联盟链环境下物联网轻量级网关研究

冯春波[1,2,3],阿不都热衣木江·阿白[4],葛　翔[5],王　轶[1,2,3],程　力[6*]
(1. 中国科学院 新疆理化技术研究所,新疆 乌鲁木齐 830011;
2. 中国科学院大学,北京 100049;
3. 新疆民族语音语言信息处理实验室,新疆 乌鲁木齐 830011;
4. 新疆维吾尔自治区信息中心,新疆 乌鲁木齐 830011;
5. 国网新疆电力公司营销服务中心(资金集约中心、计量中心),新疆 乌鲁木齐 830011;
6. 湖北大学,湖北 武汉 430000)

摘　要:区块链的分布式和去中心化特性能够有效应对传统物联网架构所面临的设备安全和数据安全挑战。网关作为区块链与物联网融合的关键节点,在融合应用中面临算力、存储资源受限的实际困难,亟需可用的轻量级设计与实现方案。针对这一问题,设计并实现了一种基于联盟链的轻量级区块链-物联网网关原型。首先,基于长安链 SPV(Simplified Payment Verification)框架,在网关中实现了关键数据的上链存证与交易数据的过滤精简;其次,从感知设备的行为模式、感知数据两方面进行模式提取与异常识别,保证设备的接入安全和运行安全;最后,针对网关轻节点所存储的区块链默克尔树,提出了一种剪枝算法,加速本组织相关交易数据的验证过程。实验结果表明,设计的轻量级网关具备设备身份可信认证和运行时异常行为检测的可行性,与其它方法相比,默克尔树剪枝优化算法能够大幅度降低交易验证时延。

关键词:区块链;物联网;网关;轻量级节点;数据上链;身份认证
中图分类号:TP393　　**文献标识码**:A　　**文章编号**:1673-629X(2023)12-0128-08
doi:10.3969/j.issn.1673-629X.2023.12.018

Research on Lightweight Gateway of IoT in Consortium Chain Environment

FENG Chun-bo[1,2,3], ABDUREYIM Abai[4], GE Xiang[5], WANG Yi[1,2,3], CHENG Li[6*]
(1. The Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China;
4. Information Center of Xinjiang Uygur Autonomous Region, Urumqi 830011, China;
5. Electric Power Research Institute and the Marketing Service Center, State Grid Xinjiang Electric Power Co, Urumqi 830011, China;
6. Hubei University, Wuhan 430000, China)

Abstract: The distributed and decentralized nature of blockchain technology can effectively address the security challenges faced by traditional Internet of Things (IoT) architectures in terms of device and data security. As a key node in the integration of blockchain and IoT, the gateway faces practical difficulties such as limited computing power and storage resources, requiring lightweight design and implementation solutions. In this paper, a blockchain-based lightweight IoT gateway prototype was designed and implemented to address this problem. Firstly, the key data of the link certificate and transaction are filtered and simplified in the gateway based on ChainMaker SPV (Simplified Payment Verification) framework. Secondly, pattern extraction and anomaly detection were carried out from both the behavior patterns of the perception device and the perception data, ensuring device access security and operational safety. Finally, a

收稿日期:2023-03-09　　**修回日期**:2023-07-12
基金项目:新疆维吾尔自治区重点研发计划项目(2022B01005-1);丝绸之路经济带创新驱动发展试验区、乌昌石国家自出创新示范区科技发展计划项目(2022LQ03003);新疆网信科创课题研究(12221604);中国科学院青年创新促进会项目(2021434)
作者简介:冯春波(1990-),男,硕士研究生,CCF 会员(K0808G),研究方向为区块链、物联网;通信作者:程　力(1973-),男,博士,教授,博导,CCF 会员(34269M),研究方向为区块链、物联网、机器学习、智能系统。

pruning algorithm was proposed for the Merkle tree stored on the gateway´s lightweight node to speed up the verification process for relevant transaction data in the organization. Experimental results show that the designed lightweight gateway has the feasibility of device identity authentication and runtime abnormal behavior detection, and the proposed Merkle tree pruning optimization algorithm can significantly reduce transaction verification latency compared to other methods.

Key words: blockchain; internet of things; gateway; lightweight node; data on-chain; identity authentication

0　引　言

区块链(Blockchain)[1]是一种由多方共同维护的分布式账本技术,具有去中心化、不可篡改、可溯源等特性。区块链技术的去中心化特性与物联网的海量感知网络具有天然的耦合性[2-3],而基于密码学原理的不可篡改性和可溯源性则为物联网设备的身份认证、数据安全和可信存储提供了理论基础和全新的设计思路。

