基于集成信用度评估智能合约的安全数据共享模型

张乐君[1] 刘智栋[1] 谢 国[2] 薛 霄[3]

摘 要 区块链技术是一种新兴技术, 它具备防篡改、去中心化、分布式存储等特点, 可以有效地解决现有数据共享模型中隐私安全、用户控制权不足以及单点故障问题. 本文以电子病历 (Electronic health record, EHR) 共享为例提出一种基于集成信用度评估智能合约的数据共享访问控制模型, 为患者提供可信 EHR 共享环境和动态访问控制策略接口. 实验表明所提模型有效解决了患者隐私安全和对 EHR 控制权不足的问题. 同时就模型的特点、安全性以及性能进行了分析.

关键词 区块链, 信用度, 智能合约, 电子病历, 访问控制

引用格式 张乐君, 刘智栋, 谢国, 薛霄. 基于集成信用度评估智能合约的安全数据共享模型. 自动化学报, 2021, **47**(3): 594−608

DOI 10.16383/j.aas.c200797

Secure Data Sharing Model Based on Smart Contract With Integrated Credit Evaluation

ZHANG Le-Jun[1] LIU Zhi-Dong[1] XIE Guo[2] XUE Xiao[3]

Abstract Blockchain technology is an emerging technology, it has the characteristics of anti-tampering, decentralization, and distributed storage. It can effectively solve the problems of privacy security, insufficient user control rights, and single point failure in the existing data sharing model. This paper takes electronic health record (EHR) sharing as an example and proposes a data sharing access control model based on smart contract with integrated credit evaluation to provide patients with a trusted EHR sharing environment and dynamic access control policy interface. Experiments show that the proposed model effectively solves the problems of patient privacy security and insufficient control of EHR. At the same time, the characteristics, safety and performance of the model are analyzed.

Key words Blockchain, credit, smart contract, electronic health record (EHR), access control

Citation Zhang Le-Jun, Liu Zhi-Dong, Xie Guo, Xue Xiao. Secure data sharing model based on smart contract with integrated credit evaluation. *Acta Automatica Sinica*, 2021, **47**(3): 594−608

云辅助电子病历 (Electronic health record, EHR) 系统的广泛部署显示出在管理医疗机构和 EHR 方面的巨大好处[1], 世界各地纷纷采用新技术来管理 EHR. 然而除了巨大的优势之外, EHR 在云上的存储还面临着安全问题[2−3]: 1) 未经患者授权的第三方可能会恶意访问 EHR, 这对于 EHR共享中数据的完整性、隐私性和安全性存在不利影响[4]. 2) 患者很难跟踪和管理存储在云中的 EHR.

收稿日期 2020-09-25 录用日期 2020-11-04
Manuscript received September 25, 2020; accepted November 4, 2020
江苏省高等学校自然科学基金 (17KJB520044), 江苏省六大人才高峰项目基金 (XYDXX-108) 资助
Supported by Natural Science Fund for Colleges and Universities in Jiangsu Province (17KJB520044) and Six Talent Peaks Project in Jiangsu Province (XYDXX-108)
本文责任编委 杨涛
Recommended by Associate Editor YANG Tao
1. 扬州大学信息工程学院 扬州 225172 2. 西安理工大学自动化与信息工程学院 西安 710048 3. 天津大学智能与计算学部 天津 300072
1. College of Information Engineering, Yangzhou University, Yangzhou 225172 2. College of Automation and Information Engineering, Xi′an University of Technology, Xi′an 710048 3. College of Intelligence and Computing, Tianjin University, Tianjin 300072

针对这些问题, 许多研究[5−8] 提出了用于云服务器的存储、管理和共享技术. 这些研究使用不同的密码学技术和云技术设计 EHR 共享访问控制模型以实现隐私保护和访问控制. 尽管这些研究高度重视数据安全和隐私保护, 但系统仍然存在患者密钥管理困难、EHR 共享透明度不高以及密钥存在泄露风险等问题.

随着区块链技术的发展, 其特有的去中心化、可追溯性和隐私性推动了信息互联网向价值互联网的转变[9]. 越来越多的学者开始研究基于区块链的 EHR 共享模型. MedRec[10] 是一种使用区块链技术处理 EHR 的新型的、分布式的管理系统. MedRec 将模块化设计与医疗提供商现有的本地数据存储解决方案集成在一起. 薛腾飞等[11] 提出基于改进的 DPOS (Delegated proof of stake) 共识的区块链医疗共享模型, 详细介绍了模型的组件以及实现原理. 基于以太坊区块链的 Ancile[12] 利用智能合约增强访问控制和数据混淆. 文中详细描述了患者、EHR 提供商和第三方之间的交互过程. MedChain[13] 与 Ancile 类似, 通过精心设计智能合约实现访问控制, 同

时, MedChain 加入了激励机制, 给出了计算 EHR 质量的方法. 张超等[14] 提出基于 PBFT (Practical Byzantine fault tolerance) 的联盟式医疗区块链系统, 具有较好的适用性. 文献 [10−14] 为现有分散在各个机构的医疗数据提供基于区块链的访问控制模型, 但分散在各个医疗服务商手中的数据仍然存在被破坏的风险.

另一部分研究人员将云服务和区块链技术相结合来实现 EHR 的共享. Xia 等[15] 提出 BBDS (Blockchain-based data sharing) 模型, 该模型利用用户的身份和成员加密密钥来验证用户是否可以从共享池中获取数据. 但是, 对于成功加入共享组的成员没有其他限制. Tang 等[16] 着重研究了基于区块链的云存储模式下 EHR 共享的身份验证问题, 提出了多方授权的身份签名模型, 并具有很好的抗共谋能力. Liu 等[17] 设计了基于 CP-ABE (Ciphertext-policy attribute based encryption) 的访问控制机制和内容提取签名模型, 在数据共享方面提供了强大的隐私保护. 此外, 通过在智能合约中预设访问权限确保数据被安全共享. 文献 [15−17]采用云服务和区块链技术相结合的新模式, 但它们的访问控制策略较单一, 无法满足患者对 EHR 动态访问控制的需求.

现有的 EHR 共享研究中缺乏信用度评估机制, 本文采用云服务和区块链技术相结合的模式, 在区块链智能合约中加入用户信用度评估机制, 为患者提供动态可调节的访问控制策略. 本文的主要贡献有以下几个方面:

1) 提出了一种信用度评估机制, 将信用度评估机制集成到智能合约访问控制中, 在患者缺乏对第三方信任的环境中为患者提供信用度参考. 通过智能合约, 患者可以动态地调节访问控制策略.

2) 提出将云存储和基于权威证明 (Proof of authority, PoA) 共识机制的区块链相结合的框架. 云存储用于存储加密的 EHR, PoA 共识区块链保留加密 EHR 的索引. 我们对交易处理速度进行了统计分析, 证明了所提模型的可行性.

3) 基于以太坊 Go Ethereum 开发了所提模型的系统, 建立了一个私有链测试网络. 我们分析了系统中可能存在的恶意攻击和共谋行为, 并建立了基于信用度的奖惩机制和监督机制. 实验结果表明我们的机制能有效阻止恶意攻击行为并及时发现共谋行为. 最后, 理论分析表明, 我们的模型较现有一些通过智能合约实现 EHR 访问控制的模型更简单, 交易反馈延时更少.

1 预备知识

本节将介绍本文用到的智能合约技术和密码学技术.

1.1 智能合约

智能合约是存储在区块链上自动运行的脚本. 1994 年 Nick Szabo 提出相关概念, 将智能合约定义为一种通过代码程序自动执行的交易协议. 满足合约条款的交易相关者, 无需第三方管理者的监督就可自动执行交易. 由于缺乏可支撑合约自动执行的平台和相关技术, 直到区块链技术的出现, 才使得智能合约这项技术得到应用. 随着区块链的不断发展, 以太坊的出现[18] 首次将区块链和智能合约结合, 通过以太坊虚拟机 (Ethereum virtual machine, EVM) 来处理区块链上的交易. 区块链确保了智能合约的用户在可信的环境下遵循合约规则自动执行合约代码, 同时利用区块链的透明性和可追溯性, 跟踪合约状态. 智能合约的可扩展性、自动化为 EHR 共享提供了便利. 利用区块链中存储的医疗数据、支持外部数据的预言机[19] 以及信用度机制, 患者可以在智能合约中设置信用度阈值和其他访问控制参数, 实现复杂的访问控制策略.

智能合约预言机机制[19] 验证外部数据. 在智能合约中使用合约自带的函数 ecrecover 可以验证外部数据写入者的签名, 该函数需要数据的 Hash 值和签名对$\{v,\ r,\ s\}$. 所以对数据的签名需要遵循 ecrecover 函数的规则. 实际上, 数据需要经过两次 sha256的 Hash 操作后才能进行签名,签名结果中 32 字节的 r 和 s 来自椭圆曲线数字签名算法 (Elliptic curve digital signature algorithm, ECDSA) 的输出值, 一个字节的 v 则是用于恢复签名结果的标识, 以太坊中为 27 或 28.

在智能合约中, 数据主要分为 Storage 和 Memory 两种类型, Storage 类型数据也可称为合约的状态变量, 会永久存储在区块链中; Memory 则是临时变量, 交易处理完成后该类型变量会被清空. 所以在编写合约时, 需要为永久存储的数据定义 Storage 类型变量, 而不仅仅是处理交易逻辑.

2 基于集成信用度评估智能合约的安全数据共享模型

本节设计了一种基于智能合约的电子病历共享访问控制模型 (EHR smart contract access control model), 为方便描述, 将其简称为 EHR-SCAC. 下面分别从模型的整体框架和工作流程进行介绍.

