约的字节码文件及数据会存储在该地址下。

在表 4 所表示的交易中，交易的发起者是节点 2，接收地址为检测合约地址，Input 字段由检测函数的 ABI 码和输入参数拼接而成，由于是节点 2 调用，因此，输入参数为 node2 的编码。对比表 2 与表 4 可以发现，表 4 所表示的交易消息有返回值。在返回值 Decoded output 字段的结果为邻居节点的编码列表和检测结果列表，其中，节点 1 被判定为恶意节点，而节点 3、节点 5 和节点 8 被判定为正常节点，与网络状态一致。节点 2 根据返回的检测结果

表 3　　**检测合约部署交易**

键	值
Transaction hash	0x546bcb4a4d69fc183336b6a227e9d02c636433e0c54537ffc3d05f3eb66a8c63
From	0x1c3edeff3e9967295084690d561bdc31fc0099b5
To	Detection.(constructor)
Gas	924166
Transaction cost	924166
Hash	0x546bcb4a4d69fc183336b6a227e9d02c636433e0c54537ffc3d05f3eb66a8c63
Input	0x6080…d029
Decoded input	[]
Decoded output	[]

表 4 检测合约调用交易

键	值
Transaction hash	0xbc390e24901488da9f6788722b62087998e2611ad936deef44f71dac3224bda6
From	0x1c3edeff3e9967295084690d561bdc31fc0099b5
To	Detection.detection(bytes32) 0xef653c4d683b88ff99f7a75040fea3cc6411e690
Gas	215661
Hash	0xbc390e24901488da9f6788722b62087998e2611ad936deef44f71dac3224bda62d
Input	0x53cb13f36e6f646532
Decoded input	“bytes32 nodeName”: “0x6e6f646532”
Decoded output	“0”: “bytes32[]: 0x6e6f646531,0x6e6f646533,0x6e6f646535,0x6e6f646538” “1”: “bool[]: true, false, false, false”

断开与节点 1 的连接，节点 1 的其他邻居节点也会根据检测结果断开与节点 1 的连接，从而将节点 1 踢出网络。

6 结束语

本文提出了一个基于区块链和智能合约的物联网恶意节点检测和定位系统，本系统部署在由 9 台树莓派组成的物联网模拟网络中，所有节点都接入以太坊的测试网络，调用检测模块进行恶意节点检测。根据实验过程可知，所有共识数据都上传至区块链以供所有节点查询和验证，实现了去中心化的共享和异地备份，解决了“数据孤岛”和维护困难问题；相比于本地检测程序，智能合约由所有参与方共同制定，合约发布后存储于区块链中，一经发布则不可篡改，并且在检测过程中所有节点同步运行，保证了检测过程的公开、透明；其次，合约执行过程全部记录在区块链的数据库中，确保了结果可追溯和可验证。

下一步将结合密码学、权限访问等技术进一步完善方案，在保证数据共享的同时，加强隐私保护；此外，也可以考虑将数据迁移至星际文件系统（IPFS, interplanetary file system）等区块链系统中，在数据上传过程中将上传数据改为上传 IPFS 数据地址，从而降低交易费用和存储费用。

参考文献：

[1] 孙玉. 我国物联网产业发展趋势[J]. 物联网学报, 2017, 1(3): 1-5.
SUN Y. Development trend of IoT industry in China[J]. Chinese Journal on Internet of Things, 2017, 1(3): 1-5.

[2] IHS Markit. The Internet of things: a movement, not a market[R]. 2017.

[3] TSITSIKLIS J N. Problems in decentralized decision making and computation[R]. 1984.

[4] DUCHI J C, AGARWAL A, WAINWRIGHT M J. Dual averaging for distributed optimization: convergence analysis and network scaling[J]. IEEE Transactions on Automatic Control, 2011, 57(3): 592-606.

[5] SAYED A H. Adaptation, learning, and optimization over networks[J]. Foundations and Trends in Machine Learning, 2014, 7(4-5): 311-801.

[6] SUNDARAM S, GHARESIFARD B. Consensus-based distributed optimization with malicious nodes[C]//53rd Annual Allerton Conference on Communication, Control, and Computing. IEEE, 2015: 244-249.

[7] WU S X, WAI H T, SCAGLIONE A, et al. Data injection attack on decentralized optimization[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 3644-3648.

[8] GENTZ R, WU S X, WAI H T, et al. Data injection attacks in randomized gossiping[J]. IEEE Transactions on Signal and Information Processing over Networks, 2016, 2(4): 523-538.

[9] GENTZ R, WAI H T, SCAGLIONE A, et al. Detection of data injection attacks in decentralized learning[C]//2015 49th Asilomar Conference on Signals, Systems and Computers. IEEE, 2015: 350-354.

[10] LI G, WU S X, ZHANG S, et al. Neural networks-aided insider attack detection for the average consensus algorithm[J]. IEEE Access, 2020, 8: 51871-51883.

[11] BOLOUKI S, NEDIĆ A, BAŞAR T. On the steady-state range of averaging dynamics[C]//American Control Conference. IEEE, 2016: 6447-6452.

[12] KAILKHURA B, BRAHMA S, VARSHNEY P K. Consensus based detection in the presence of data falsification attacks[J]. arXiv:1504.03413, 2015.

[13] SU L, VAIDYA N. Byzantine multi-agent optimization: part I[J]. arXiv: 1506.04681, 2015.