物联网网关是物联网中感知设备和上层应用连接的枢纽,同时也是物联网与区块链融合的关键设备,将区块链技术引入网关,可以增强物联网系统的安全性。然而,网关作为一种资源受限的物联网设备,一般不具备足够的计算和存储资源,无法支撑区块链节点功能的正常运行。其次,在网关中对感知设备进行可靠的身份认证,是保障物联网-区块链融合安全的重要基础。最后,作为一种数据汇集设备,网关运行时持续从感知设备获取感知数据,感知数据包含大量的正常感知数据与少量的异常感知数据。相较于正常感知数据,由于环境异常或设备本身异常而产生的异常感知数据对应用安全和设备安全更具价值。因此,如何在资源受限的物联网网关中,通过区块链技术保障设备接入和运行状态安全,仍是一个值得探究的重要问题。

针对以上问题,该文设计了一种基于联盟链的轻量级区块链-物联网网关原型。首先,基于联盟链的SPV[1](Simplified Payment Verification)架构搭建了轻量级的物联网-区块链网关原型框架,该框架仅存储本组织相关的交易数据;其次,基于感知设备的流量指纹与行为模式进行感知设备运行状态下的可信接入检测;再次,通过高效的机器学习算法识别关键异常感知数据并进行上链可信存储;最后,在资源受限网关的轻节点中对区块头中的默克尔树进行剪枝,进一步降低节点的存储空间需求并大幅度优化关联交易数据的链上验证时间性能。

1　相关工作

近年来,区块链技术的兴起为物联网安全研究开辟了新方向。区块链-物联网融合系统中,在融合架构方面,轻量级的融合架构是当前研究中需要解决的重要问题。B. Bünz 等人[4]提出了基于 NIPoPoWs(Non-Interactive Proofs of Proof-of-Work)的 FlyClient,其只存储本地区块头数据。王思源等人[5]在其工作中引入 SPV 节点和超轻节点的概念,将权能令牌以交易的形式存储在区块链上,并且 SPV 节点只需存储区块头就能完成交易验证,而超轻节点只需维护一个令牌哈希值。H. Yahya 等人[6]提出每个节点随机存储部分区块链数据,并通过鼓励相互验证提高数据可信性。上述工作对区块链节点进行的轻量化改进,在进行交易验证时需要向全节点请求数据,增加了网络中的通信开销。

在感知设备安全认证方面,S. Saha 等人[7]提出了一种方案,即感知设备和网关之间相互认证,同时网关和边缘服务器之间也相互认证。边缘服务器从网关中提取传感数据并添加到区块链中,在该方案中,它能够将传感数据添加到区块链中,从而保证数据的安全性和可靠性,但是感知设备无法与区块链直接建立信任。S. C. Cha 等人[8]提出为网关和每个新增设备分别创建智能合约,将两个智能合约绑定在一起,以实现设备在网关中的认证。沈海波等人[9]和 F. Xu 等人[10]则直接在网关中增加区块链相关功能模块。尽管这些方案都有一定的优点,但它们在智能合约、账本数据等区块链模块在资源受限的传统网关中应用所带来的压力问题方面考虑不足。

在融合框架的数据处理方面,张利华等人[11]针对电力物联网数据的数据存储提出了一种基于联盟链的解决方案。P. Kumar 等人[12]面向智慧城市场景,通过区块链技术保证相关的隐私安全。G. S. Aujla 等人[13]将区块链技术应用于室内健康监测,通过区块链技术进行数据的安全传输。T. H. Pranto 等人[14]和 M. Sultan 等人[15]将区块链技术应用于农产品供应链场景中相关数据的安全存储。上述区块链在物联网中的应用多是将相关的数据全部存储在区块链中,这样物联网系统产生的高频感知数据占用了大量的存储资源。

2　基于联盟链的轻量级网关框架

该文提出了一种基于联盟链的轻量级网关框架,该框架主要由 SPV 模块、边缘计算模块和外部接口模块组成,如图 1 所示。

SPV 模块是一个轻量级的联盟链节点,其中,数据同步功能同步与本组织相关的交易数据,以降低节点的存储压力。默克尔树剪枝功能对 SPV 节点区块头的默克尔树进行剪枝优化,进一步降低节点的存储

空间需求,并大幅降低交易验证速度。本地存储以 LevelDB[16] 为载体将区块链数据存储在网关节点本地,数据服务为其他应用或设备提供交易查询、验证服务;边缘计算模块包括设备行为模式监测和感知数据异常检测两个核心功能,用于感知设备身份合法性判断和筛选出高价值的异常数据;外部接口模块用于支撑网关与 IPFS(Inter Planetary File System)[17]、云服务器和区块链节点等系统或平台进行交互,从而实现分布式存储、大数据分析、智能合约等功能。