2.1 EHR-SCAC 系统模型

如图 1 所示, EHR-SCAC 分为三层架构, 由数据获取层、数据存储层和数据共享层组成.

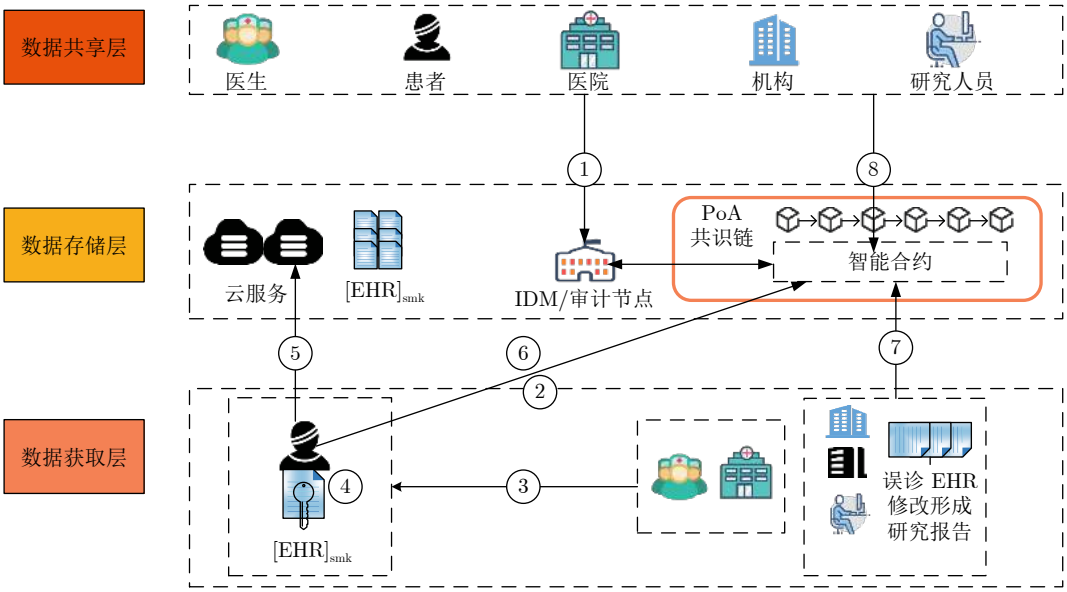


图 1 EHR-SCAC 整体架构

Fig. 1 EHR-SCAC overall framework

2.1.1 数据获取层

在本层中, EHR 由医生创建并发送给医院, 医院整理格式后, 由医生签名发送给患者. 签名的目的是确保 EHR 的完整性. EHR 的共享权和所有权属于患者. 同时, 机构在研究患者 EHR 时, 可能发现 EHR 存在误诊, 所以机构可以共享他们的研究报告.

2.1.2 数据存储层

数据存储层的主要功能是存储加密的 EHR 和给区块链提供 EHR 的存储索引 url, 数据存储层由以下两部分组成.

1) 云存储. 存储患者加密的 EHR, 给出 EHR 的存储索引 url.

2) PoA 区块链. 存储 EHR 的索引 url 并实现 EHR 共享. 患者通过区块链中智能合约预先定义访问控制策略, 确保 EHR 的安全共享, 任何人对 EHR 的访问都将保存在区块链网络中. 同时, 身份管理中心 (Identity management center, IDM) 和审计节点充当了链外数据进入区块链的媒介, 起到身份认证、审计和监督的作用.

2.1.3 数据共享层

在本层中, 已身份认证过的医疗工作者、机构可以访问患者的 EHR. 方便医生了解患者的病历史和给予患者更好的治疗, 同时也为医疗机构提供了重要的研究资料.

2.2 EHR-SCAC 工作流程

如图 1 所示, 模型的工作流程大致如下:

① 用户身份登记: 用户将真实身份 ID 和身份证明 V_{ID} 发送给 IDM, IDM 验证用户身份. 同时, IDM 会调用智能合约对用户进行成员登记, 用户以太坊公钥 pk 即为成员公钥. 将必要的用户信息存储在区块链中. 如果用户是机构, 则机构会成为审计节点.

② 身份认证后的患者, 可以调用智能合约设置 EHR 的全局访问控制策略.

③ EHR 的生成: 该过程参与的实体有医生、医院和患者. 为了确保 EHR 的完整性, 在医院将 EHR 发给患者前, 医生需要对 EHR 的 Hash 值 ehr_hash 使用以太坊私钥进行签名, 记为 sig_d. 如果 EHR 存在问题, 则医疗纠纷的责任最终由医生承担, 这也符合实际情况. 医院将 EHR、医生的签名 sig_d 发给患者.

④ 患者从医院获取到 EHR 和 sig_d 后, 患者的本地客户端随机生成一个对称密钥 smk 加密 EHR, 得到 $[\mathrm{EHR}]_{\mathrm{smk}}$.

⑤ 患者将 $[\mathrm{EHR}]_{\mathrm{smk}}$ 和 ehr_hash 上传到云端, 云端返回 EHR 的存储索引 url.

⑥ 患者使用以太坊公钥 pk_p 加密 smk 获得 $[smk]_{pk_p}$, 然后调用智能合约的共享 EHR 接口函数, 将 EHR 的索引等信息存入区块链. 智能合约检测患者的身份注册信息, 验证通过后合约会记录 url 与患者以太坊公钥 pk_p 的对应关系, 将必要的数据存储到合约相关的变量中. 同时, 合约会为该 EHR 初始化一个白名单并将患者加入其中. 随后, 患者便可调用合约给白名单添加成员, 方便非机构用户如医生的访问.

⑦ 机构在研究 EHR 时可能发现患者的 EHR 存在误诊, 机构可以更正 EHR, 将研究报告证明发布到云端. 云端返回研究报告的索引 r_url. 机构调用智能合约共享研究报告 r_url. 与预言机机制相同, 合约会触发投票事件通知审计节点根据研究报告索引在链下对报告进行审核. 审计节点会调用智能合约改变研究报告投票状态变量. 当投票数超过设定的阈值, 机构便成功共享研究报告, 这有利于机构信用度的增加. 同时, 机构链下通知 IDM, IDM 联系患者重新共享更正后的 EHR. 该过程患者重复②③④操作, EHR 的医疗纠纷也由原先的医生负责变为机构负责, 且需要将更正后的 EHR 的索引记录到错误 EHR 记录的结构体变量中.

⑧ 如⑥中所述, 若请求者在 EHR 的白名单中, 则可以直接查询到解密密钥. 否则, 请求者需要调用智能合约获取 EHR 的请求权限, 智能合约会触发事件随机选择一个审计节点对请求者发起工作量证明 (Proof of work, PoW) 难度挑战 (第 2.3.2 节中将详细介绍). 完成 PoW 挑战后, 请求者获得审计节点的签名 sig_A, 然后请求者再次调用智能合约请求 EHR 的解密密钥. 智能合约根据患者制定的访问控制策略验证请求者. 通过访问控制策略后, 请求者会被加入患者 EHR 的白名单中, 同时合约触发代理重加密事件, IDM 通知患者生成重加密密钥 $[K]_{p\text{-}r}$ 并发送给云端, 云端执行代理重加密任务, 生成 $[smk]_{pk_r}$ 并发给请求者, 请求者可以调用智能合约将解密密钥存入智能合约中.

2.3　EHR-SCAC 的信用度评估机制

本节提出一种信用度评估方法, 同时设计信用度奖惩机制和监督机制用来维护系统的稳定和安全.

2.3.1　EHR-SCAC 信用度评估

机构的信用度并不能从单一特性进行评判, 本文采用模糊层次分析法 (Fuzzy analytic hierarchy process, FAHP)[20−21] 对用户的信用度进行评判. 先将用户的信用度分为 n 个特性, 再把每个特性分为若干个特征类型, 将模糊的用户行为信用评估问题转化为简单的、明确的信用特征加权求和问题.

FAHP 解决问题的步骤分为 4 步: 1) 分析问题, 将问题划分为多层次结构; 2) 以上一级要素为准, 将同一层次的特征两两比较, 获得初始判断矩阵; 3) 将初始判断矩阵转换为模糊判断一致矩阵, 通过计算确定各特征以及各特性的权重; 4) 根据规范化的特征值和权重计算出信用度.

步骤 1. 如图 2 所示, 将信用度分为 3 层. 为了不给区块链造成负担, 需要建立易检测、易收集的用户行为特征方案. 在处理合约交易时会更新机构的这些行为特征 (在第 2.4 节中将作详细说明).

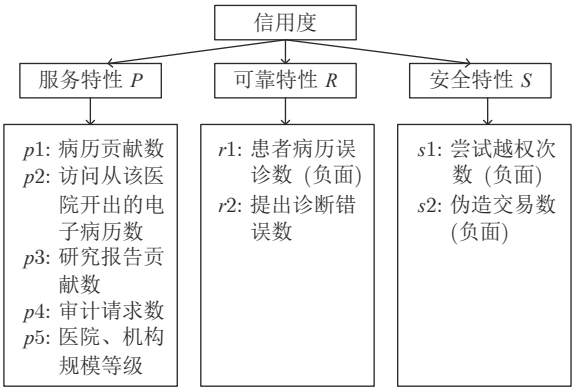


图 2　用户行为特征分类

Fig. 2　Classification of user behavior characteristics

步骤 2. 建立行为特征矩阵 $C=[c_{ij}]_{n\times m}$, n 为特性个数, m 为特性中行为特征个数的最大值, 不足的项用零表示. 由于各特征的值区别较大, 我们需要对矩阵 C 进行归一化处理, 将值规范为 [0,1] 的特征矩阵 $E=[e_{ij}]_{n\times m}$. 将同一特性下的特征的重要性两两比较获得初始判断矩阵 $EQ=[eq_{ij}]_{v\times v}$, v 是某个特性下行为特征的个数, 如服务特性 P 下 $v=5$.

