基于区块链的对等网络信任模型

巫岱玥 李 强 余 祥 黄 郡
(国防科技大学 合肥 230037)

摘 要 目前,在信任模型的信任评估过程中,评价数据的来源不统一,使得不同节点获取评价数据的能力不同,不同节点对数据的认可度也不同,从而导致计算结果精度不高且较为主观,难以作为参考。针对此问题,提出基于区块链的对等网络信任模型 ChainTrust。首先,定义评价序列图,根据评估节点在网络中间接信任度的可靠程度来确定间接信任度的权重。同时,改进已有区块链结构,使用 Merkle Patricia 树和二叉 Merkle 树对评价数据进行存储,进一步提高评价数据的安全性,并给出对应的存储、读取算法。仿真与分析结果表明,ChainTrust 能较好地抵御恶意攻击,有效降低共谋攻击对信任评估带来的影响,并能通过调整模型参数改变模型的敏感程度。因此,ChainTrust 模型是有效的,且具有较高的灵活性和普适性。

关键词 信任评估,评价数据,区块链,Merkle Patricia 树,评价序列图,共谋攻击
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/jsjkx.181202307

Trust Model for P2P Based on Blockchain

WU Dai-yue LI Qiang YU Xiang HUANG Jun
(National University of Defense Technology, Hefei 230037, China)

Abstract At present, in the process of trust evaluation of trust model, because the sources of evaluation data are not uniform, the ability of different nodes to obtain evaluation data is different, and the recognition degree of different nodes to data is also different, the computational results are low accuracy, subjective and difficult to be used as a reference. Aiming at these problems, this paper proposed a blockchain-based peer-to-peer network trust model, named ChainTrust. The evaluation sequence graph is defined. The indirect trust weight is determined according to the reliability of the indirect trust degree of the evaluation node. Meanwhile, this paper improved the current blockchain structure, by using the Merkle Patricia tree and the binary Merkle tree to store the evaluation data, and gave the corresponding storing and reading algorithms. Simulation and analysis results show that ChainTrust can better resist a variety of malicious attacks, thus reducing the impact from the collusion attack, changing the sensitivity of the model by adjusting the model parameters. Therefore, ChainTrust is effective and has high flexibility and universality.

Keywords Trust model, Evaluation data, Blockchain, Merkle patricia tree, Evaluation sequence graph, Collusion attack

1 引言

近年来,对等网络(Peer-to-Peer,P2P)以其去中心化、灵活、动态的特点,被应用到越来越多的场景中,为大规模系统的扩展提供了解决方法,提高了系统的资源利用率和灵活性。但是,其具有的开放、动态和自治特性,使得网络中存在大量欺诈行为和不可靠服务,增加了网络的脆弱性。因此,在 P2P 网络中建立有效的信任模型,对 P2P 网络的健康发展十分有必要。

在目前的信任模型中,EigenTrust[1] 模型通过迭代节点间的推荐度来计算目标节点的全局信任。EigenTrust++[2] 模型对 EigenTrust 模型的信任计算方法进行了改进,使用信任相似度来区分正常推荐和恶意推荐。文献[3-7]提出了信任可靠度,指出信任可靠度会影响信任度的计算,但是这些模型默认高信任度的节点拥有高可靠度,这样虽然提高了信任计算的准确性,但是当节点面临诋毁或夸大时,模型的准确性会受到严重影响。

文献[8]提出了交易信息参数、推荐信息的评级可信度和动态平衡权值参数来描述节点的综合信任度,减少了网络中的自私行为和恶意行为,但其没有研究直接信任权重和推荐信任权重的取值,需要专家系统对这两个权重进行指定。

文献[9-10]指出正面评价与负面评价的次数和出现顺序会同时影响信任计算的准确度,但其信任计算的权重依靠专家系统,面对诋毁和夸大攻击时,模型的准确性会受到影响。

到稿日期:2018-12-14 返修日期:2019-04-26 本文受国防科技大学科研基金项目(KYJ2017J351)资助。
巫岱玥(1994-),男,主要研究方向为区块链、信息安全;李 强(1962-),男,教授,硕士生导师,主要研究方向为区块链、软件工程、信息安全,E-mail:lychfeei@163.com(通信作者);余 祥(1986-),男,博士生,讲师,主要研究方向为信息安全;黄 郡(1983-),男,博士,讲师,主要研究方向为信息安全。

文献[11]根据信任可靠度确定信任计算时的权重,然而该模型仅使用了本地数据,且需要推荐节点的可靠度,在实际运用中存在本地存储的数据不足和数据准确性较低的问题。

除这些问题外,信任模型[12-14]还面临着数据来源不统一的问题,这使得对等网络中不同节点对数据的获取能力不同,不同节点对数据的认可度也不同,并且缺少一个统一的或被所有节点都认可的数据来源。因此,即使采用相同的信任模型,计算出的信任度也存在偏差,使得计算出的结果趋于主观,从而失去参考价值,尤其在需要其余节点的信任度进行信任推荐的场合。

区块链[15-17]具有不可抵赖、不可篡改、时序数据、集体维护的特点,通过区块链存储评价数据不仅能保证评价数据的真实性,而且为评价数据增加了时间维度。同时,区块链的集体维护使得全网任意节点都能读取区块链中的评价数据,增大了节点能获取的数据量。更为重要的是,区块链中的数据被全网所有节点所认同,解决了数据来源不统一的问题,为信任模型提供了统一的数据来源。

基于上述分析,本文提出了一种基于区块链的信任模型 ChainTrust,研究了评价数据在区块链中的存取;考虑评价发生的顺序以及评价发出的对象,提出了评价序列和评价序列图的概念,进而根据评价序列计算直接信任度和间接信任度,并依据间接信任度的可靠程度来计算间接信任度的权重,最终得到节点的综合信任度。

本文第 2 节提出一种基于区块链的信任模型 ChainTrust,详细叙述节点的信任计算过程,并给出利用区块链进行数据存储的方法;第 3 节描述模型针对恶意行为的抵御策略,从理论上分析模型对恶意行为的抵御情况;第 4 节给出仿真与分析,通过仿真对模型的各项指标进行分析比较,验证模型抵御恶意节点的能力;最后总结本文工作并指出今后的工作方向。

2　基于区块链的信任模型 ChainTrust

2.1　基本定义

定义 1(实体)　网络中的一个可交互个体。

定义 2(信任度)　表征实体可信程度的数值。

定义 3(信任阈值)　判定实体能否在网络中进行交互的门限。

定义 4(评价)　在某次交互过程中,交互方根据此次交互结果,对另一方满意情况的记录。评价可表示为五元组 $M=(P,I,T,a,w)$。其中,节点集合 $P=\{p_1,p_2,\cdots,p_n\}$是网络中所有实体的集合;交互集合 $I=\{i_1,i_2,\cdots,i_n\}$是网络中所有交互记录的集合;时间集合 $T=\{t_1,t_2,\cdots,t_n\}$是网络中所有交互时刻的集;映射函数 $a:I\times T\to P\times P$,将交互记录与交互时间映射到交互节点上;满意度函数 $w:I\times P\to\{-1,1\}$,用来描述节点对此次交互的满意情况。若满意,$w(i,p)=1$,则为此次交互给出正面评价;若不满意,$w(i,p)=-1$,为此次交互给出负面评价。