[14] YAN Q, LI M, JIANG T, et al. Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks[C]//2012 Proceedings IEEE INFOCOM. IEEE, 2012: 900-908.

[15] 谢晋阳, 李平, 谢桂芳. 基于特征节点分析的恶意节点检测算法研究[J]. 计算机工程与科学, 2015, 37(1): 78-83.
XIE J Y, LI P, XIE G F. Study on the malicious nodes detection algorithm based on feature nodes analysis[J]. Computer Engineering & Science, 2015, 37(1): 78-83.

[16] 王欣, 胡平, 景波. 基于度量阈值裁决的 WSN 恶意节点筛选算

法[J]. 计算机工程与设计, 2017, 38(5): 1142-1147, 1172.
WANG X, HU P, JING B. Malicious node filtering algorithm of wireless sensor network based on metric threshold decision[J]. Computer Engineering and Design, 2017, 38(5): 1142-1147, 1172.

[17] 季薇, 李炳星, 郑宝玉. 基于信誉与共识的分布式智能入侵防御方案[J]. 系统工程与电子技术, 2018, 40(3): 665-670.
JI W, LI B X, ZHENG B Y. Distributed international registration intrusion prevention scheme based on reputation and consensus[J]. Systems Engineering and Electronics, 2018, 40(3): 665-670.

[18] BOGNER A. Seeing is understanding: anomaly detection in blockchains with visualized features[C]//International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of International Symposium on Wearable Computers. ACM, 2017: 5-8.

[19] SHINDE K, TODKARI S V. Securing wireless sensor network against pollution attack with block chain[J]. International Journal for Modern Trends in Science and Technology, 2019: 2455-3778.

[20] SILVA B N, KHAN M, HAN K. Internet of things: a comprehensive review of enabling technologies, architecture, and challenges[J]. IETE Technical Review, 2018, 35(2): 205-220.

[21] CUI W, WU S, WANG Y, et al. A gossip-based TDOA distributed localization algorithm for wireless sensor networks[C]//2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation. IEEE, 2013: 783-788.

[22] DIMAKIS A G, KAR S, MOURA J M F, et al. Gossip algorithms for distributed signal processing[J]. Proceedings of the IEEE, 2010, 98(11): 1847-1864.

[23] 吴俊宏, 谢胤喆, 王玥, 等. 基于改进 Gossip 算法的多微网孤岛系统分布式电力交易策略[J]. 现代电力, 2019, 36(2): 88-94.
WU J H, XIE Y Z, WANG Y, et al. Distributed electronics trading strategical off interconnected microgrids in islanding mode based on improved gossip algorithm[J]. Modern Electric Power, 2019, 36(2): 88-94.

[24] DAVIES R. The Internet of things opportunities and challenges[J]. European Parliamentary Research Service, 2015.

[25] BILAL M. A review of Internet of things architecture, technologies and analysis smartphone-based attacks against 3D printers[J]. arXiv: 1708.04560, 2017.

[26] BOYD S, GHOSH A, PRABHAKAR B, et al. Randomized gossip algorithms[J]. IEEE Transactions on Information Theory, 2006, 52(6): 2508-2530.

[27] 中国信息通信研究院. 区块链白皮书(2019)[S]. 2019.
China Academy of Information and Communications Technology. Blockchain white paper (2019)[S]. 2019.

[28] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. 2019.

[29] CACHIN C. Architecture of the hyperledger blockchain fabric[C]// Workshop on Distributed Cryptocurrencies and Consensus Ledgers. 2016, 310: 4.

[30] BUTERIN V. Ethereum white paper: a next generation smart contract & decentralized application platform[J]. First Version, 2014, 53: 1-36.

[31] SZABO N. Smart contracts: building blocks for digital markets[J]. EXTROPY: the Journal of Transhumanist Thought, 1996, 18(16): 2.

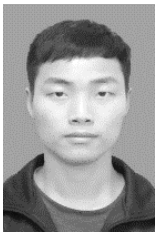
[32] AURORA L. IDEX: a real-time and high-throughput Ethereum smart contract exchange[S]. 2019.

[33] MCCORRY P, SHAHANDASHTI S F, HAO F. A smart contract for boardroom voting with maximum voter privacy[C]//International Conference on Financial Cryptography and Data Security. Springer, 2017: 357-375.

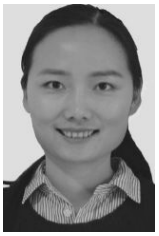
[34] FAWCETT T. An introduction to ROC analysis[J]. Pattern Recognition Letters, 2006, 27(8): 861-874.

[35] Ethereum Foundation. Solidity documentation: Release 0.4.12[R]. 2017.

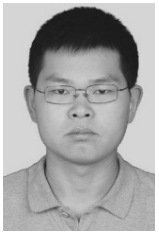
[作者简介]



黄豪杰（1997– ），男，湖北天门人，深圳大学电子与信息工程学院硕士生，主要研究方向为物联网、区块链、数据挖掘等。



吴晓晓（1982– ），女，湖北鄂州人，博士，深圳大学助理教授，主要研究方向为社交网络中的数据挖掘算法、5G 通信网络关键技术研究、信道编码理论等。



李刚强（1991– ），男，河南驻马店人，深圳大学电子与信息工程学院博士生，主要研究方向为社交网络中的数据挖掘算法、分布式协议、机器学习等。