以服务特性为例, 其特征矩阵 $E_p=[e_1,e_2,\cdots,e_v]$, 利用式 (1) 获得初始判断矩阵 EQ.

$$eq_{ij}=\begin{cases}0, & e_i<e_j\\ 0.5, & e_i=e_j\\ 1, & e_i>e_j\end{cases}\tag{1}$$

然后, 通过式 (2) 将初始判断矩阵转化为模糊判断一致矩阵 $Q=[q_{ij}]_{v\times v}$.

$$\begin{aligned}q_i&=\sum_{k=1}^{v}eq_{ik}\\ q_{ij}&=\frac{q_i-q_j}{2v}+0.5\end{aligned}\tag{2}$$

步骤 3. 使用式 (3) 计算服务特性下各特征的权重向量

$$w_{pi}=\frac{\sum\limits_{k=1}^{v}q_{ik}-0.5}{\frac{v(v-1)}{2}}\tag{3}$$

从而得到 $\boldsymbol{w}_P=[w_{p1},w_{p2},w_{p3},w_{p4},w_{p5}]$. 对于其他特性, 同样利用式 (1) ~ (3) 可得可靠特性 R 的特征权重向量 $\boldsymbol{w}_R=[w_{r1},w_{r2}]$, 安全特性 S 的特征权重向量 $\boldsymbol{w}_S=[w_{s1},w_{s2}]$, 以及特性权重向量 $\boldsymbol{w}_F=$

$[w_{f1}, w_{f2}, w_{f3}]$.

步骤 4. 根据式 (4) 计算机构的信用度.

$$Credit = Cr^{Pt} + \sum_{i=1}^{n}\sum_{j=1}^{m} w_f^{Ng} w_c^{Ng} - Cr^{Ng} \tag{4}$$

其中, w_c^{Ng} 属于 $\boldsymbol{w}_P$、$\boldsymbol{w}_R$ 和 $\boldsymbol{w}_S$, 表示负面特征的权重. w_f^{Ng} 属于 $\boldsymbol{w}_F$, 表示负面特征权重所在特性的特性权重. Cr^{Pt} 和 Cr^{Ng} 分别是正面信用度和负面信用度, 它们的计算可统一为式 (5), 其中, w_f 属于 $\boldsymbol{w}_F$, w_c 属于 $\boldsymbol{w}_P$、$\boldsymbol{w}_R$ 和 $\boldsymbol{w}_S$.

$$Cr = \sum_{i=1}^{n}\sum_{j=1}^{m} w_f w_c e_{ij} \varepsilon \tag{5}$$

即正面信用度为正面行为特征与其权重乘积的和, 负面信用度为负面行为特征与其权重乘积的和. 但是, 对于图 2 中 $s1$ 和 $s2$ 这类违规行为的计算需要进行改进. 对这两个行为目的是监督, 同时考虑到机构的误操作会导致此类事件发生, 所以机构如果存在 $s1$ 和 $s2$ 违规时, 系统应能及时对机构的信用度进行惩罚, 并且机构的信用度随时间流逝能够逐渐恢复. 所以, e_{ij} 为 $s1$ 和 $s2$ 时, 这两个行为特征的计算需乘以一个系数 ε, ε 如式 (6) 所示. n_{ij} 表示机构做出负面行为特征 e_{ij} 的次数. bn 表示当前区块号, bn_{ij} 表示负面行为特征 e_{ij} 最后一次发生时所在的区块号.

$$\varepsilon = \begin{cases} 1, & e_{ij} \notin \{s1, s2\} \\ \dfrac{2^{n_{ij}}}{bn - bn_{ij}}, & e_{ij} \in \{s1, s2\} \end{cases} \tag{6}$$

2.3.2 信用度奖惩机制

比特币和以太坊采用 PoW 共识机制来维护区块链的安全, 使得恶意节点很难成功攻击区块链, 除非恶意节点掌握了全网 51% 的算力. 虽然 PoW 机制对于交易验证速度要求极高的场景应用十分有限, 但其思想值得学习[22].

第 2.3.1 节给出了计算信用度的方法. 从第 2.3.1 节中的图 2 可以看到两种恶意行为, 在介绍信用度奖惩机制前首先了解两种恶意行为.

1) 未经授权访问行为

机构请求 EHR 时, url 对应的患者 pk_p 可能会被未通过患者访问控制策略的机构替换, 选择访问控制策略要求低的患者, 从而绕过 EHR 持有者的访问控制策略. 与前面类似, 未通过患者访问控制策略的机构利用其他高信用度节点的成员公钥 pk 来发送合约交易, 以此达到访问患者 EHR 的目的.

2) 伪造签名行为

机构请求 EHR 的解密密钥前需要获取审计节点的签名 sig_A, 所以机构可能伪造签名直接申请解密密钥.

在我们的系统中, 患者共享 EHR 时, 智能合约会记录 url 与患者的成员公钥 pk 的关系. IDM 在用户通过系统身份认证后调用智能合约将以太坊地址和成员公钥 pk 的对应关系写入了区块链. 同时, 智能合约的 ecrecover 函数可以验证签名. 所以, 以上行为可以得到阻止. 但系统无法阻止恶意节点发送大量此类的无效交易, 如果不能有效制约, 将会对系统的稳定性造成影响. 考虑到比特币和以太坊通过 PoW 机制使得恶意节点的攻击成本很高. 所以本文设计了基于信用度的 PoW 奖惩机制. 在我们的系统中, 智能合约记录了机构的信用度属性 (即图 2 中的行为特征), 一旦检测到违规行为, 违规行为会被保存到信用度属性中, 机构的信用度也随之下降. 式 (7) 给出了 PoW 难度与信用度的关系

$$Diff = \lambda \times (1 - Credit) \tag{7}$$

其中, $Diff$ 表示机构请求 EHR 的难度. λ 是一个固定值, 其应该根据机构的普遍算力进行设置, 实际应用中可以参考比特币的算力更新方法, 根据出块时间即算力调整 λ. 本文测试了实验设备的 PoW 难度与 PoW 算法执行时间, 根据测试结果将 λ 设为 8, 使机构违规情况下 PoW 算法执行时间能够达到惩罚目的, 同时保证正常信用度机构 PoW 算法执行时间很低.

下面介绍机构请求 EHR 前解决 PoW 难题的过程. 如图 3 所示, 1) 机构调用智能合约请求索引 url; 2) 智能合约触发事件随机选举一个审计节点处理机构的请求; 3) 审计节点根据机构信用度将挑战难度 $Diff$ 和时间戳 $timestamp$ 发送给机构; 4) 机构根据式 (8)算出能够使得 out 小于 $Diff$ 的 $nonce$, 并将 out 和 $nonce$ 发送给审计节点; 5) 审计节点对 pk 和 url 进行签名, 并将签名 sig_A 发送给机构, 机构再次调用智能合约请求 EHR 的解密密钥, 智能合约会检测审计节点的签名是否有效.

$$out = hash\{pk||url||timestamp||nonce\} \tag{8}$$

通过审计节点发起 PoW 挑战是一个可行办法, 但审计节点不是 IDM 这样的可信政府机构. 所以, 如果攻击者买通了大部分的审计节点, 则攻击者可以直接拿到审计节点的签名 sig_A, 这样依旧可以短时间发起大量攻击. 为了遏止审计节点的共谋行为, 需要在智能合约中建立检测共谋行为的机制, 及时通知 IDM 采取相应的惩罚. 注意到如果攻击者连续攻击, 式 (6) 中 n_{ij} 就会起到关键作用, 攻击者的信用度会快速下降, 所以攻击者的挑战难度会随之升高. 定义

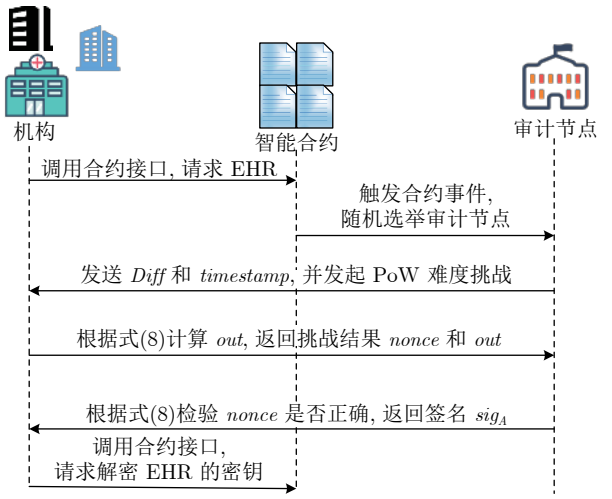


图 3　机构解决 PoW 难题的过程

Fig.3　The process of the institution solving the PoW problem

$$\eta = \left(1 - \frac{n_g + Diff_{\max}^{\text{normal}}}{Diff_{\text{last}}^{\text{illegal}}}\right) \times 100\% \tag{9}$$

其中, n_g 表示攻击的区块间隔, 由当前区块号 bn 减去上一次攻击发生的区块号 bn_{ij} 获得, 可知连续攻击时 n_g 很小. $Diff_{\max}^{\text{normal}}$ 表示未违规情况下系统中信用度最低的用户对应的难度, $Diff_{\text{last}}^{\text{illegal}}$ 表示最后一次违规时机构信用度对应的 PoW 难度, η 则表示审计节点共谋的概率. 当 η 超过了合约中设定的阈值, 就会触发事件通知 IDM 对共谋双方采取相应的惩罚.