定义 5(评价序列)　若干评价按照产生的先后顺利排列形成的有序集合,记为 L。

定义 6(评价电平)　评价序列在坐标轴上的反映,横坐标轴为评价发生的先后顺序,纵坐标轴为满意情况。

定义 7(周期)　评价电平上每两次跳变为一个周期,记作 T。在一个周期内,评价电平仅发生一次跳变。一个周期上的评价记为元组$(\overrightarrow{V^+},\overrightarrow{V^-})$,其中$\overrightarrow{V^+}$表示周期中正面评价的数目及所在位置,$|\overrightarrow{V^+}|$是周期中正面评价的数目,$|\overrightarrow{V^-}|$是周期中负面评价的数目。$\overrightarrow{V^+}$的正负则说明高电平位于前半周期还是后半周期,若一个周期中$\overrightarrow{V^+}>0$且$\overrightarrow{V^-}<0$,则说明该周期的评价电平由高向低跳变,反之则由低向高跳变。

定义 8(摇摆频率)　评价序列 L 中电平跳变的次数,记作 f_c。一般来说,$f_c=2n-1$,n 是评价序列的周期。

图 1 给出评价序列 $L=\{+1,-1,+1,+1,-1,+1,-1\}$的评价电平,该电平有 5 次跳变,该评价序列有 3 个周期,摇摆频率为 5。

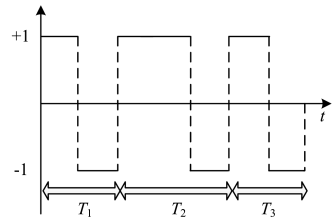

图 1　信任电平

Fig. 1　Trust level

定义 9(评价序列图)　用于描述节点间的评价序列的图,表示为 $G=\{V,E,W\}$。其中,$V:=\{v_p:p\in P\}$,为结点集合,对应网络中的节点。$E:=\{(v_p,v_q):i\in I',t\in T':a(i,t)=(p,q)\}$,为有向边集合,$I'$是给定区块链或区块链片段中的交互记录,$T'$是给定区块链或区块链片段中的时间戳。$W((v_p,v_q)):=L_{p\times q}$,为边权重集合。边权重是给定区块链或区块链片段中节点对另一节点的评价序列。

图 2 所示为评价序列图,图中结点代表网络中的节点,两结点间的有向边代表节点有过评价,有向边的权重代表评价序列。如图中从结点 p 到结点 q 的有向边,其权重为 $L_{p\times q}$,$L_{p\times q}$是 p 对 q 发起过的评价所组成的评价序列。称所有与实体 p 有过交互并对 p 进行过评价的实体为来源实体,在评价序列图中表现为来源实体代表的结点均存在有向线段指向实体 p 代表的结点。

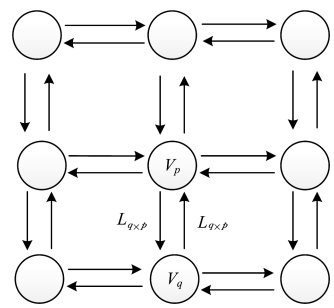

图 2　评价序列图

Fig. 2　Evaluation sequence graph

2.2　信任计算

当网络中节点 i 要了解节点 j 的信任度时,需进行信任计算。信任计算的过程是:节点 i 首先获取节点 j 所有参与评价的记录,评价记录中的被评价方为节点 j,评价方是曾与

j 有过合作的节点，即节点 j 的来源实体；节点 j 将该评价记录按照评价发起者分为若干评价序列，即 $L_{1\times j}$，$L_{2\times j}$，…，$L_{n\times j}$，分别计算每一条评价序列，得到每一个评价发起者对 j 的直接信任度，将直接信任度进行加权得到间接信任度，最后得到节点 i 对节点 j 的综合信任度。

定义 10（周期收益） 在实体 i 对实体 j 的第 k 个评价周期 $T_{ij}(k)$ 内，正面评价和负面评价的加权代数和称为周期收益，记作：

$$P_{ij}(k)=|\overrightarrow{V^{+}}|-\gamma|\overrightarrow{V^{-}}| \tag{1}$$

其中，γ 为惩罚系数，若 $\gamma=1$，意味着一个恶意服务带来的损失可以用一个良好服务去抵消。γ 越大，说明一次恶意服务带来的损失需要越多的良好服务才能抵消。

定义 11（序列收益） 评价序列 L 包含的周期收益的加权代数和称为评价序列 L 的序列收益，记作：

$$I_{ij}=\frac{1}{f_c+1}\sum_{k=1}^{n}\rho^{n-k}P_{ij}(k),\rho\in(0,1] \tag{2}$$

其中，f_c 是评价序列的摇摆频率，$\frac{1}{f_c+1}$ 是节点的摇摆系数。可见，若一个序列中电平频繁发生跳变，则摇摆频率 f_c 较大，而系数 $\frac{1}{f_c+1}$ 较小。常数 $\rho\in(0,1]$ 为时间衰减因子，ρ 越小表示过去评价对序列收益计算所带来的影响越小。

定义 12（评价满意率） 实体 i 对实体 j 所有评价的总体满意程度。

若评价序列 $L_{i\times j}$ 有 n 个周期，则对于序列 $L_{i\times j}$，实体 i 对实体 j 的评价满意率为：

$$A_{ij}=\begin{cases}\dfrac{I_{ij}}{\sum_{k=1}^{n}\rho^{n-k}|T_k|}, & I_{ij}>0 \text{ and } \sum_{k=1}^{n}\rho^{n-k}|T_k|>0\\ 0, & \text{other}\end{cases} \tag{3}$$

其中，$|T_k|=|\overrightarrow{V_k^{+}}|+|\overrightarrow{V_k^{-}}|$，$k=(1,2,\cdots,n)$，$|T_k|$ 是第 k 周期的总评价数，$\sum_{k=1}^{n}\rho^{n-k}|T_k|$ 是当 $L_{i\times j}$ 序列中所有评价都是正面评价时的序列收益，I_{ij} 是当前序列的序列收益。因此，评价满意率的物理含义是现实收益与理想收益的比值。而评价是由评价方根据交互情况给出的，评价序列是若干次评价形成的序列，评价满意率可以认为是评价方对其与被评价方交互的满意程度，反映了评价方对与被评价方交互的信任程度。

由式(3)可知，若节点 j 是诚实节点，节点 j 的评价都是正面评价，则 $f_c=0$，$A_{ij}=1$。若节点 j 是恶意节点，节点 j 的评价都是负面评价，$I_{ij}<0$，则 $A_{ij}=0$。若节点 j 是恶意节点，进行摇摆攻击，则 $f_c>0$，$I_{ij}<\sum_{k=1}^{n}\rho^{n-k}|T_k|$，$0<A_{ij}<1$，且摇摆次数越多，评价满意率 A_{ij} 越低。