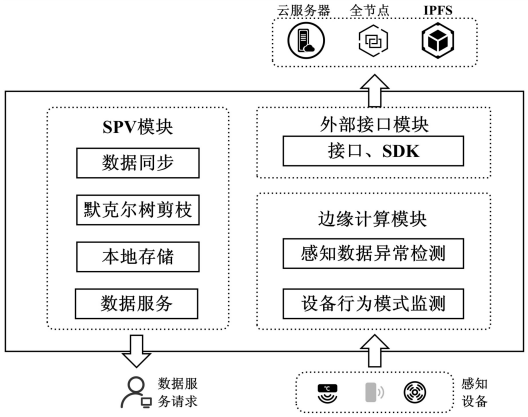


图 1 轻量级网关原型框架

网关从感知设备采集感知数据,并从这些数据中提取感知设备的行为模式数据。网关主要处理这两类数据,其处理流程如图 2 所示。

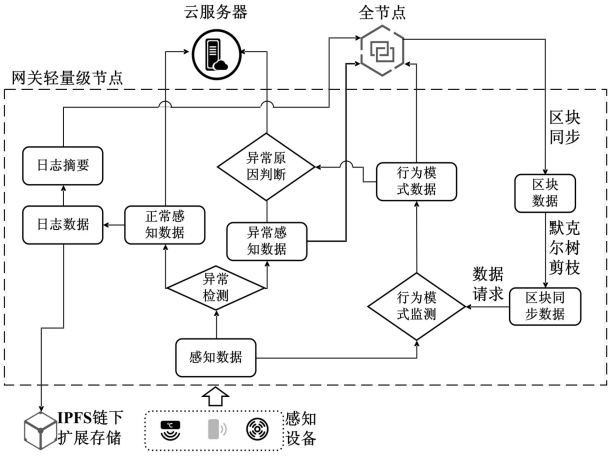


图 2 数据处理流程

网关对设备行为模式数据进行上链存储,并同步存储在网关的 SPV 模块,用于设备的可信接入认证。感知数据经过边缘计算模块中感知数据异常检测功能的处理之后分为正常感知数据和异常感知数据。对于正常感知数据在边缘侧进行聚合形成日志数据,定期存入链下扩展存储—IPFS 中,而日志数据的哈希值存储到区块链上。对于异常感知数据,它们可能对应用的安全或设备的安全产生积极的意义,在边缘侧通过设备的运行状态判断异常产生的原因,然后将设备状态与感知数据一起进行数据上链。

3 轻量级网关核心功能

3.1 设备可信接入

设备可信接入既是保障物联网系统安全的重要手段,也是区块链中数据可信的基础。本团队前期研究工作通过基于感知设备流量指纹[18]的方式实现了设备的可信接入,但是缺少对感知设备在运行时行为合法性的监控。为此,该文进一步在网关中实时采集感知设备的行为模式,并通过 JS 散度(Jensen-Shannon Divergence)衡量感知设备当前的行为模式和链上存储到历史行为模式的差别,进而判断感知设备行为的合法性。这种基于 JS 散度对设备行为进行合法性判断的算法是一种具有敏捷性的轻量化身份认证算法,只需消耗较少的计算资源。

JS 散度是对 KL 散度(Kullback-Leibler Divergence)缺少对称性的改进,可以更好地度量两个概率分布的相似程度。KL 散度和 JS 散度的计算方法如公式:

$$D_{KL}(P \parallel G) = \int_{-\infty}^{+\infty} P(x) \log \frac{P(x)}{G(x)} dx \tag{1}$$

$$D_{JS}(P \parallel G) = \frac{1}{2} D_{KL}\left(P \parallel \frac{P+G}{2}\right) + \frac{1}{2} D_{KL}\left(G \parallel \frac{P+G}{2}\right) \tag{2}$$

公式中,P 表示真实的概率分析,G 表示预期的概率分布;$D_{JS}(P \parallel G)$ 取值范围为[0,1],其取值越小表示两个分布越相似。

用 D 表示一个感知设备,T 表示行为模式监测周期,p 为判断阈值。通过设备的行为模式判断设备行为合法性的步骤为:

步骤 1:初始化阶段,定期统计设备 D 向网关发送数据包的时间间隔,计算其均值和标准差,并使用得到的均值和标准差进行正态分布建模。记得到的正态分布的概率密度为 $f_0(x)$。

步骤 2:设备正常运行时,持续计算网关接收到设备 D 数据包的时间间隔。然后,在每个周期结束计算当前周期网关接收到设备 D 发送的数据包时间间隔的均值和标准差,并将其进行正态分布建模,记其概率密度函数为 $f_t(x)$。

步骤 3:从本地同步的区块数据中获取设备 D 在当前周期之前 3 个周期设备的行为模式,记设备 D 在上述 3 个周期中网关接收到设备 D 发送数据时间间隔的概率密度函数为 $f_1(x)$,$f_2(x)$,$f_3(x)$。

步骤 4:将 $\{f_t(x), f_0(x)\}$,$\{f_t(x), f_1(x)\}$,$\{f_t(x), f_2(x)\}$ 和 $\{f_t(x), f_3(x)\}$ 4 组概率密度函数带入公式,得到 $D_{JS}(f_t \parallel f_0)$,$D_{JS}(f_t \parallel f_1)$,$D_{JS}(f_t \parallel f_2)$,$D_{JS}(f_t \parallel f_3)$。

步骤 5:为 $D_{JS}(f_t \| f_0)$,$D_{JS}(f_t \| f_1)$,$D_{JS}(f_t \| f_2)$ 和 $D_{JS}(f_t \| f_3)$ 分配权重,分别为 0.5,0.25,0.15 和 0.1,计算得到加权 JS 散度 D_{JS}(AVG)。