以上是本文通过信用度奖惩机制实现对机构恶意行为的约束, 从而保护系统稳定性的方法.

2.4　EHR-SCAC 的智能合约访问控制

在我们的系统中, 智能合约负责所有交易的逻辑处理. 客户端利用 Web3.js[23] 使患者、机构可以调用智能合约接口函数. 图 4 展示了合约中数据结构和函数的设计. 图中展示了变量在合约中的实际存储类型、合约的内部函数和对外的接口函数以及系统中涉及的结构体. 函数的参数图中未给出, 在后文算法描述过程中将给出函数的参数. 其中, "+"表示外部可调用, "–"表示只能是合约内部可调用, ":"后表示数据存储类型, "→"表示映射关系.

从图 4 中可以看到智能合约分为三个部分: 智能合约状态变量、智能合约事件以及智能合约函数. 其中智能合约状态变量提供了永久性存储在区块链中的数据, 为用户身份验证、患者共享 EHR、机构共享研究报告、机构信用度计算提供数据支撑; 智能合约事件用于通知链外世界, 实现链外信息与区块链信息的交互; 智能合约函数分为内部函数和接口函数, 内部函数用于合约内部事务的处理, 接口函数供用户调用. 表 1 给出了这三个部分变量和函数的具体作用.

以上是 EHR-SCAC 系统合约的设计, 接下来将对系统中节点注册、患者共享 EHR、机构共享研究报告、患者设置访问控制参数、机构请求 EHR 和信用度特征收集这六个过程进行详细描述.

2.4.1　节点注册

算法 1 由用户和 IDM 执行, 输入的参数有用户的真实身份 ID, 用户身份证明信息 V_{ID}, 用户以太坊地址 $Eth\_address$, 用户以太坊公钥 pk. 该算法用于登记用户, 同时将必要的用户信息存入区块链中.

算法 1.　节点注册算法

输入.　ID, V_{ID}, $Eth\_address$, pk

1)　用户发送 ID, V_{ID}, $Eth\_address$ 和 pk 给 IDM

2)　IDM 验证用户身份信息

3)　IDM 调用合约函数 $node\_register(pk, Eth\_address, role)$

4)　更新 $address\_pk[Eth\_address]$

5)　**if** 用户是机构 **then**

6)　　初始化机构的信用度属性信息 $pk\_credit[pk]$

7)　　将机构信息加入变量 $audit\_node$ 中

8)　**end if**

用户注册了以太坊账号后可以向 IDM 申请将信息登记到区块链中. 在该过程中, IDM 验证用户身份信息, 身份验证成功后 IDM 调用合约接口 $node\_register()$ 将用户的信息写入区块链. 如果用户是机构, 智能合约会初始化机构的信用度属性, 属性信息被记录到变量 $pk\_credit[pk]$ 中, 然后将机构信息更新到审计节点变量 $audit\_node$ 中. 节点注册使得用户成为 EHR-SCAC 系统的一员, 为患者共享 EHR 和机构请求患者共享的 EHR 做准备.

2.4.2　访问控制参数设置

算法 2 由患者执行, 输入的参数有患者成员公钥 pk_p, 患者以太坊地址 $Eth\_address$, 访问控制参数集 $attr$. 交易成功返回交易 ID: $Policy\_TXID$, 失败则返回 faile.

患者成功加入区块链后可以调用合约设置访问控制参数, 患者调用合约函数 $set\_strategy()$ 创建交易 $Policy\_TXID$, 合约通过 $verify\_pk()$ 验证患者是否完成节点注册, 验证后将患者访问控制策略 $attr$ 存入合约变量 $strategy[pk]$ 指向的结构体 $pk\_strategy$ 中.

算法 2. 访问控制参数设置算法

输入. pk_p, $Eth\_address$, $attr$

1) 患者调用合约函数 $set\_strategy(attr, pk_p)$

2) **if** $verify\_pk(pk_p, Eth\_address) = forged$ **then**

3) **return** faile

4) **else**

5) 智能合约使用 $attr$ 更新患者的访问控制策略集 $strategy[pk_p]$ //$strategy[pk_p]$ 是结构体变量 $pk\_strategy$

6) **return** $Policy\_TXID$

7) **end if**

2.4.3 EHR 的共享

算法 3 由患者执行, 输入的参数有 EHR 的索引 url, url 的 hash 值 url_hash, 患者成员公钥 pk_p, 患者以太坊地址 $Eth\_address$, 患者加密 EHR 的对称密钥 $[smk]_{pk_p}$, EHR 的 hash 值 ehr_hash, 医院的成员公钥 pk_h, 医生的签名 sig_d. 交易成功返回交易 ID: EHR_TXID, 失败则返回 faile.

算法 3. EHR 共享算法

输入. url, url_hash, pk_p, $Eth\_address$, $[smk]_{pk_p}$, ehr_hash, pk_h, sig_d

1) 患者调用合约函数 $contribute\_EHR(url, url\_hash, pk_p, [smk]_{pk_p}, pk_h, ehr\_hash, sig_d)$