定义 13（直接信任度） 评估节点根据自身交互历史对被评估节点的信任程度。

节点间交互的满意程度决定了节点间的信任程度。实体 i 对实体 j 的直接信任度 DT_{ij} 为实体 i 对实体 j 的评价满意率 A_{ij}：

$$DT_{ij}=A_{ij} \tag{4}$$

定义 14（间接信任度） 除评估主体节点外，对被评估节点有过评价的节点对被评估节点的直接信任度的加权和为被评估节点的间接信任度：

$$IT_{ij}=\sum_{k=1,k\neq i}^{n}\frac{|L_{k\times j}|}{N}DT_{kj} \tag{5}$$

其中，$N=\sum_{k=1,k\neq i}^{n}|L_{k\times j}|$ 是除评估主体节点给的评价外，被评估节点获得的评价总数，$|L_{k\times j}|$ 是单个节点给出的评价数目。可见，给出评价越多的节点在计算间接信任度时的权重越高。

定义 15（综合信任度） 直接信任度与间接信任度的加权代数和。

信任评估主体通过自身经验得到直接信任度 DT_{ij}，通过聚合间接信任度 IT_{ij}，得到节点 j 对节点 i 的综合信任度：

$$R_{ij}=\max\{(1-\mu)DT_{ij}+\mu IT_{ij},\delta\},\sigma>0 \tag{6}$$

其中，μ 为间接信任度的权重，代表信任评估主体对间接信任度的信任程度。当 $\mu=0$ 时，评估主体仅信任自身经验；当 $\mu=1$ 时，评估主体仅信任他人的经验。δ 是节点的初始综合信任度，当节点新加入网络时没有任何评价，其信任度为 δ。

2.3 间接信任度的权重计算

在已有信任模型中，间接信任度权重 μ 的取值是定值，一般由专家系统给出，或直接取 0.5，不能根据现实情况进行动态调节，难以抵抗复杂的共谋攻击。ChainTrust 模型提出一种计算间接信任度的权重 μ 的方法，即通过评估节点在网络中间接信任度的可靠程度来计算 μ 的取值，以提高信任评估的准确度，降低共谋攻击带来的危害。其基本思想是：若间接信任度可靠，则 μ 取较大值；若计算间接信任度不可靠，则 μ 取较小值。

假设实体 q 的来源实体集合为 $P_q=\{p_1,p_2,\cdots,p_n\}$，则来源实体 q 对被评价节点的评价次数集合为 $S=\{|L_{p_1\times q}|,|L_{p_2\times q}|,\cdots,|L_{p_n\times q}|\}$，评价来源实体对被评价节点的直接信任度集合为 $R=\{DT_{p_1q},DT_{p_2q},\cdots,DT_{p_nq}\}$。间接信任度是否可靠取决于以下两点。

1）来源实体的广泛程度。集合 S 的元素越多，n 越大，则被评估节点的间接信任度的来源实体越广，越能反映其真实水平，同时伪造难度也越大。

2）除评估主体节点外，评价来源实体对被评估节点的直接信任度的离散程度。评价来源实体对被评价节点的直接信任度集合中元素的离散程度越大，说明这些节点在评价被评价节点的信任程度上存在的分歧越大，从而被评估节点在交互时的表现越不稳定，间接信任度的可靠度越低。

若间接信任度的来源广泛且对评估节点的直接信任度的离散程度较小，则说明间接信任度可靠，其参考价值较高，间接信任度的权重取大；若间接信任度来源单一且对评估节点的直接信任度的离散程度较大，则说明间接信任度的可靠性低，其参考价值较低，间接信任度的权重取小。因此，间接信任度的权重可以表示为：

$$\mu=\alpha\cdot\beta \tag{7}$$

其中，α 是广泛度系数，用来量化评价来源的广泛度：

$$\alpha=\frac{n}{n+1} \tag{8}$$

广泛度系数 $\alpha\in[0,1)$。除评估主体外没有其余评价来源实体，则 $n=0$，$\alpha=0$，$\mu=0$，评估主体完全依赖自身经验；评

价的来源实体越多,则 n 越大,α 越大,在 β 一定的情况下,μ 也越大。因此,式(8)满足广泛度系数的要求。

β 是信任离散系数,用来量化来源实体对被评价节点信任度的离散情况:

$$\beta=\frac{1}{\sigma+1} \tag{9}$$

其中,σ 是来源实体对被评价节点信任度集合的标准差:

$$\sigma=\sqrt{\sum_{k=1,k\neq i}^{n}(DT_{kj}-IT_{ij})^2/n} \tag{10}$$

可见,若评价来源实体对被评估节点的看法较为一致,各实体对被评估节点的直接信任度与间接信任度的差距较小,则 σ 偏小,信任离散系数 β 偏大,在 α 一定的情况下,μ 也偏大;若评价来源实体对被评估节点的看法有较大分歧,则 σ 偏大,信任离散系数 β 偏小,μ 偏小。因此,式(9)满足信任离散系数的要求。

2.4　综合信任度的计算

综合信任度的计算主要涉及评价数据的存取和综合信任度计算两个问题,即评价数据的存储、读取与使用的问题。由于 ChainTrust 模型进行信任度计算时需要被评估节点的完整评价数据,因此,需要保证评价数据的完整性和易获取性。

2.4.1　评价数据的存取

本节改进区块链结构[15-17],使用 Merkle Patricia 树和二叉 Merkle 树结构来存储评价数据,防止评价数据被篡改,提高评价数据的查询效率。需要说明的是,本节研究用于存储评价数据的区块链结构和该结构对评价数据的逻辑存取方法,文献[18-22]已经实现了分布式存储,因此本文不研究具体分布式节点的存储方案。

2.4.1.1　数据结构

存储评价数据的区块链结构如图 3 所示,区块链通过存储评价数据的区块的 hash 值将区块链连接。存储评价数据的区块分为区块头和区块体两部分。区块头包括 PrevBlockHash,State Root,Tx root 和 Nonce 这 4 个字段;区块体存放 Merkle Patricia 树和二叉 Merkle 树。