步骤 6:如果 D_{JS}(AVG) 大于 p,则将结果判定为异常,否则判定结果为正常。

3.2　感知数据异常发现

在生产环境中,网关采集到的感知数据通常包含大量的正常数据和少量的异常数据。由于区块链的存储方式采用了多节点冗余存储的架构,在资源受限的边缘侧尤其需要降低网关节点的存储压力。因此,该文选择只存储高价值的异常感知数据,以减少节点存储空间需求。通过基于 LSTM(Long Short Term Memory)加滑动窗口的方式进行感知数据的异常检测,这种异常检测方法通过 Tensorflow Lite 实现,只需要较少的计算和存储资源。

用 P 和 R 分别表示在 t 时刻感知设备的预测值和实际值,在 t 时刻的误差计算公式为:

$$e_t = \frac{|P - R|}{|R|} \tag{3}$$

异常区间指以两个连续异常为端点所形成的区间。滑动窗口是从历史误差数据中取出的一段连续误差的集合,用于确定当前时刻误差在历史误差中的相对位置。为了具有代表性,滑动窗口应包含一定数量的异常区间。然而,在某个时间段内,异常发生的情况可能比较密集,也可能比较稀疏。因此,为滑动窗口长度设置最小值 minLen 和最大值 maxLen。

感知数据异常检测分为建立模型、确定滑动窗口、异常判断和上链存储四个阶段。

(1)建立模型:用$\{\cdots t_{-2}, t_{-1}, t_0, t_1, t_2 \cdots\}$表示一个时间序列,其中 t_0 表示当前时刻,用 r_0 表示 t_0 时刻网关采集到的实际感知数据,在 t_{-1} 时刻,将 t_0 之前一段时间的感知数据作为 LSTM 模型的输入,可以得到 t_0 时刻的预测值 p_0。将 p_0, r_0 带入公式,可以得到 t_0 时刻的误差 e_0,使用同样的方法可得到 t_0 之前的误差集 E。

(2)确定滑动窗口:用[ws,we]表示一个滑动窗口。用 wn 表示滑动窗口中包含异常区间的数量。设 l_n 为当前时刻之前出现的第 n($n \geqslant 0$)个异常的时间点,则 ws 等于 l_0,we 等于 l_{wn}。当历史异常值的数量小于 wn 时,将 we 的值设为 ws+minLen。

(3)异常判断:计算 E 中数据的平均值(μ)和标准差(σ)。当 e_0 大于 $\mu + 1.96\sigma$ 时,将结果判定为异常,否则判定为正常。

(4)上链存储:当结果判断为异常时,根据网关边缘计算模块中的设备行为模式数据对感知数据异常的来源进行判断。若是因为设备自身而产生的异常,则将异常数据进行简化上链存储。否则,将异常数据进行完整的上链存储。

3.3　默克尔树剪枝

网关作为一种资源受限的设备,与其相连的设备对数据的需求大多为本组织中的交易数据,因此在网关中可以只同步存储与本组织相关的交易数据。

默克尔树剪枝算法的基本思想为:网关轻量级节点中只存储了本组织相关的交易数据,默克尔树剪枝算法在上述基础上实现了在区块头的默克尔树中剪掉与本组织无关交易的哈希值。这进一步简化了默克尔树,并降低了交易验证时的时间消耗。

在默克尔树剪枝过程中,首先计算交易的数量,并添加虚拟交易使交易能够构成一个高度最小的满二叉树。将虚拟交易的交易字段设为空,此时在以虚拟交易生成的节点作为叶子节点的树中,各层的哈希值是一致的,从而,在交易验证阶段不需要多次进行哈希计算,这保证了即使大量增加了虚拟交易,交易验证的工作量不会出现大幅的增加。

对连续多个不属于本组织的交易形成的叶子节点集进行分组,分组的方法为依次从叶子节点集取出能组成最高子树的叶子节点作为一个子集,直到叶子节点集变为空集,然后,用各个子集中的节点作为叶子节点所组成树的根作为该子集中节点的摘要存储在网关本地。具体示例如图 3 所示。

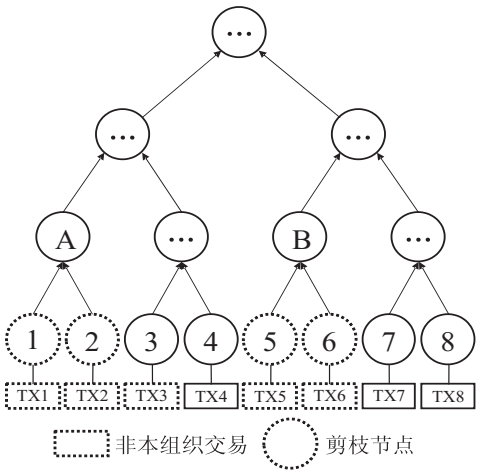


图 3　默克尔树剪枝示例

图 3 中包含 8 个交易(Transaction,TX),默克尔树剪枝时对非本组织的交易{TX1,TX2,TX3,TX5,TX6}作丢弃处理。对相应的叶子节点{1,2,3,5,6},将节点 A 和 B 分别作为叶子节点{1,2}和{5,6}的摘要存储在网关本地。为了保证 TX4 可验证,将叶子节点 3 存储在网关中,最终形成一个以{A,3,4,B,7,8}为叶子节点,以{TX4,TX7,TX8}为交易的简化默克尔树。简化默克尔树能够在节省存储空间的情况下,保证{TX4,TX7,TX8}的可验证性,并加快交易的验证速度。