2) **if** $verify\_pk(Eth\_address, pk_p) =$ true

3) 合约将 EHR 的相关信息存入 $EHR\_share[url\_hash]$ 变量中

4) 合约初始化 EHR 的白名单, 将患者的信息加入白名单 $url\_whitelist[url\_hash]$ 中

5) 合约调用内部函数 $set\_url\_pk(url\_hash, pk_p)$

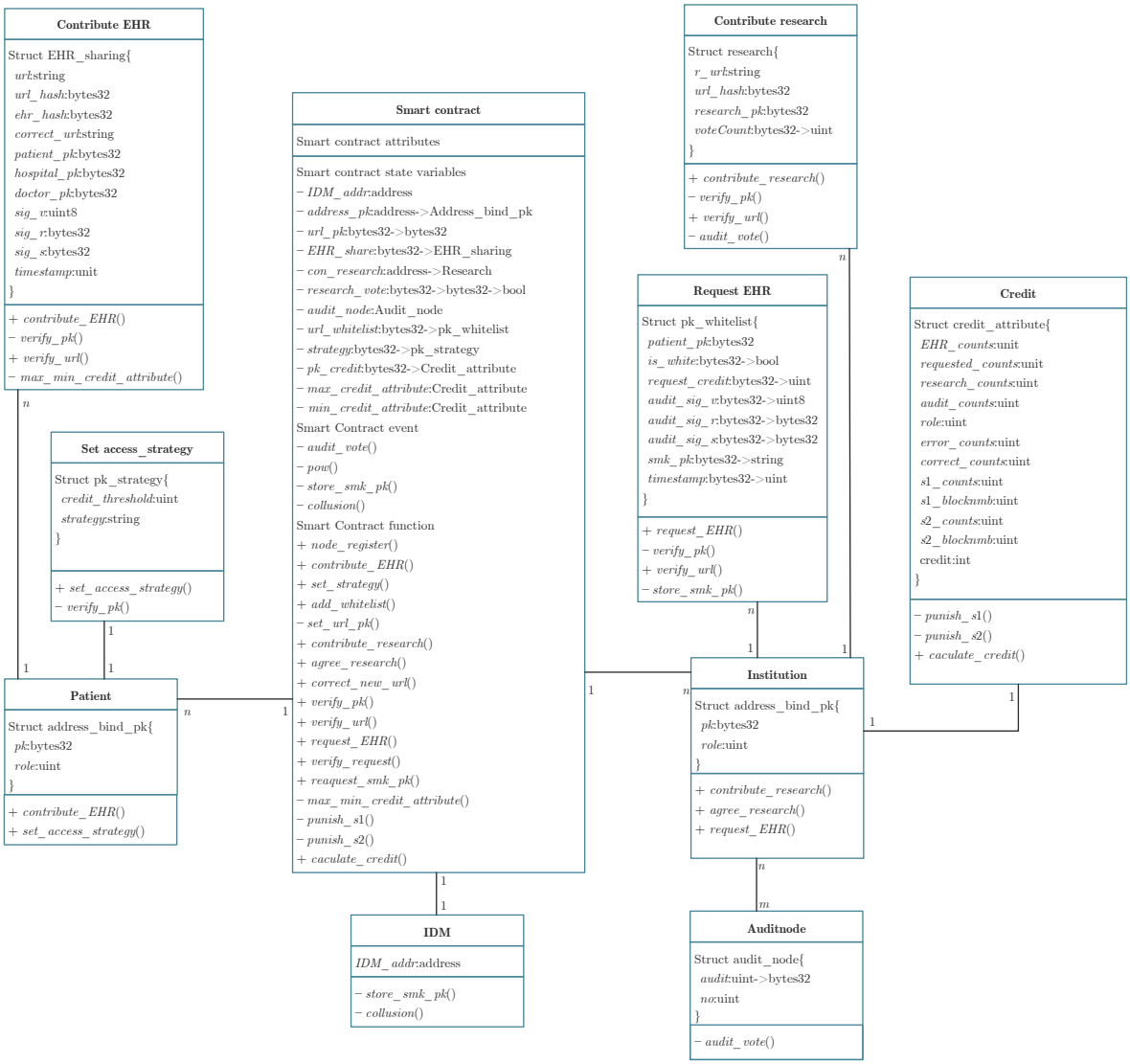


图 4 EHR-SCAC 的智能合约实现

Fig. 4 Smart contract implementation of EHR-SCAC

表 1　EHR-SCAC 智能合约变量和函数说明

Table 1　Description of EHR-SCAC smart contract variables and functions

智能合约状态变量	功能描述
IDM_addr	IDM 的以太坊地址, 用于检测调用合约者是否是 IDM
address_pk	由用户的以太坊地址映射到结构体 *Address_bind_pk*, 记录了用户成员公钥 *pk* 与以太坊地址的对应关系和用户的角色信息
url_pk	由 EHR 的索引*url* 的 hash 值映射到 *pk*, 记录了*url* 与患者pk_p的对应关系
EHR_share	由 EHR 的索引*url* 的 hash 值映射到 EHR 结构体 *EHR_sharing*, 记录了患者共享的 EHR 的具体信息
strategy	由患者成员公钥 pk_p 映射到策略结构体 *pk_strategy*, 记录了患者的访问控制策略
con_research	由 *r_url* 的 hash 值映射到研究报告结构体 *Research*, 记录了机构共享的研究报告相关信息
research_vote	由审计节点的成员公钥 pk_A 映射到 *r_url* 的 hash 值, 再映射到 bool 类型值, 记录了审计节点对 *r_url* 研究报告是否已经投过票, 防止重复投票
audit_node	该变量是审计节点结构体 *Audit_node* 类型变量, 结构体中的 *no* 变量用于记录审计节点个数, 也为随机选举审计节点提供参考
pk_credit	由机构的成员公钥 pk_I 映射到信用度结构体 *Credit_attribute*, 记录了机构的信用度属性
url_whitelist	由 EHR 的索引*url* 的 hash 值映射到白名单结构体 *pk_whitelist*, 记录患者为共享的 *url* 设置的白名单成员的相关信息
max_credit_attribute	记录了所有机构中各信用度属性的最大值, 用于信用度计算时信用度属性的归一化处理, 该变量数据类型是信用度结构体 *Credit_attribute*
min_credit_attribute	记录了所有机构中各信用度属性的最小值, 用于信用度计算时信用度属性的归一化处理, 该变量数据类型是信用度结构体 *Credit_attribute*
智能合约事件	功能描述
audit_vote()	机构共享研究报告时, 将会触发 *audit_vote*() 事件, 监听该事件的审计节点链下审核研究报告, 在时间阈值内达到投票阈值后交易成功上链
pow()	机构请求申请 EHR 的权限时会触发该事件, 审计节点链下会对请求者发起 PoW 挑战
store_smk_pk()	机构在成功请求患者 EHR 后会触发 *store_smk_pk*() 事件, 监听该事件的IDM会链下通知患者生成重加密密钥 $[K]_{p\text{-}r}$. 然后云端执行代理重加密任务, 并将重新加密的解密密钥 $[smk]_{pk_r}$ 发给请求者
collusion()	在请求者调用 *request_smk_pk*() 函数后, 如果合约检测到请求者存在违规行为, 则合约会计算审计节点共谋的概率 η , 如果 η 超过阈值说明审计节点参与共谋, 这时触发 *collusion*() 事件, 监听该事件的 IDM 会给与共谋节点相应的惩罚
智能合约函数	功能描述
node_register()	用于登记通过身份验证的用户, 仅 IDM 可以调用, 该函数将用户的以太坊地址与成员公钥 *pk* 的关系记录在变量 *address_pk* 中
contribute_EHR()	用于患者共享 EHR, 该函数会将 EHR 的 *url*、hash 值等重要信息记录到变量 *EHR_share* 中
set_strategy()	患者设置访问控制策略的接口, 患者的访问控制策略参数会存储到变量 *strategy* 中
add_whitelist()	患者设置白名单的接口, 可以为指定的 EHR 设置白名单, 白名单的成员信息被记录在变量 *url_whitelist* 中
set_url_pk()	将 EHR 的 *url* 与患者成员公钥 pk_p 的关系记录在变量 *url_pk* 中
contribute_resarch()	机构共享研究报告的接口函数, 研究报告的相关信息被记录在变量 *con_research* 中
agree_research()	审计节点对机构共享修改的EHR投票的接口, 投票结果记录在变量 *con_research* 中
correct_new_url()	患者将机构更正后的 EHR 重新上传云端后, 通过调用该函数接口可以将更正后的 *url* 记录到原先错误 EHR 的记录中, 方便请求者知晓原先的 EHR 存在误诊
verify_pk()	通过变量 *address_pk* 验证用户是否通过 IDM 的身份认证
verify_url()	通过变量 *url_pk* 验证 *url* 的拥有者
request_EHR()	机构请求 EHR 的接口, 申请请求 EHR 的权限(即触发事件 *pow*(), 随机选举一个审计节点对请求者发起 PoW 挑战)
verify_request()	验证请求者是否获取请求该 EHR 的权限(即第2.3.2节中是否解决了 PoW 难题和获得审计节点的签名 sig_A)
request_smk_pk ()	完成 PoW 挑战获得审计节点的签名 sig_A 后请求者调用该函数, 验证签名后, 触发事件 *store_smk_pk*()
max_min_credit_attribute()	用于更新所有机构中信用度属性最大值和最小值的函数, 方便计算机构信用度时数据的归一化处理
punish_s1()	惩罚 *s*1 违规行为函数, 违规行为会被记录到信用度属性 *pk_credit* 中
punish_s2()	惩罚 *s*2 违规行为函数, 违规行为会被记录到信用度属性 *pk_credit* 中
caculate_credit()	计算机构信用度的函数

将 EHR 与患者的关系记录到变量 $url\_pk[url\_hash]$ 中

6) **return** EHR_TXID

7) **else**

8) **return** faile

9) **end if**

该过程的执行前提是患者已经完成第 2.2 节工作流程的②③两步, 得到必要的参数后患者调用合约函数 $contribute\_EHR()$ 创建共享 EHR 合约交易 EHR_TXID. 合约内部执行函数 $verify\_pk()$ 验证患者注册身份, 验证通过后, EHR 的相关信息存入 $EHR\_share[url\_hash]$ 变量中, 该变量指向 $EHR\_sharing$ 结构体, 同时合约为该 EHR 初始化白名单 $url\_whitelist[url\_hash]$, 将患者的信息以及解密密钥 $[smk]_{pk_p}$ 存入该白名单所指向的结构体 $pk\_whitelist$ 中. 最后, 合约调用内部函数 $set\_url\_pk()$ 将患者与 EHR 的关系记录到变量 $url\_pk[url\_hash]$ 中.

2.4.4 研究报告的共享

算法 4 由机构和审核节点执行, 输入的参数有研究报告索引 r_url, 索引的 hash 值 r_hash, 机构成员公钥 pk_I, 机构的以太坊地址 $Eth\_address_I$, 审核节点成员资格公钥 pk_A, 审核节点的以太坊地址 $Eth\_address_A$. 交易成功返回此次交易 ID: $Research\_TXID$; 失败返回 faile.

算法 4. 研究报告共享算法

输入. r_url, r_hash, pk_I, $Eth\_address_I$, pk_A, $Eth\_address_A$

1) 机构调用合约函数 $contribute\_research(r\_url, pk_I)$ 共享研究报告

2) **if** $verify\_pk(Eth\_address_I, pk_I)$ = false **then**

3) **return** faile

4) **end if**

5) 触发合约事件 $audit\_vote(r\_url, pk_I)$, 监听该事件的审计节点链下查看研究报告

6) **while** ($con\_research[r\_hash].voteCount < set\_threshold$ && $timestamp < set\_timestamp$)

7) **if** 审计节点审核通过 **then**

8) 审计节点调用合约函数 $agree\_research(r\_url, r\_hash)$

9) **if** $verify\_pk(Eth\_address_A, pk_A)$ = true && $address\_pk[Eth\_address_A].role$=$audit\_node$ && $research\_vote[pk_A][r\_hash]$=false **then**

10) $con\_research[r\_hash].voteCount$++, $research\_vote[pk_A][r\_hash]$=true //投票数增加, 更新 pk_A 已投票 r_hash

11) **end if**

12) **end if**

13) **end while**

14) **if** $con\_research[r\_hash].voteCount >= set\_threshold$ **then**

15) **return** $Research\_TXID$

16) **else**

17) **return** faile

18) **end if**

患者看病时无法避免误诊情况, 机构在发现患者共享的 EHR 存在误诊后, 可以进行修改, 并将研究报告发送到云端供审计节点审查. 随后机构调用合约函数 $contribute\_research()$ 共享研究报告. $contribute\_research()$ 会触发事件 $audit\_vote()$, 监听该事件的审计节点链下审核研究报告. 审计节点 pk_A 审核通过后调用合约函数 $agree\_research()$ 来增加 $con\_research[r\_hash].voteCount$ 的值, 重复投票不会增加投票数. $Research\_TXID$ 中的 $voteCount$ 需在规定时间 $set\_timestamp$ 内超过预先设定的阈值 $set\_threshold$. 同时链下机构将修改的 EHR 和对 EHR 的 hash 值的签名 sig_I 发送给 IDM, IDM 转发给患者, 患者重复算法 3 的过程重新共享更正后的 EHR, 并通过调用 $correct\_new\_url()$ 函数将更正后的 url 存入错误 EHR 的记录中.