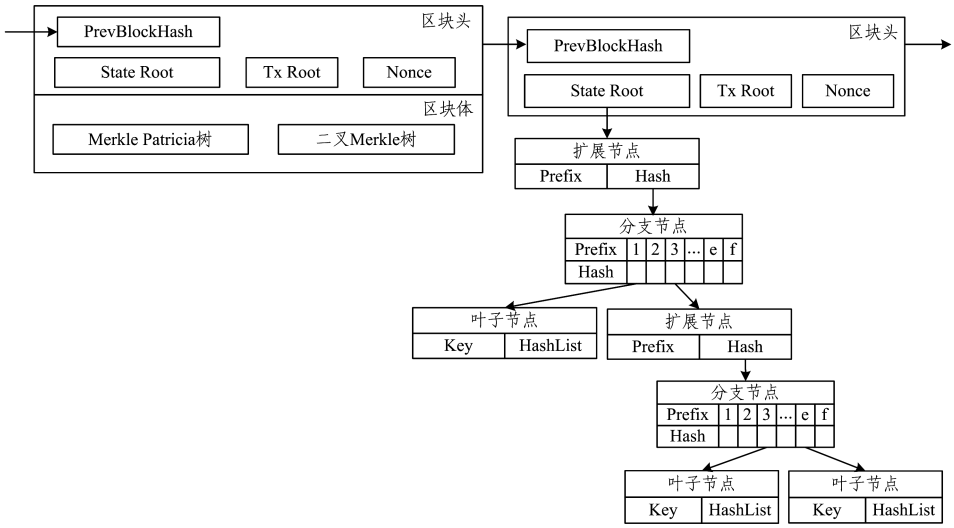

图 3　区块链结构

Fig. 3　Blockchain structure

1)PrevBlockHash

PrevBlockHash 是当前区块的前一区块的 hash 值。在当前区块中存入前一区块的 hash 值,可实现区块间的链式连接。

2)State Root

State Root 是记录网络中实体状态的 Merkle Patricia 树的根。Merkle Patricia 树包括扩展结点、分支结点和叶子结点,这些结点都是通过键值对 key/value 来记录数据。其中,扩展结点分为 prefix 和 hash 字段,prefix 记录 key 的相同前缀,hash 字段存储该扩展结点的子结点的 hash 值。分支结点通常在扩展结点之后,分支结点的 key 为单个 16 进制的字符或空字符 Null,Null 所对应的 value 是其父结点结尾的 key 所对应的 HashList。HashList 结构如图 4 所示。基于单个 16 进制字符的 key 前缀实现了树的分支,key 所对应的 value 值存放含有该 key 前缀的儿子结点的 hash 值。叶子结点包含了 key 和 HashList,依次组合 Merkle Patricia 树根、扩展结点、分支结点、叶子结点可获得实体名称 key。HashList 字段以列表形式保存 key 对应的实体评价数据的 hash 值。

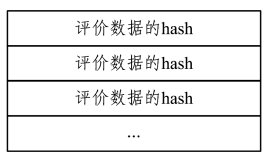

图 4　HashList 结构

Fig. 4　HashList structure

Merkle Patricia 是 Merkle 树和压缩前缀树的结合,扩展结点和分支结点记录了它们子结点的 hash 值,并通过 key/value 的形式将实体的名称对应起来,叶子结点的 HashList 记录了实体的所有评价数据的 hash 值,使得通过 Merkle Patricia 树向下可以迅速找到与特定实体有关的评价数据的 hash 值。

State Root 实现了对实体评价数据的快速查找。当构建新区块时,State Root 仅需要计算新区块中变化了的账户状态,即仅需更新与当前区块记录的评价数据有关的实体的分支,没变化的分支可以直接引用,而无需重新计算整棵树,减少了 State Root 的构建时间。

如图 5 所示，Merkle Patricia 树记录了实体“ab0c”“ab2e29”“ab2e28”的 HashList，若当前区块中新增一条实体“ab1c”的评价数据，则实体“ab0c”对应的叶子结点的 HashList 增加一条哈希记录。当前区块重新计算 State Root 时，只需直接引用没有改变的分支，再重新计算灰色部分的结点哈希值，就可以迅速计算出 State Root 的值。

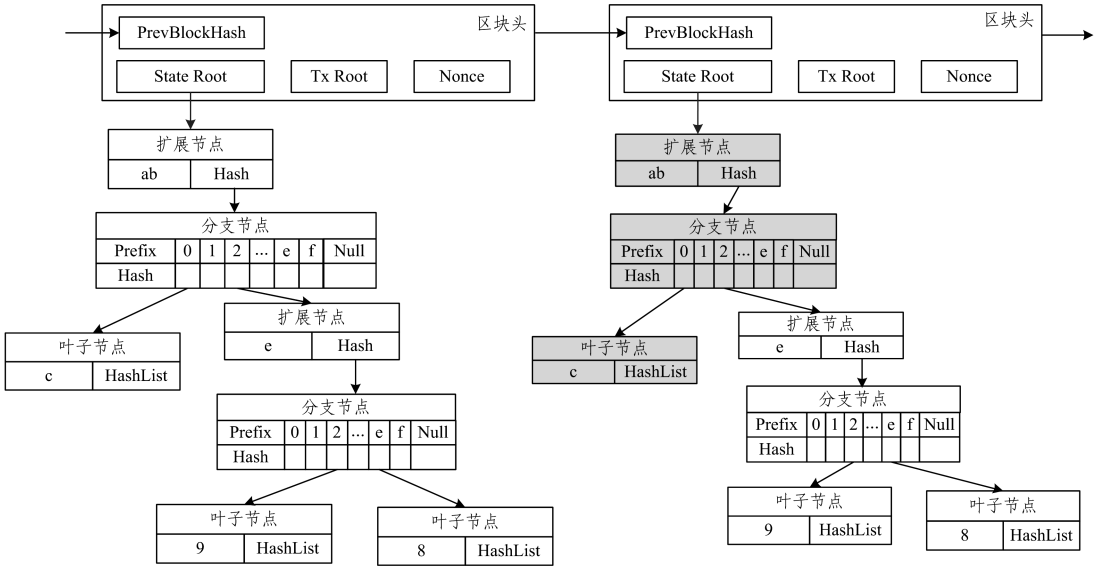

图 5 State Root 更新

Fig. 5 State Root updating

3）Tx Root

Tx Root 是二叉 Merkle 树的根，记录从上一区块结束到现在这个时间段内网络中产生的评价数据的摘要。树上的每个结点都是 hash 值，每个叶子结点对应一条评价数据的 hash 值。如图 6 所示，两个子结点的值连接之后，再次进行 hash 运算，可以得到父结点的值；如此反复，直到生成根 hash 值，即 Tx Root。通过 Merkle 根，块内任何数据的篡改都会改变 Tx Root 的值。

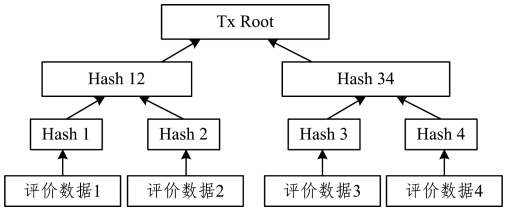

图 6 Tx Root

Fig. 6 Tx Root

4）Nonce

Nonce 是随机数，指使区块头部哈希值小于难度目标的整数。其与比特币区块链[15]中的 Nonce 的意义相同，这里不再赘述。

对区块头中的前块哈希 PrevBlockHash、随机数 Nonce、State Root 和 Tx Root 等元数据进行哈希运算后，可以得到此区块的哈希值，将其继续作为当前区块的下一区块的前块哈希，从而形成了区块链。区块头中包含 Merkle Patricia 根和二叉 Merkle 根，可以验证区块中的数据是否被篡改；区块头还包含 PrevBlockHash，可以向前验证区块是否被篡改。

2.4.1.2 存取算法

在信任计算前，需要通过查询 Merkle Patricia 树获取节点的评价记录，查找算法如算法 1 所示。

算法 1 Merkle Patricia 树查找算法 find(node=root,name)