简化默克尔树中的叶子节点使用起始值和终止值字段标识该节点所代表的原始叶子节点的范围,例如,图 3 中的叶子节点 A 的起始值和终止值字段的数值分别为 1 和 2,节点{4,7,8}的起始值和终止值字段的数值均为其自身的编号。

通过默克尔树剪枝算法,可以降低默克尔树的存储空间,同时,能够加快交易验证效率。默克尔树剪枝算法如下(算法 1):

算法 1:默克尔树剪枝伪代码

输入:交易数据集 TXs,虚拟交易哈希序列 virtualTXHashs,本组织编号 LocOrgID

输出:剪枝后的交易数据集

1. 计算交易组成的 MerkleTree 高度 TXsHeight

2. for mkH:=0; mkH<TXsHeight; TXsHeight++ //mkH 用于标记处理交易树的高度

3. distance := 2^i

4. 定义 tempTXs 用于存储临时交易列表

5. for i:=0;i<Len(TXs); i++

6. if TXs[i]. OrgID! =LocOrgID && TXs[i+1]. OrgID! = LocOrgID

7. if TXs[i]和 TXs[i+1]的叶子节点距离均等于 distance

8. 由 TXs[i]和 TXs[i+1]计算生成哈希值 TxHASH,若其叶子节点为虚拟节点则 TxHASH=virtualTXHashs[mkH]

9. 将 TxHASH 添加到 tempTXs 中,i+=2

10. else

11. 将 TXs[i]添加到 tempTXs 中,i+1=1

12. Return tempTXs //返回剪枝后交易

交易验证的基本思路是根据最新节点的区块头哈希获取上一个区块头,并在其中寻找到指定高度的区块。一旦找到该区块,就从其区块头中获取叶子节点列表。接着,需要遍历叶子节点列表中的哈希值,若目标哈希值在哈希值列表中存在,则可以验证该交易的合法性。

交易合法性的验证流程如下:首先,从区块头中取出叶子节点列表,并确定默克尔树的高度。接下来,按照从下到上的顺序对默克尔树进行逐层验证。在进行每一层验证时,假设当前层高为 n,则交易间隔为 2^n。遍历叶子节点列表中的节点,通过节点中的起始值和终止值确定该节点的交易间隔。当节点的交易间隔为 2^n 时,对该节点及其相邻的后续节点进行哈希计算。依此类推,直到叶子节点列表中只剩下一个节点时,该节点即为交易默克尔树的根节点。交易验证流程的算法伪代码如下(算法 2):

算法 2:交易验证伪代码

输入:交易所在区块高度 blockH,交易哈希值 TxHash,最新区块头哈希 HeadHash

输出:交易是否合法

1. 根据 HeadHash 查询区块,并从中取前区块的 HeadHash,直到查询到目标区块 block

2. merkleLeafNodes := block. MerkleLeafNodes

3. for i := 0; i < Len(merkleLeafNodes); i+=1

4. if merkleLeafNodes[i]. TxHash == TxHash && blockH == blockHeight Then flag = True

5. if flag == False Then return Fale

6. for i:=0; i<$\log_2$(Len(merkleLeafNodes)); i+=1

7. 定义 tempNodes 用于存储临时节点数据

8. for j:=0; j<Len(merkleLeafNodes);

9. if merkleLeafNodes[j]. range == 2^i //2^i 表示当前处理节点对应原始节点的数量

10. TxHash = SHA (merkleLeafNodes[j]. key + (merkleLeafNodes[j+1]. key)

11. 将通过 merkleLeafNodes 第 j,j+1 生成的节点放入 tempNodes,j+=2

12. else

13. 将 merkleLeafNodes[j]放在 tempNodes 中,j+=1

14. merkleLeafNodes = tempNodes

15. root = merkleLeafNodes[0]

16. if block. MerkleRoot ! = root

17. return False

18. return True

4 实验设计与评估

4.1 实验设计

4.1.1 实验硬件基础

实验环境硬件包括 1 台在 VMware ESXi™ 7.0 平台中部署的虚拟机、1 台树莓派 4B 和 1 台 PC,分别在其中部署了长安链 2.1.0 网络、文中的网关系统和感知设备模拟程序。虚拟机硬件资源配置为 CPU:Intel(R) Xeon(R) Gold 6140 CPU @ 2.30 GHz,16 核心,内存 16 GB,操作系统为 CentOS8.2。树莓派 4B 硬件资源配置为 CPU:Broadcom BCM2711 @ 1.5 GHz,4 核心,内存 4 GB,操作系统为 Ubuntu21.04。PC 硬件配置为 CPU:11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40 GHz,4 核心,操作系统为 Windows10。