2.4.5 EHR 的访问

算法 5 由请求患者 EHR 的机构执行. 输入的参数有患者的成员资格公钥 pk_p, EHR 索引 $hash$ 值 url_hash, 请求者成员资格公钥 pk_r, 请求者的以太坊地址 $Eth\_address$, 审计节点的签名 sig_A, 审计节点的成员资格公钥 pk_A. 交易成功返回交易 ID: $Requet\_TXID$, 失败则根据失败类型返回 $illegal\_request$, $forged$, faile.

如果机构在患者 EHR 白名单中, 则可以直接查询到存储的解密密钥. 否则机构执行该算法前需要获取申请该 EHR 的权限, 所以机构需要先调用函数 $request\_EHR()$ 获取审计节点的签名 sig_A. 拿到签名 sig_A 后机构调用合约函数 $request\_smk\_pk()$, 智能合约执行函数 $verify\_pk()$,$verify\_url()$,$verify\_request()$ 验证请求者是否存在违规行为, 其中第一个验证函数检测 $s1$ 违规行为, 其余两个验证函数检测 $s2$ 违规行为. 一旦验证出请求者存在违规, 就会计算审计节点的共谋概率 η , η 超过阈值则会触发事件 $collusion()$ 通知 IDM, IDM 链下给与共谋双方惩罚. 通过验证以及患者访问控制策略 $strategy[pk_p]$ 后, 合约触发事件 $store\_smk\_pk()$, IDM 通知患者生成重加密密钥 $[K]_{p\text{-}r}$, 患者将重加密密钥发送到云端, 云端执行代理重加密后将解密密钥

$[smk]_{pk_r}$ 发给请求者.

算法 5. EHR 访问算法

输入. pk_p, url_hash, pk_r, $Eth\_address$, sig_A, pk_A

1) 请求者调用合约函数 $request\_smk\_pk(pk_p, url\_hash, pk_r, sig_A)$

2) **if** $verify\_pk(Eth\_address, pk_r) = forged$ **then**

3) 计算 η, 判断 η 是否超过阈值, 超过则触发事件 $collusion()$

4) **return** $forged$

5) **end if**

6) **if** $url\_whitelist[url\_hash][is\_white][pk_r]$=true **then**

7) pk_r 的访问时间戳 $timestamp$ 更新

8) **return** $Requet\_TXID$, $[smk]_{pk_r}$

9) **end if**

10) **if** $verify\_url(url\_hash, pk_p) =$ false **then**

11) 计算 η, 判断 η 是否超过阈值, 超过则触发事件 $collusion()$

12) **return** $illegal\_request$

13) **end if**

14) **if** $verify\_request(sig_A, pk_A) =$ false **then**

15) 计算 η, 判断 η 是否超过阈值, 超过则触发事件 $collusion()$

16) **return** $illegal\_request$

17) **end if**

18) **if** 请求者满足 $strategy[pk_p]$ **then**

19) pk_r 相关信息被添加到 EHR 的白名单 $url\_whitelist[url\_hash]$ 中 //请求时间戳 $timestamp$, 审计节点的签名 sig_A, 第一次请求成功时信用度

20) 触发合约事件 $store\_smk\_pk([smk]_{pk_p}, pk_p, pk_r)$

21) IDM 转发事件信息给患者, 患者生成重加密密钥 $[K]_{p\text{-}r}$ 并交由云端执行代理重加密 $//[smk]_{pk_p}$ 转为 $[smk]_{pk_r}$

22) **return** $Requet\_TXID$, $[smk]_{pk_r}$

23) **else**

24) **return** faile

25) **end if**

2.4.6 信用度属性收集

算法 6 由区块链中的共识节点执行. 输入的参数有新区块 $new\_blockId$, 机构成员公钥 pk_I 和区块中的合约交易 $TXID$. 该算法应该包含在算法 3 ~ 5 中, 为了方便说明, 将所有涉及改变机构信用度属性的交易集中到一起说明.

为了给患者营造可信任的 EHR 共享环境, 在智能合约中我们提供了机构信用度属性的实时更新和计算. 共识节点生成新区块时, 智能合约在处理这些交易时会自动更新交易中相关机构的信用度属性信息, 即图 2 中的行为特征的值.

算法 6. 信用度属性收集算法

输入. $new\_blockId$, pk_I, $TXID$

1) 共识节点构造新的区块 $new\_blockId$

2) **for** $TXID$ in $new\_blockId$ **do**

3) **if** $TXID$ 是 EHR 的共享 **then**

4) $pk\_credit[pk_I].EHR\_counts$++ //EHR 出自机构 pk_I

5) **end if**

6) **if** $TXID$ 是研究报告的共享 **then**

7) $pk\_credit[pk_{Ia}].research\_counts$++, $pk\_credit[pk_{Ia}].correct\_counts$++, $pk\_credit[pk_{Ib}].error\_counts$++ // pk_{Ia} 表示贡献研究报告的机构, pk_{Ib} 表示给患者诊断错误的机构

8) **end if**

9) **if** $TXID$ 是请求 EHR **then**

10) $pk\_credit[pk_{Ic}].requested\_counts$++, $pk\_credit[pk_{Id}].audit\_counts$++ // pk_{Ic} 表示 EHR 出自哪家机构, pk_{Id} 表示请求 EHR 交易中的审计节点

11) **end if**

12) **if** $TXID$ 是非法交易 **then**

13) **if** $s1$ 违规 **then**

14) 合约调用函数 $punish\_s1(pk_I)$, $pk\_credit[pk_I].s1\_counts$++, $pk\_credit[pk_I].s1\_blocknmb$ 更新为当前区块号, 调用函数 $caculate\_credit(pk_I)$ 将机构违规时的信用度记录到 $pk\_credit[pk_I].credit$ 中

15) **end if**

16) **if** $s2$ 违规 **then**

17) 合约调用函数 $punish\_s2(pk_I)$, $pk\_credit[pk_I].s2\_counts$++, $pk\_credit[pk_I].s2\_blocknmb$ 更新为当前区块号, 调用函数 $caculate\_credit(pk_I)$ 将机构违规时的信用度记录到 $pk\_credit[pk_I].credit$ 中

18) **end if**

19) **end if**

20) **end for**

3 理论与实验分析

本节对系统的模型特点、安全性、用户信用度以及区块链性能进行理论和实验分析. 我们在配置为 I7-4720HQ 处理器、12 GB 内存、1 TB 机械硬盘的 Windows 10 系统下进行实验. 我们给出了 EHR-SCAC 模型系统的实现, 如图 5 所示. 区块链维护方面使用 geth@1.8.3-stable, 智能合约的部署使用

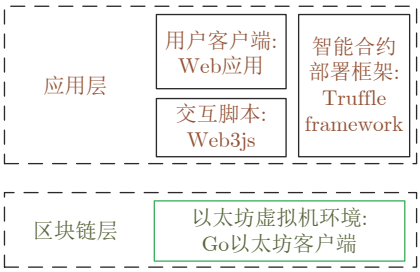


图 5 EHR-SCAC 系统整体框架

Fig. 5 EHR-SCAC system overall framework

truffle@v5.1.18. 使用 Web3.js@1.2.6 开发与区块链交互的前端. 数据分析采用 MATLAB R2018a.

通过 Go Ethereum[24] 我们建立了 PoA[25] 私有区块链. Web 客户端通过 Web3.js 发送交易并与用 Solidity[26] 编写的智能合约进行交互.

3.1 系统模型特点

与其他访问控制模型相比, EHR-SCAC 具有一定的优势. EHR-SCAC 采用模糊层次分析法, 提供具有参考性的信用度计算方法. 且机构信用度属性的动态变化使得机构的信用度具有实时性, 较文献 [15] 的组织成员访问控制和文献 [17] 中属性访问控制更具动态性. 患者可以根据合约中提供的参数, 动态地修改访问控制策略, 较文献 [12−13]的访问控制更加灵活, 智能合约的设计更轻量. 表 2 展示了 EHR-SCAC 模型与其他文献模型的对比.

表 2 EHR-SCAC 与其他模型功能特性的对比

Table 2 Comparison of the functional characteristics of EHR-SCAC and other models

模型	动态控制	数据完整性	身份认证	隐私保护	信用度分析	稳定性
文献 [12]	×	×	√	√	×	√
文献 [13]	×	×	√	√	×	√
文献 [15]	×	√	√	√	×	√
文献 [17]	×	√	√	√	×	√
本文模型	√	√	√	√	√	√

3.2 安全性分析

1) 隐私性. 本文利用密码学技术和区块链技术, 保证共享数据的隐私性, 患者 EHR 中的真实身份可以采用成员公钥 pk, 保证了用户的匿名性, 并且不会对监管造成困扰.

2) 不可伪造性. 在患者将 EHR 上传到云服务器之前, 医生需要签署 EHR. 当患者将 EHR 上传到云时, 必须将医生的签名一起上传. 这不仅划定了医疗纠纷责任归属, 还确保了电子病历的完整性.

3) 透明性. 请求者对 EHR 的访问会记录在区块链中. 请求 EHR 交易中记录了请求者的各项属性和请求时间, 这些属性无论是通过区块链获取还是通过预言机模式获取都是公开可信的. 患者可以在区块链中查看获取请求权限的访问者以及他们访问的时间, 保证了系统的透明性.

4) 稳定性. 我们采用 PoA 共识机制, 只有少数节点需要对区块进行验证, 随着区块链规模的扩大, 这项措施可以降低区块的挖掘成本. 同时, 我们的信用度奖惩机制和监督机制有效地阻止了节点的无效交易攻击行为, 维护了区块链的稳定性.