输入：当前所在结点 node，默认是 root 结点，需要搜索的结点名 name

输出：结点名 name 中存放的 value

```
find(node,name):
if node is 叶子结点
  {
    if node.key==name
        return node.value
    else return Null
}
if node is 扩展结点
  {
    if node.key is name 的前缀
      name'←name−node.key
      go to son node //到该结点的儿子结点
      return find(node,name')
    else node.key is not name 的前缀
      if brother node != null
        go to brother node
        return find(node,key)
      else return Null
}
if node is 分支结点
  {
    if name==Null
      return key==Null 时所对应的 value
    if node.key 有 name 的前缀
      name'←name−node.key
      go to son node //到该结点的儿子结点
      return find(node,name')
}
end
```

Merkle Patricia 树查找算法 find(node,name)的输入是当前查找所在的结点 node 和需要搜索的结点名 name,node 结点最初指向 root 结点。

1)若结点是叶子结点,且该结点的 key 字段中的内容与 name 相同,则表示找到该结点,否则该结点不存在于树中。

2)若结点是扩展结点,且结点中的 key 是 name 的前缀,则将 name 减去前缀 key 生成新的字符串 name′,并将 node 指向 key 对应的儿子结点,继续递归调用 find(node,name);否则将 node 指向该结点的兄弟结点,继续递归调用 find(node,name),若所有兄弟结点不能匹配,则该结点不存在于树中。

3)若结点是分支结点,且此时 name 为空,则所寻找结点的 value 是该分支结点的 Null 对应的 value 值;若 name 不为空,匹配具有相同前缀的 key,去掉重复的前缀部分得到 name′,并将 node 结点指向这个 key 所指向的儿子结点,继续调用 find(node,name′)进行递归查询。

Merkle Patricia 树查找算法的时间复杂度为 O(log(*n*))。

节点获取了新的评价数据后,需要将新的评价数据插入,并更新 Merkle Patricia 树,具体由算法 2 实现。

算法 2　Merkle Patricia 树插入算法 Insert(name)

```
输入:待插入的结点名 name
输出:新的 Merkle Patricia 树
找到与待插入结点拥有最长路径前缀 prefix 的结点,记为 node
name←name−prefix
if node is 分支结点
  if name != null
    将待插入结点作为叶子结点插入到 node 结点对应的儿子列表中
  else
    将待插入结点的 value 存储在分支结点的 null 字段对应的 value 中
else
  if name==当前结点的 key
      update HashList
  if name !=当前结点的 key
      将 node 的儿子结点转变成分支结点
      node 结点转换为扩展结点
      将新结点与当前结点 key 的共同前缀作为当前结点的 key
      将新结点与当前结点的儿子结点作为儿子结点插入到分支结
      点的儿子列表
end
```

Merkle Patricia 树查找算法 Insert(name)的输入是待插入的结点名 name,算法首先获取与待插入结点拥有最长前缀的结点(最长前缀字符串记为 prefix,最长前缀的结点记为 node),然后减去相同的前缀,得到新的 name。如果 node 是分支结点且 name 不为空,将待插入结点作为叶子结点插入到 node 结点对应的儿子列表中;若 name 为空,将待插入结点的 value 存储在分支结点的 null 字段对应的 value 中。假设 node 是叶子结点或者扩展结点,如果 name 与当前结点的 key 一致,则直接更新当前结点的 HashList,完成插入;如果不一致,将 node 的儿子结点转变成分支结点,将 node 结点转换为扩展结点,将新结点与当前结点 key 的共同前缀作为当前结点的 key,将新结点与当前结点的儿子结点作为儿子结点插入到分支结点的儿子列表,完成插入。

Merkle Patricia 树查找算法的时间复杂度为 O(log(*n*))。

2.4.2　综合信任度计算

通过读取 Merkle Patricia 树,可以找到节点的所有评价数据,由算法 3 可以计算出综合信任度。

算法 3　综合信任度计算算法 CalcTrust(i,j)

```
输入:评估节点 i,被评估节点 j
输出:节点 i 对节点 j 的综合信任度 R_ij
1. find(name=j)
2. 从分布式数据库中取得评价数据
3. 将评价数据分为若干评价序列 L_1×j,L_2×j,…,L_i×j,…,L_n×j
4. 通过式(4)计算 DT_1j,DT_2j,…,DT_i×j,…,DT_n×j
5. 通过式(5)计算 IT_ij
6. 通过式(7)计算 μ
7. 通过式(6)计算 R_ij
8. end
```

从分布式数据库中获取评价数据,在 Merkle Patricia 树中找到被评估节点 j 所对应的叶子节点,获取被评估节点 j 的所有评价数据的哈希值,在分布式数据库中查询到对应的评价数据,然后将评价数据分解为若干评价序列 $L_{1\times j}$,$L_{2\times j}$,…,$L_{i\times j}$,…,$L_{n\times j}$。根据式(4)计算节点 i 对节点 j 的直接信任度 DT_{ij} 和其他节点对节点 j 的直接信任度 DT_{1j},DT_{2j},…,$DT_{n\times j}$。根据式(5)计算节点 i 对节点 j 的间接信任度 IT_{ij}。根据式(7)计算间接信任度的权重 μ。根据式(6)计算得出综合信任度 R_{ij}。

通过区块链结构可以快速找到实体完整的评价数据,间接信任度权重会随着评价数据的增加而调整。在每一次信任计算中,ChainTrust 模型都可以依据评价数据计算出间接信任度的权重。

3　针对恶意行为的策略

针对恶意行为的抵御分析是对 ChainTrust 模型抵御静态攻击、动态攻击和共谋攻击的效果进行分析。

1)静态攻击

静态攻击指攻击实体始终向其余实体提供虚假服务或恶意服务的行为。节点对外提供虚假服务或恶意攻击后,受害节点会对该节点提出负面评价,由于 ChainTrust 中惩罚因子 γ 的作用,静态攻击将导致攻击实体的信任度迅速下降。

2)动态攻击

动态攻击指恶意节点交替执行诚实行为和恶意攻击行为,其特点是良好行为和恶意行为交替出现,且攻击实体始终维持自身信任度在信任阈值之上。恶意节点攻击的前提是其信任度在信任阈值之上,因此节点在执行恶意行为后,需要执行良好的行为来提升其信任度。对动态恶意节点来说,其恶意行为执行的次数越多,其评价序列中的周期越多,摇摆频率 f_c 就越高。

ChainTrust 模型关注每一次评价电平的跳变,可以准确感知节点的摇摆频率。假设恶意节点的信任度始终维持一个定值 R_c,定值 R_c 略大于信任阈值,则由式(2)可以绘制出当前周期所需的周期收益图。以 $R_c=0.5$,$\rho=0.5$ 为例,其周期收益图如图 7 所示,图中横坐标表示第 n 个周期,纵坐标表示第 n 个周期所需要的周期收益。

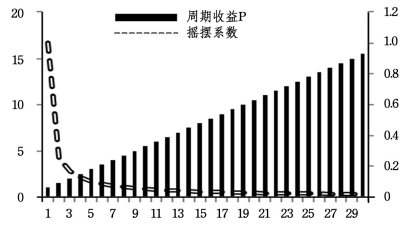

图 7 周期收益图

Fig. 7 Cycle income graph

由图 7 可知，随着周期的增加，摇摆频率提高，摇摆系数 $(f_c+1)^{-1}$ 降低。节点要想使信任度 R_c 维持定值，需要提高 $\sum_{k=1}^{n}\rho^{n-k}P_{ij}(k)$ 的值，又因为惩罚因子 ρ 对信任计算的影响，使得恶意节点针对本次恶意行为所进行的良好行为的补偿总多于前一次为恶意行为所做的补偿，即本周期的收益 $P_{ij}(k)$ 必须要大于上一周期的收益 $P_{ij}(k-1)$ 才能使得信任度 R_c 维持定值。