4.1.2 区块链网络

虚拟机中的长安链网络由 4 个通过 Docker 启动的共识节点组成,网络采用 RAFT 共识算法,单个区块的最大交易数为 100,出块间隔为 2 000 ms。感知数据异常检测滑动窗口的 minLen 和 maxLen 分别为 60 和 200。

4.1.3 数据集

英特尔伯克利研究室(IBRL)传感器数据集为 IBRL 在 2004 年使用 54 个传感器采集到的约 230 万条的感知数据。该文将原始数据按照设备编号分组并按时间顺序排序整合为标准数据集。在标准数据集的基础上为每个设备添加了 5% A. B. Sharma 等人[19]

所定义的异常数据,并将得到的数据集发布在 Github (https://github. com/Trible863/GatewayTestDataset) 上。通过将原始标准数据集进行复制实现增加设备数量到 90 个。取每个设备前 10 000 条数据,其中,前 8 000条作为模型训练集,后 2 000 条用于本实验测试。

4.2　安全性分析

4.2.1　感知设备安全

为了验证设备可信接入功能的有效性,进行了感知设备异常行为检测实验,实验分为正常分组和异常分组。正常分组中各个设备以其自身初始化阶段的行为模式为基准,异常分组中各个设备以其它设备初始化阶段的行为模式为标准模型。之后以 100 秒为周期计算各个设备的 JS 散度值。

实验结果如图 4 所示。

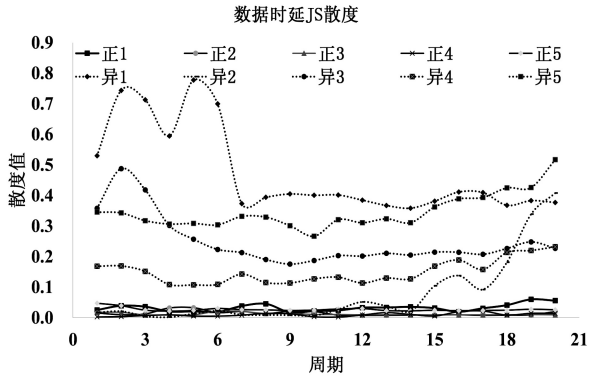


图 4　数据包时延分布 JS 散度

根据实验结果显示,在正常分组中各个设备的数据包时延 JS 散度几乎均小于 0.1,这表明正常设备的数据包时延呈现出一定的相似性和稳定性。相反,在异常分组中,各个设备的数据包时延 JS 散度几乎均大于 0.1,这意味着异常设备的数据包时延具有更大的差异性和不确定性。综合分析可以得出结论,将判断阈值设为 0.1 时,通过设备行为模式对设备行为进行监测能够有效地识别设备的正常行为和异常行为。因此,该感知设备异常检测方案不仅能够帮助检测感知设备的异常行为,而且还能够增强物联网系统的安全性。该方案可以在现有的物联网系统中应用,以提高物联网系统的可靠性和安全性。

4.2.2　感知数据安全

感知数据异常检测能够从大量低质量的重复数据中筛选出有价值的异常数据,该文采用了 LSTM 和滑动窗口方法对感知数据进行异常检测。在提出的预测方法中进行了 3 000 次的预测,图 5 显示了将预测误差按 1‰的组距分组统计的结果。

从实验结果看,随着误差率的增加,分组统计和累积比重逐渐增加。同时,当预测误差在 0.9% 以下时,占比达到了 97.8% 。这表明该预测模型的效果能够满足生产环境中的异常检测需求。

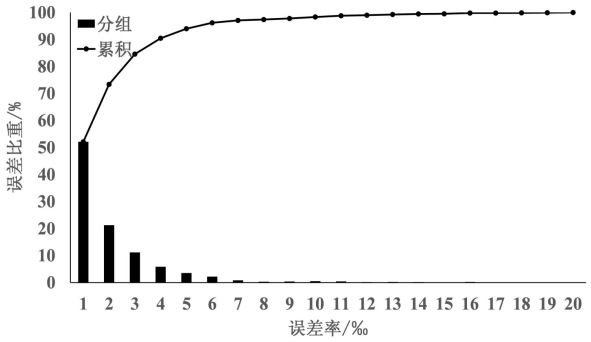


图 5　模型预测性能

4.3　轻量级节点分析

4.3.1　节点运行时延分析

为了分析在网关中采用轻量级节点和在长安链全节点中进行数据上链和交易查询的时延情况,分别在网关节点和长安链节点中进行了数据上链和交易验证的实验。实验分 9 组进行测试,模拟感知设备的数量分别为{10,20,30,40,50,60,70,80,90},实验结果如图 6 所示,图 6 中“本地”和“节点”分别指在网关节点和长安链中进行实验得到的实验结果。