3.3 用户信用度分析

第 2.3.1 节中介绍了信用度的计算方法, 使用 FAHP 对行为特征的重要性程度进行划分, 最后计算出各部分的特征权重和特性权重. 其中: 功能特性 P 的重要性划分为 $p2 > p1 = p3 > p4 = p5$,可靠特性 R 的重要性划分为 $r1 = r2$, 安全特性的重要性划分为 $s1 > s2$, 特性的重要性划分为 $S > R > P$. 通过 MATLAB 计算出各特性下特征的权重以及特性的权重分别为

$$\boldsymbol{w}_P = [0.2250, 0.3000, 0.2250, 0.1250, 0.1250]$$

$$\boldsymbol{w}_R = [0.5000, 0.5000]$$

$$\boldsymbol{w}_S = [0.7500, 0.2500]$$

$$\boldsymbol{w}_F = [0.1667, 0.3333, 0.5000]$$

得到权重数据后, 就可以在智能合约写出信用度计算公式. 需要注意的是目前以太坊的智能合约还不支持浮点数运算, 所以在实际设计中需要将这些权重扩大为整数. 为了模拟现实应用场景, 我们预先配置了两家三级、两家二级和两家一级共 6 个授权的医院节点和一个 IDM 节点, 其中三个级别医院正面行为特征数值变化速度约为 3:2:1, 负面行为特征数值变化约为 1:2:3. 通过 Web3.js编写的客户端模拟用户发送合约交易到区块链的过程, 在共识节点处理这些合约交易时, 智能合约会记录各医院信用度属性的变化.

当 PoA 私有链运行一段时间后, 可以看到三种级别的医院信用度的波动范围, 如图 6 所示, 三个级别医院信用度最终分别在 0.94, 0.86 和 0.73 左右波动. 高等级的机构拥有雄厚的资金和人力去研究医学难题, 随着机构等级的降低, 机构信用度下降, 这不仅符合实验结果, 也符合实际情况.

待系统运行稳定后, 对三个级别的医院分别执行违规行为 $s1$ 操作, 信用度变化如图 7 所示, 其中三甲医院一开始信用度处于恒定不变的原因是该医院此阶段信用度属性是系统中最高的, 由于要对信

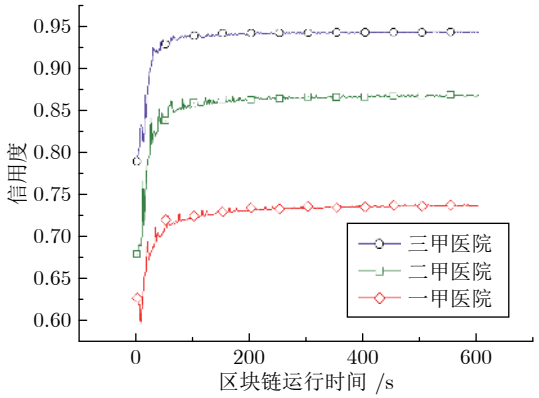


图 6　三种信用度变化趋势

Fig. 6　Three levels of credit changes

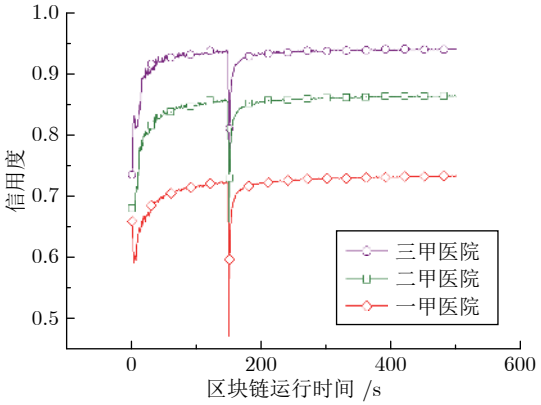


图 8　违规行为 $s2$ 信用度变化

Fig. 8　Changes in credit rating of $s2$ violation

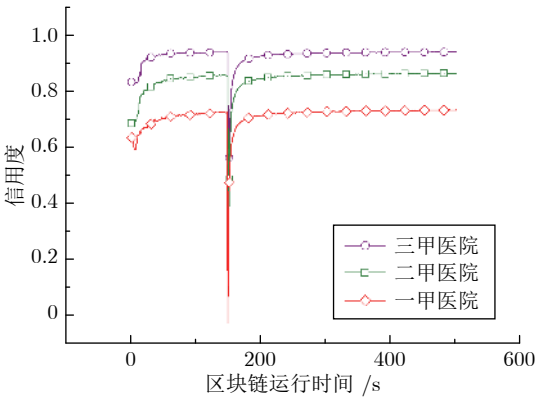


图 7　违规行为 $s1$ 信用度变化

Fig. 7　Changes in credit rating of $s1$ violation

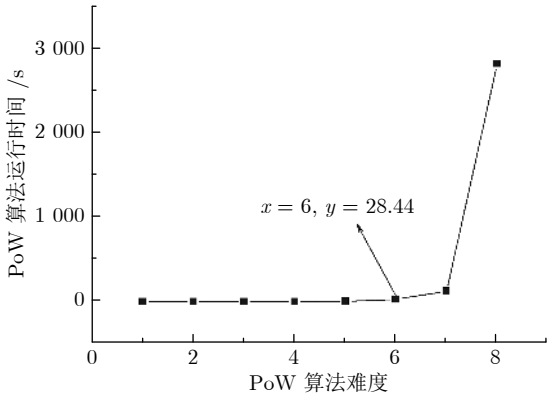


图 9　PoW 难度变化与运行时间

Fig. 9　PoW difficulty change and running time

用度属性进行归一化操作 (系统中信用属性最低值均为 0), 所以三甲医院各项信用度属性归一化后的值保持不变. 然后对三个级别的医院执行违规行为 $s2$ 操作, 信用度变化如图 8 所示. 从这两张图可以看出, 机构在做出违规行为 $s1$ 操作后, 信用度有明显的下降, 而做出违规行为 $s2$ 操作后, 机构信用度的下降不是很明显. 随着时间的推移, 三个级别医院的信用度逐渐恢复到原来的水平. 违规行为 $s2$ 操作对医院信用度惩罚不是很明显, 在合约中我们记录了医院违规行为的次数, 如式 (6) 中所示, 如果医院再进行一次违规行为 $s2$ 操作, 其信用度将快速下降.

为了避免 $s1$, $s2$ 行为攻击占用系统资源, 我们将信用度和请求 EHR 的难度相关联. 医院请求 EHR 时必须解决 PoW 难题. 经过测试, 难度变化与完成 PoW 难题的时间如图 9 所示. 根据图 9 的结果, 当医院进行违规操作时, 医院执行 PoW 算法的时间将迅速上升.

图 10 展示了三个等级医院正常信用度、$s1$ 违规一次、$s2$ 违规一次以及 $s2$ 违规两次情况下请求

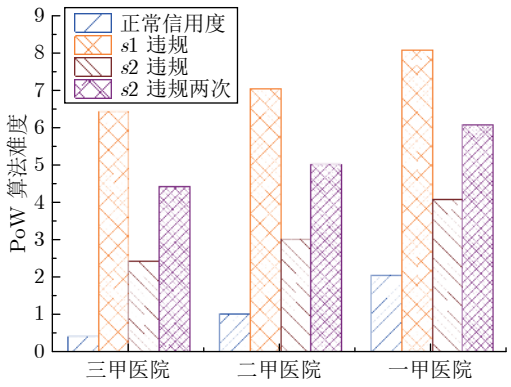
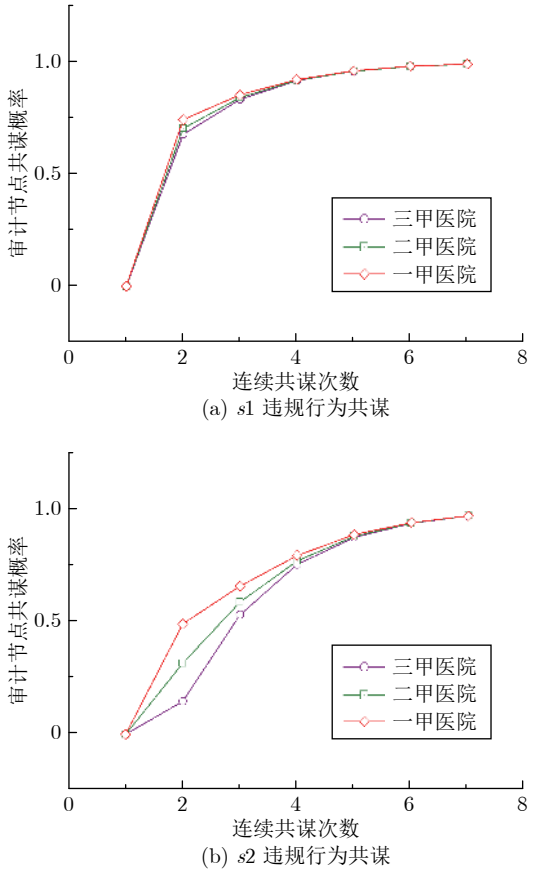


图 10　基于信用度的 PoW 难度

Fig. 10　PoW difficulty based on credit

EHR 交易的 PoW 难度对应图. 可以看到当恶意攻击发生时, 医院请求 EHR 的 PoW 难度增加.