3)共谋攻击

共谋攻击指多个攻击实体给出高于或低于实际水平的评价，使得被评价实体的信任度与实际信任度偏离。根据攻击效果，可将共谋攻击分为诋毁攻击和夸大攻击。

①诋毁攻击

若多个恶意节点对服务节点进行诋毁攻击，被攻击的节点所收到的评价应包括正常评价和恶意评价两部分，此时所有与节点有过评价的节点对服务节点的信任度存在较大分歧。根据式(8)、式(9)，来源实体对被评价节点信任度集合的标准差 σ 偏大，离散系数 β 较小，在来源广泛度一定的情况下，间接信任度的权重 μ 较小。在计算节点综合信任度时，间接信任度的权重减小，则评估节点更多地参考自身经验，从而可以降低团体诋毁攻击对信任度计算带来的影响。

②夸大攻击

若多个节点对被评估节点进行夸大攻击，则服务节点和评价节点是同伙关系，使服务节点获得高于其实际服务质量的评价，目的是提高服务节点的综合信任度。夸大攻击通常是对恶意节点或低信任度节点的洗白操作，因此被评估节点中往往存在较多负面评价。负面评价是其对外提供恶意服务产生的，而正面评价是同伙节点给出的虚假评价。评价分歧较大，使得来源实体对被评估的节点的信任度差异较大。由式(8)、式(9)可知，来源实体对被评价节点信任度集合的标准差 σ 偏大，离散系数 β 较小，在来源广泛度一定的情况下，间接信任度的权重 μ 较小。在计算节点综合信任度时，间接信任度的权重减小，则评估节点更加关注自身经验，从而可以降低夸大攻击对信任度计算带来的影响。

4 仿真与分析

建立对等网络服务共享仿真环境，计算 ChainTrust 模型的计算交互成功率 S、误报率 F、漏报率 L 和信任误差 E，并将其与 EigenTrust＋＋[2]，PeerTrust[3]，DDTM-TR[11] 这 3 种目前较为典型的信任模型进行对比，以验证 ChainTrust 抵御静态攻击、动态攻击和共谋攻击的能力。改变参数 γ 和 ρ 取值，分析不同取值对信任度计算的影响。其中：

$$S=\frac{\text{高质量服务}}{\text{低质量服务数}+\text{高质量服务数}}$$

$$F=\text{Max}\left(\frac{\text{计算出的恶意节点数}-\text{实际恶意节点数}}{\text{实际恶意节点数}},0\right)$$

$$L=\text{Max}\left(\frac{\text{实际恶意节点数}-\text{计算出的恶意节点数}}{\text{实际恶意节点数}},0\right)$$

$$E=\left|\frac{\text{实际信任度}-\text{理论信任度}}{\text{理论信任度}}\right|$$

实际信任度指在受到共谋攻击后计算出的信任度，理论信任度指未受到共谋攻击计算出的信任度。

4.1 仿真环境

对等网络服务共享仿真环境通过 Netlogo[23] 和 Java Develop Kit(JDK)构建。Netlogo 是用于模拟自然和社会现象的建模平台，可以在建模中控制多个主体，能够很好地表现单个个体和宏观模型之间的关系，并且 Netlogo 提供 Java 编程接口。Java Develop Kit 是 Java 的开发工具包，可用于构建 Java 平台上的应用程序和组件等。通过编写 Java 程序对 Netlogo 进行扩展，可以实现复杂仿真环境或系统的定制。

网络中的节点既是服务提供方也是服务请求方，其作为服务请求方时，随机选择网络中的一个节点，获取该节点的评价数据，使用 ChainTrust 信任模型计算该节点的信任度。若节点信任度大于信任阈值，则发起服务共享请求，否则重新选择节点。交互完成后，服务请求方根据服务质量为服务提供方做出评价，服务请求方为高质量服务做出正面评价，为低质量服务做出负面评价。节点作为服务提供方时，当接收到服务请求后，以给定概率为请求方提供高质量服务或低质量服务。

在服务共享仿真环境中，所有节点的交互服从参数为 5 的泊松分布，即网络中每 5 s 有一次服务共享发生，各节点间的交互概率相等，仿真参数如表 1 所列。设置 1000 个节点，正常节点有 5％的概率向请求方提供低质量服务，恶意节点有 50％的概率向请求者提供低质量服务。

表 1 仿真参数

Table 1 Simulation parameters

名称	符号	默认值
节点总数	N	1000
恶意节点比例	M	20％
正常节点提供低质量服务概率	d_n	5％
静态攻击节点提供低质量服务概率	d_m	50％
惩罚因子	γ	5
折扣因子	ρ	0.8
初始信任度	δ	0.3
信任阈值	R_T	0.25

4.2 静态攻击

本组实验中，网络规模为 1000 个节点，共完成 1000 次服务共享操作，静态恶意攻击节点的比例从 0 增长到 50％，不同模型下的交互成功率如图 8 所示。

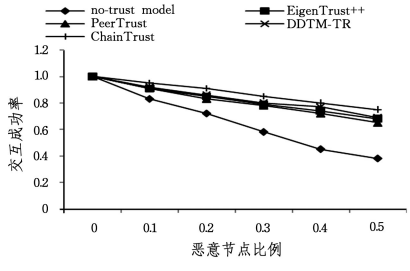

图 8 交互成功率

Fig. 8 Interaction success rate

由图 8 可知，4 种信任模型的下载成功率明显高于未采用信任模型的情况，说明 4 种信任模型抵抗静态恶意攻击的能力强于未采用信任的模型。其中，ChainTrust 模型的成功率略强于其余 3 种信任模型，这是由于 ChainTrust 模型在信任计算时考虑了节点的所有操作，能够较为准确地反映节点的品质；而 EigenTrust＋＋虽然也通过全局迭代，但是存在信任传递过程中将恶意节点的信任度提高的情况；PeerTrust 仅记录了节点的部分评价信息；DDTM-TR 模型对节点的信任计算只使用本地保存的信息，降低了信任计算的精确度。

4.3　动态攻击

通过仿真恶意节点的动态恶意行为，分析节点恶意行为与信任度的关系，同时将本文所提模型与另外 3 种模型进行对比，分析 4 种模型的抗动态攻击的水平。

动态恶意行为表现为节点在刚开始进行良好服务，在累积起一定的信任度后，再发起恶意行为，伴随着信任度的下降，当节点的信任度接近信任阈值时，停止恶意行为，重新开始良好服务以重新提升信任度。实验设置信任阈值为 0.25，节点的初始信任度为 0.3，刚开始提供良好服务以累积信任度，当信任度到达 0.75 以后开始提供恶意服务，当信任度下降至[0.25，0.3]区间时重新开始提供良好服务，直到信任度恢复至 0.75 以上，又重复之前的行为。图 9 记录 ChainTrust 模型中的动态恶意节点在前 500 次交互中的信任变化情况。