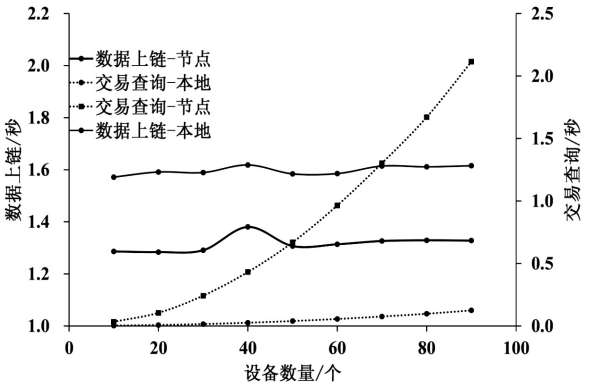


图 6　网关运行时延

实验结果显示:在数据上链时延方面,在网关节点中和在长安链节点中的时延几乎一致,说明在网关中进行数据上链不会明显增加时延开销。在交易验证时延方面,在网关中向网关节点进行交易验证请求相较于向长安链节点进行交易验证请求,能够大幅度降低交易验证时延开销。

4.3.2　默克尔树剪枝分析

为了验证文中默克尔树剪枝方案的优势,进行了区块节点压缩测试、交易验证和默克尔树验证实验,对比方案为 K. Saito 等人[20] 提出的轻量级节点方案(K. Saito 方案),将 K. Saito 方案通过代码实现进行相关实验。

为了测试默克尔树剪枝方案相较于 K. Saito 方案在节点压缩上的优势,首先,根据不同的组织数量随机生成了 5 000 个交易,并模拟 K. Saito 方案区块中不

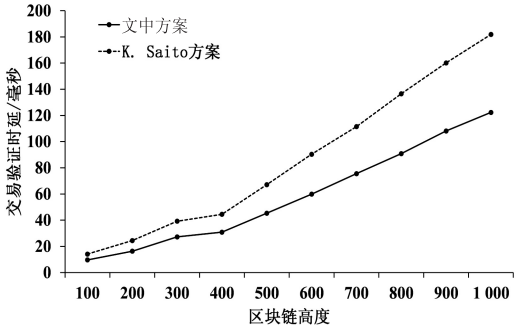
同的交易数随机生成 2 000 个区块。然后,采用默克尔树剪枝算法对区块中默克尔树进行了剪枝,并计算默克尔树剪枝后叶子节点的平均数量。表 1 显示了文中方案相较于 K. Saito 方案(表 1 中基线方案)在叶子节点压缩上的效果。

表 1 默克尔树剪枝结果

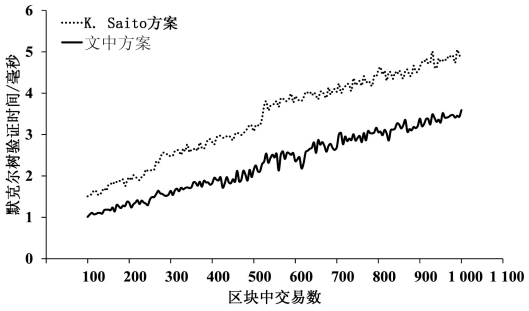
基线方案区块中叶子节点数	组织数量			
	4	6	8	10
	剪枝后区块中叶子节点数			
100	64.85	51.71	43.78	38.5
200	127.7	102.75	85.85	74.5
300	191.63	153.31	128.56	112.78
400	253.41	201.99	170.48	148.40
500	314.23	251.44	211.21	182.65
600	380.02	303.56	255.39	222.68
700	442.29	352.33	297.58	257.84
800	503.83	403.18	339.82	294.93
900	568.88	452.85	379.89	330.06
1 000	627.71	500.63	420.27	365.28

从实验结果看,当组织数量分别为 4,6,8 和 10 时,文中方案区块中的叶子节点数相较于 K. Saito 方案分别降低了 36.6%,49.3%,57.4% 和 62.9%。这表明提出的默克尔树剪枝算法能够显著降低区块中的叶子节点数。

将组织数设为 4,进行交易验证时延和默克尔树验证时延测试。在进行交易验证时延实验时,将区块中交易数设为 100,然后,在不同的区块高度情况下测试了 5 000 次交易验证时延的情况并计算其均值,实验结果如图 7(a)所示。进行默克尔树验证时延实验时,在默克尔树中包含一定交易数的情况下,随机生成了 5 000 个默克尔树。然后,测试默克尔树的交易验证时延,并计算默克尔树交易验证时延的均值,实验结果如图 7(b)所示。