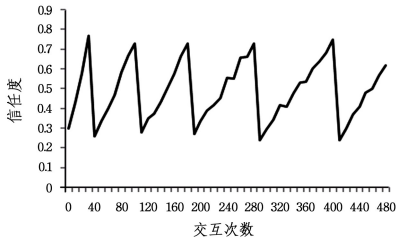

图 9　恶意节点的信任曲线

Fig. 9　Trust curve of malicious node

由图 9 可见，单个周期内，节点需要通过较多次连续的良好评价才能获得较高的信任度，而即使节点拥有较高的信任度，连续实施恶意行为也会致使节点的信任度迅速下降，ChainTrust 模型能在当前周期有效遏制住节点继续进行恶意行为。在后一个周期中，节点到达同样信任度所需要的正面评价数相比前一周期的斜率所需的评价数有所增加，当前周期中上升段的斜率比前一周期有所减小，而下降段的斜率相比前一周期有所增加，这说明随着摇摆周期的增多，节点的信任累积速率下降，信任损失速率上升，从而能够有效遏制节点通过积累信任实施恶意攻击的行为。

图 10 给出 5 个模型的交互成功率，横坐标是模型的交互成功率，纵坐标是动态攻击节点的比例。从图中可以看出，no trust 的曲线明显低于 EigenTrust＋＋，PeerTrust，DDTM-TR 和 ChainTrust 的曲线，ChainTrust 的曲线高于 EigenTrust＋＋，PeerTrust，DDTM-TR 的曲线，这说明 ChainTrust 模型抵御动态攻击的效果好于其余 4 个模型。

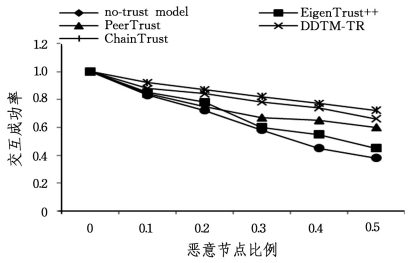

图 10　交互成功率

Fig. 10　Interaction success rate

4.4　共谋攻击

考虑到真实网络中同时存在诋毁攻击和夸大攻击，即恶意节点诋毁诚实节点同时抬高其余恶意节点的情况，在实验网络中设置执行动态恶意攻击、诋毁攻击和夸大攻击的恶意节点比例为 1∶1∶1。其中恶意节点对普通节点执行诋毁评价，对动态恶意行为节点执行夸大评价，这两种统称虚假评价。设置虚假评价占总评价数的 20%。

正常节点受到恶意节点的诋毁时，信任度下降，当信任度低于信任阈值时，该节点被认为是恶意节点，从而导致对该节点的误判。图 11 反映了诋毁评价比例与误报率的关系，可见 ChainTrust 对诋毁行为的防御能力稍好于 DDTM-TR，明显好于 EigenTrust，而 PeerTrust 的表现最差。

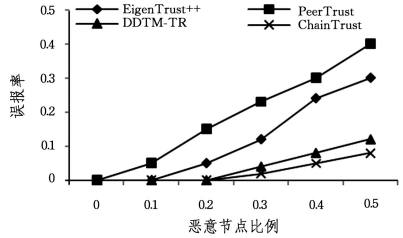

图 11　误报率的对比

Fig. 11　False positive rate comparison

节点对外提供恶意服务后其信任度会降低，同伙的夸大攻击可以提高其信任度，使其高于信任阈值，从而出现了对恶意节点的漏报。图 12 反映了夸大评价比例与漏报率的关系，可见 ChainTrust 对夸大攻击的抵御能力好于 DDTM-TR，EigenTrust 和 PeerTrust。

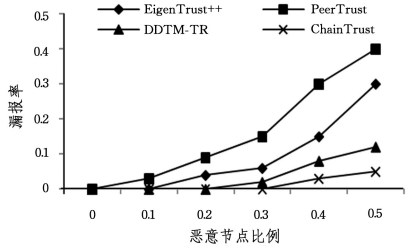

图 12　漏报率的对比

Fig. 12　Missing rate comparison

记录单个节点收到的虚假评价占节点收到的总评价数的比例与节点信任度的关系，并将其与完全真实评价的节点的信任度进行比较，得到信任误差，如图 13 所示。

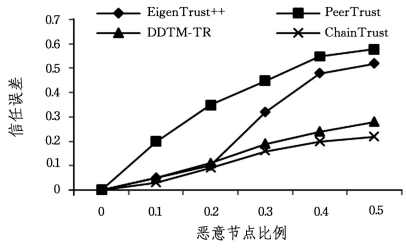

图 13 信任误差的对比

Fig. 13 Trust error comparison

由图 13 可以看出,信任误差随恶意节点比例的增大而升高。从整体表现来看,PeerTrust 对评价型攻击的抵御能力最弱,这是由于 PeerTrust 模型没有抵御共谋攻击的措施。EigenTrust++在恶意节点数量少于 20%时对评价型攻击的抵御情况与 DDTM-TR 和 ChainTrust 模型接近;随着虚假评价比例的上升,EigenTrust++的表现弱于 DDTM-TR 和 ChainTrust 模型,这是由于 EigenTrust++使用相似度来应对虚假评价攻击,当评价中的虚假评价占比较低时,模型认为虚假评价部分不可信,当评价中的虚假评价比例较高时,模型反而认为虚假评价部分为可信,而真实评价部分不可信,因此加大了虚假评价带来的影响。ChainTrust 模型对评价型攻击的抵御表现略好于 DDTM-TR 模型,在面对评价差异较大的节点时其能降低间接信任度的权重,更加依赖自身的历史经验,而 DDTM-TR 通过反馈算法对虚假评价进行过滤,因此二者的结果略有不同。

4.5 参数 γ 和 ρ 对信任度计算的影响

本组实验仿真了动态攻击节点,研究惩罚因子 γ 和折扣因子 ρ 对节点信任度变化的影响。摇摆型节点的初始信任度为 0.3,其起初进行良好服务,在初始信任度达到 0.75 以后开始进行恶意行为,当信任度低于 0.3 之后重新开始累积信任。在 200 次实验中,分析不同惩罚因子和折扣因子对节点信任度的影响。

惩罚因子 γ 代表对负面评价的敏感程度。由图 14 可以看出,惩罚因子 γ 越大,对负面评价的惩罚力度越大,节点信任度下降阶段的斜率越大,曲线越陡峭,信任度的下降速度越快。

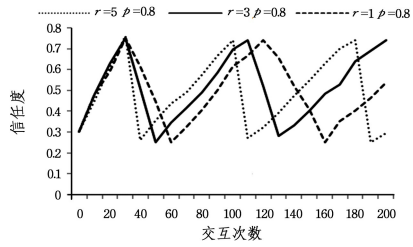

图 14 惩罚因子 γ

Fig. 14 Punish factor γ

折扣因子 ρ 代表对历史评价的敏感程度。由图 15 可以看出,折扣因子 ρ 越小,对历史评价的敏感程度越小,前期的评价对后续周期的信任度计算的影响越小。在不同折扣因子的相同周期内,单次评价的信任变化率随折扣因子数值的降低而增大,在图 15 中体现为在同一周期内折扣因子越小,曲线斜率越大。

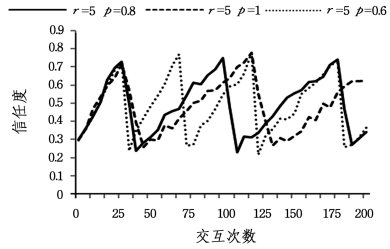

图 15 折扣因子 ρ

Fig. 15 Discount factor ρ

由以上分析可知,可以通过改变参数 γ 和 ρ 的取值来调节模型对恶意行为和历史评价的敏感程度,从而提升模型的灵活性和普适性。

结束语 本文研究了对等网络中节点的信任计算问题,提出了一种基于区块链的对等网络信任模型 ChainTrust。文中提出了评价序列、评价序列图、周期收益、序列收益、评价满意率等概念;根据评价序列图求解间接信任度的权重,解决了目前间接信任度的权重需要人工设置的问题;提出了基于区块链结构的评价数据存储方案,提升了评价数据的安全性,提高了查询效率;对网络中存在的静态攻击、动态攻击和共谋攻击进行分析,并设计仿真实验。分析和仿真表明,所提模型对静态攻击、动态攻击和共谋攻击均有较好的抵抗能力,且可以通过调节 γ 和 ρ 参数来调整模型对恶意行为和历史评价的敏感程度,提升模型的灵活性和普适性。

下一步研究包括细化交互类型、区分交互类型的不同评价、刻画节点在不同交互类型下的信任度,进一步提高信任模型对节点的评估能力,为对等网络应用提供信任支持。

参 考 文 献

[1] KAMVAR,SEPANDAR D,SCHLOSSER,et al. The EigenTrust Algorithm for reputation management in P2P networks[J]. Proceedings of the 12th international conference on World Wide Web,2003,12(1):640-651.

[2] FAN X,LIU L,LI M,et al. EigenTrust++:Attack resilient trust management[C]// International Conference on Collaborative Computing: Networking, Applications and Worksharing. IEEE Computer Society,2012:416-425.

[3] XIONG L,LIU L. PeerTrust:Supporting reputation-based trust for peer-to-peer electronic communities[J]. IEEE Transactions on Knowledge & Data Engineering,2004,16(7):843-857.

[4] NISHIKAWA T,FUJITA S. A reputation management scheme for peer-to-peer networks based on the EigenTrust trust management algorithm[J]. Journal of Information Processing,2012,20(3):578-584.

[5] LU K,WANG J,LI M. An EigenTrust dynamic evolutionary model in P2P file-sharing systems[J]. Peer-to-Peer Networking and Applications,2016,9(3):599-612.

[6] ZHANG L,LIU J W,WANG R C,et al. Trust evaluation model based on improved D- S evidence theory[J]. Journal on Commu-

nications,2013,34(7):167-173. (in Chinese)
张琳,刘婧文,王汝传,等. 基于改进 D-S 证据理论的信任模型[J]. 通信学报,2013,34(7):167-173.

[7] HAN Q W,REN M Y,WEN H. Topological potential based recommendation trust model for P2P communities system[J]. Journal of Electronics and Information Technology, 2015, 37(6):1279-1284. (in Chinese)
韩祺祎,任梦吟,文红. 基于拓扑势的 P2P 社区推荐信任模型[J]. 电子与信息学报,2015,37(6):1279-1284.

[8] CHEN S S. Trust model based on weight factor in P2P network [J]. Journal of Computer Applications,2013,33(6):1612-1614. (in Chinese)
陈珊珊. P2P 网络中基于权重因素的信任模型[J]. 计算机应用,2013,33(6):1612-1614.

[9] FANG Q,JI Y,WU G X,et al. Run length coding-based dynamic trust model for P2P network[J]. Journal of Software,2009, 20(6):1602-1616. (in Chinese)
方群,吉逸,吴国新,等. 一种基于行程编码的 P2P 网络动态信任模型[J]. 软件学报,2009,20(6):1602-1616.

[10] LIU Y C,LIANG Y H. Dynamic P2P trust model based on context factors[J]. Journal on Communications, 2016, 37(8): 34-45. (in Chinese)
刘义春,梁英宏. 基于上下文因素的 P2P 动态信任模型[J]. 通信学报,2016,37(8):34-45.

[11] YOU J,SHANG G J L,XU S K,et al. Distributed dynamic trust management model based on trust reliability[J]. Journal of Software,2017,28(9):2354-2369. (in Chinese)
游静,上官经伦,徐守坤,等. 考虑信任可靠度的分布式动态信任管理模型[J]. 软件学报,2017,28(9):2354-2369.

[12] RITU,JAIN S. A trust model in cloud computing based on fuzzy logic[C]// IEEE International Conference on Recent Trends in Electronics. IEEE,2017,47-52.

[13] LI H,MA X P,SHI J,et al. A recommendation model by means of trust transition in complex network environment[J]. Acta Automatica Sinica,2018,44(2):363-376. (in Chinese)
李慧,马小平,施珺,等. 复杂网络环境下基于信任传递的推荐模型研究[J]. 自动化学报,2018,44(2):363-376.

[14] MENG X,LIU D. GeTrust: A guarantee-based trust model in Chord-based P2P networks[J]. IEEE Transactions on Dependable & Secure Computing,2018,PP(99):54-68.

[15] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system [OL]. https://bitcoin.org/.

[16] DING W. Block chain based instrument data management system[J]. China Instrumentation,2015,1(10):15-17.

[17] Ethereum White Paper. A next-generation smart contract and decentralized application platform [OL]. (2015-11-12). https://github.com/ethereum/wiki/wiki/WhitePaper.

[18] TIAN J,DAI Y F. Study on durable peer-to-peer storage techniques[J]. Journal of Software,2007,18(6):1379-1399. (in Chinese)
田敬,代亚非. P2P 持久存储研究[J]. 软件学报,2007,18(6):1379-1399.

[19] CHEN Y,HUI L,LI K,et al. An improved P2P file system scheme based on IPFS and Blockchain[C]// IEEE International Conference on Big Data. IEEE,2018.

[20] TIAN R H,LU X L,HOU M S,et al. P2P based distributed storage system[J]. Computer Science,2007,34(6):47-48. (in Chinese)
田荣华,卢显良,侯孟书,等. P2P 分布式存储系统[J]. 计算机科学,2007,34(6):47-48.

[21] LI J,PENG Z,LI Y,et al. A data-check based distributed storage model for storing hot temporary data[J]. Future Generation Computer Systems,2017,73:13-21.

[22] YANG L,HUANG H,LI R F,et al. Composite P2P storage system based on group management[J]. Computer Science, 2010,37(1):64-67. (in Chinese)
杨磊,黄浩,李仁发,等. 一种基于分组管理的混合式 P2P 存储系统[J]. 计算机科学,2010,37(1):64-67.

[23] NetLogo. NetLogo[EB/OL]. http://ccl.northwestern.edu/netlogo/index.shtml.