(a)交易验证时延



(b)默克尔树验证时延

图 7 验证时延

从图 7 可以看出,文中方案相比于 K. Saito 方案,具有更优秀的性能表现。在图 7(a)中,文中方案的交易验证时延均小于 K. Saito 方案的交易验证时延,而且总体上比 K. Saito 方案优化了 32.3%。这表明文中方案可以更快速地完成交易验证,从而提高了整个交易处理过程的效率。在图 7(b)中,文中方案的验证时延相较于 K. Saito 方案明显降低。综上,文中方案可以被广泛应用于区块链轻量级节点技术的实践中,提高区块链系统的整体性能和效率。

5 结束语

该文提出了一种基于联盟链的轻量级区块链-物联网网关模型,在资源受限的设备上实现了物联网设备的可信接入、感知数据与设备的快速异常发现以及高效的数据处理,为区块链技术在物联网场景中的融合应用提供了可行的方案。

未来的工作将基于提出的新型智能物联体系架构,结合实际的物联网场景和感知数据类型,进一步优化提出的区块链网关系统。特别是,在区块链默克尔树中交易排序方面进行研究,以更好地优化轻量级节点的存储需求,实现轻量级、高可信和自适应的特性。这将有助于提升物联网系统的安全性和效率,并为实际应用场景提供更好的服务。

参考文献:

[1] NAKAMOTO S. Bitcoin:a peer-to-peer electronic cash system[EB/OL]. 2008[2019-11-13]. https://bitcoin.org/bitcoin.pdf.

[2] REYNA A,MARTÍN C,CHEN J,et al. On blockchain and its integration with IoT. challenges and opportunities[J]. Future Generation Computer Systems,2018,88:173-190.

[3] KHAN M A,SALAH K. IoT security:review,blockchain solutions,and open challenges[J]. Future Generation Computer Systems,2018,82:395-411.

[4] BÜNZ B,KIFFER L,LUU L,et al. Flyclient:super-light clients for cryptocurrencies[C]//2020 IEEE symposium on security and privacy. San Francisco:IEEE,2020:928-946.

[5] 王思源,邹仕洪. 多域物联网中基于区块链和权能的访问控制机制[J]. 应用科学学报,2021,39(1):55-69.

[6] HASSANZADEH-NAZARABADI Y,KÜPÇÜ A,ÖZKA - SAPÖ. Lightchain:scalable dht-based blockchain[J]. IEEE Transactions on Parallel and Distributed Systems,2021,32(10):2582-2593.

[7] SAHA S,CHATTARAJ D,BERA B,et al. Consortium blockchain - enabled access control mechanism in edge computing based generic Internet of Things environment[J]. Transactions on Emerging Telecommunications Technologies,2021,32(6):e3995.

[8] CHA Shi-Cho,TSAI Tsung-Ying,PENG Wei-Ching,et al. Privacy-aware and blockchain connected gateways for users to access legacy IoT devices[C]//2017 IEEE 6th global conference on consumer electronics. Nagoya:IEEE,2017:1-3.

[9] 沈海波,陈　强,黄　海. 面向物联网的基于智能合约的认证和授权方案[J]. 计算机应用与软件,2020,37(1):309-313.

[10] XU Fangmin,YANG Fan,ZHAO Chenglin,et al. Edge computing and caching based blockchain IoT network[C]//2018 1st IEEE international conference on hot information-centric networking (HotICN 2018). Zhenzhen:IEEE,2018:238-239.

[11] 张利华,张赣哲,曹　宇,等. 基于联盟链的泛在电力物联网数据存储方案[J]. 计算机工程与设计,2022,43(9):2415-2422.

[12] KUMAR P,KUMAR R,SRIVASTAVA G,et al. PPSF:a privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities[J]. IEEE Transactions on Network Science and Engineering,2021,8(3):2326-2341.

[13] AUJLA G S,JINDAL A. A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring[J]. IEEE Journal on Selected Areas in Communications,2020,39(2):491-499.

[14] PRANTO T H,NOMAN A A,MAHMUD A,et al. Blockchain and smart contract for IoT enabled smart agriculture[J]. PeerJ Computer Science,2021,7:e407.

[15] SULTAN M. Agriculture-food supply chain management based on blockchain and IoT:a narrative on enterprise blockchain interoperability[J]. Agriculture,2021,12(1):1-25.

[16] CHANG F,DEAN J,GHEMAWAT S,et al. Bigtable:a distributed storage system for structured data[J]. ACM Transactions on Computer Systems,2008,26(2):1-26.

[17] BENET J. IPFS-content addressed,versioned,P2P file system (DRAFT 3)[J]. arXiv:1407.3561, 2014.

[18] GONG Liangqin,ALGHAZZAWI D M,CHENG Li. BCoT sentry:a blockchain-based identity authentication framework for IoT devices[J]. Information,2021,12(5):1-20.

[19] SHARMA A B,GOLUBCHIK L,GOVINDAN R. Sensor faults:detection methods and prevalence in real-world datasets[J]. ACM Transactions on Sensor Networks,2010,6(3):1-39.

[20] SAITO K,WATANABE S. Lightweight selective disclosure for verifiable documents on blockchain[J]. ICT Express,2021,7(3):290-294.