在第 2.3.2 节中, 我们提到机构与审计节点存在共谋行为并设计了监测共谋行为的方法. 图 11 展示了机构与审计节点共谋攻击系统时, 系统监测



(a) $s1$ 违规行为共谋

(b) $s2$ 违规行为共谋

图 11 当机构与审计节点共谋攻击系统时, 共谋概率 η 的变化

Fig. 11 Changes in the collusion probability η when the organization and the audit node collude to attack the system

到的共谋概率 η 的变化. 从图 11 中可以看出, $s1$ 共谋攻击更容易被监测到. 图 11(b) 中高信用度的节点第 2 次执行 $s2$ 攻击后, 信用度惩罚不够明显. 但是, 高信用度节点不断发起 $s2$ 攻击后, 系统监测到的共谋概率持续上升. 所以, 我们制定的共谋监测制度可以很好地维护系统稳定.

3.4 区块链性能评估

我们在 PoA 私有链中设置了 6 个节点, 以 1 个节点挖矿, 其余 5 个节点不停地发送交易. 我们分别测试了发送共享 EHR 和请求 EHR 交易的处理时间. 同时对 PoA 私有链出块时间进行了更改, 通过出块时间与区块中交易数的比值来观察交易的处理速度. 从表 3 中可以看出交易平均处理时间在一定的范围内波动, 出块速度对于交易处理速度没有影响. 由于出块速度的快慢会影响用户客户端的响应速度, 所以更快的出块速度有利于及时向用户反馈交易结果.

表 3 不同出块时间区块吞吐量和交易处理速度

Table 3 Block throughput and transaction processing speed at different block generation times

出块时间 (s)	平均每个区块交易数	交易吞吐量 (个数/s)	交易处理速度 (ms)
1	110.5	110.5	9.04
2	222.5	111.25	8.98
3	332	110.6	9.03
4	439	109.75	9.11
5	551	110.2	9.07
6	658	109.6	9.11
7	764	109.1	9.16
8	875	109.3	9.14
9	986	109.5	9.12
10	1096	109.6	9.12

本文是通过智能合约实现访问控制, 而文献 [14, 16] 主要使用密码学技术实现访问控制, 与本文差异较大, 性能上不适合进行比较. 文献 [12–13] 与本文均采用智能合约进行访问控制的设计. 文献 [12–13] 通过触发合约事件通知代理重加密节点, 使用同态代理重加密技术为请求者重新加密解密 EHR 的对称密钥 $[smk]_{pk}$. 不同于文献 [12–13], 本文由患者生成重加密密钥 $[K]_{p\text{-}r}$, 再由云端链下为请求者重新加密解密 EHR 的对称密钥. 文献 [13] 与文献 [12]的智能合约类似, 所以我们仅讨论文献 [12] 中的交易流程. 为了更好地研究本文智能合约访问控制性能, 我们将节点登记、患者共享 EHR 和请求者请求 EHR三个主要交易的交易反馈时延与文献 [12] 进行比较.

为了方便比较, 将区块生成时间、同态代理重加密时间、本文的代理重加密时间分别记为 T_{BT}, T_{HT}, T_{RT}. 对文献 [12] 和本文节点登记、患者共享 EHR 和请求者请求 EHR 的分析如下:

1) 文献 [12] 中节点注册首先由患者所在的医疗提供商调用智能合约创建节点登记交易, 然后智能合约触发事件让投票节点验证新节点, 投票节点创建交易写入验证结果, 最后智能合约触发事件通知患者节点, 患者节点创建交易写入是否同意加入区块链, 所以文献 [12] 的节点登记至少需要 $3T_{BT}$ 的时间. 本文节点登记需要用户调用合约创建一笔交易, 即至少需要 T_{BT} 的时间.

2) 文献 [12] 中患者共享 EHR 时, 由医疗提供商调用智能合约创建交易将 EHR 相关信息写入区块链. 本文患者共享 EHR 时, 由患者调用智能合约创建交易将 EHR 相关信息写入区块链中, 与文献 [12]

类似, 均至少需要 T_{BT} 的时间.

3) 文献 [12] 中请求者请求患者 EHR 时, 首先需要患者 EHR 所在的医疗服务商调用智能合约创建交易验证请求者身份. 然后触发合约事件, 代理重加密节点需要响应事件创建交易将各自加密的大素数 p 发送给智能合约, 智能合约利用同态加密方法将这些大素数 p 生成主密钥 *master-p*, 并用请求者的公钥加密, 同时智能合约使用同态加密方法用 *master-p* 加密 *smk* (用于加密 EHR 的对称密钥) 并发送给代理节点. 代理节点最后解密自己的部分, 然后再创建交易发送给智能合约将部分解密后的盲信息写入区块链, 最后由智能合约利用同态法计算出解密 EHR 的对称密钥. 所以在文献 [12] 中请求 EHR 的交易的反馈时延至少需要 $3T_{BT}+T_{HT}$. 在本文中, 请求者需要调用两次合约, 一个是获取审计节点的签名 sig_A, 另一个是请求 EHR 的解密密钥, 最终由云端执行代理重加密将解密密钥发给请求者.

由上述分析可知, 本文基于智能合约的 EHR 共享访问控制模型较文献 [12−13] 交易反馈延时更具优势, 具体比较如表 4 所示.

表 4　区块链交易时延比较

Table 4　Blockchain transaction delay comparison

交易类型	节点登记	EHR共享	请求EHR
本文模型	T_{BT}	T_{BT}	$2T_{BT}+T_{RT}$
文献 [12−13]	$3T_{BT}$	T_{BT}	$3T_{BT}+T_{HT}$

4　结束语

在当今信息爆炸的时代, 数据共享对各行各业的发展起着至关重要的作用. 本文就医疗行业中 EHR 共享的特点和安全隐私问题, 提出一种基于集成信用度评估智能合约的安全数据共享访问控制模型. 我们将信用度评估和访问控制策略集成到智能合约中, 并提出信用度奖惩机制和共谋监督机制来维护区块链的安全和稳定. 同时, 借助区块链所具有的不可篡改、可追溯、透明性的特点, 为患者提供了安全可信的 EHR 共享平台. 实验表明, 该模型可以有效地分析用户行为, 动态更新用户的信用度属性和动态控制 EHR 的访问. 未来我们将研究多链合作, 将信用度、审计、共享等分隔开来, 形成各功能领域的去中心化和全局上的统一.

References

1 Cao S, Zhang G X, Liu P F, Zhang X S, Neri F. Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain. *Information Sciences*, 2019, **485**: 427−440

2 Eman A K, Nader M, Jameela A J. E-health cloud: Opportunities and challenges. *Future Internet*, 2012, **4**(4): 621−645

3 Meingast M, Roosta T, Sastry S. Security and privacy issues with health care information technology. In: Proceedings of the 2006 IEEE International Conference on Engineering in Medicine and Biology Society. New York, USA: IEEE, 2006. 5453−5458

4 Esposito C, Santis A D, Tortora G, Chang H, Choo K K R. Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Computing*, 2018, **5**(1): 31−37

5 Liu X H, Liu Q, Peng T, Wu J. Dynamic access policy in cloud-based personal health record (PHR) systems. *Information Sciences*, 2017, **379**: 62−81

6 Liu X J, Xia Y J, Yang W, Yang F L. Secure and efficient querying over personal health records in cloud computing. *Neurocomputing*, 2018, **274**: 99−105

7 Au M H, Yuen T H, Liu J K, Susilo W, Huang X Y, Xiang Y, Jiang Z L. A general framework for secure sharing of personal health records in cloud system. *Journal of Computer and System Sciences*, 2017, **90**: 46−62

8 Singh A, Chandra U, Kumar S, Chatterjee K. A secure access control model for e-health cloud. In: Proceedings of TENCON 2019−2019 IEEE Region 10 Conference (TENCON). Kochi, India: IEEE, 2019. 2329−2334

9 Yuan Yong, Wang Fei-Yue. Blockchain: the state of the art and future trends. *Acta Automatica Sinica*, 2016, **42**(4): 481−494
(袁勇, 王飞跃. 区块链技术发展现状与展望. 自动化学报, 2016, **42**(4): 481−494)

10 Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: Using blockchain for medical data access and permission management. In: Proceedings of the 2nd International Conference on Open and Big Data (OBD). Vienna, Austria: IEEE, 2016. 25−30

11 Xue Teng-Fei, Fu Qun-Chao, Wang Cong, Wang Xin-Yan. A medical data sharing model via blockchain. *Acta Automatica Sinica*, 2017, **43**(9): 1555−1562
(薛腾飞, 傅群超, 王枞, 王新宴. 基于区块链的医疗数据共享模型研究. 自动化学报, 2017, **43**(9): 1555−1562)

12 Dagher G G, Mohler J, Milojkovic M, Marella P B. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustainable Cities and Society*, 2018, **39**: 283−297

13 Daraghmi E Y, Daraghmi Y, Yuan S. MedChain: A design of blockchain-based system for medical records access and permissions management. *IEEE Access*, 2019, **7**: 164595−164613

14 Zhang Chao, Li Qiang, Chen Zi-Hao, Li Zu-Rui, Zhang Zhen. Medical chain: Alliance medical blockchain system. *Acta Automatica Sinica*, 2019, **45**(8): 1495−1510
(张超, 李强, 陈子豪, 黎祖睿, 张震. Medical Chain: 联盟式医疗区块链系统. 自动化学报, 2019, **45**(8): 1495−1510)

15 Xia Q, Sifah E B, Smahi A, Amofa S, Zhang X. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information*, 2017, **8**(2): 44−59

16 Tang F, Ma S, Xiang Y, Lin C L. An efficient authentication scheme for blockchain-based electronic health records. *IEEE Access*, 2019, **7**: 41678−41689

17 Liu J W, Li X L, Ye L, Zhang H L, Mohsen G. BPDS: A blockchain based privacy-preserving data sharing for electronic medical records. In: Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi, United Arab Emirates: IEEE, 2018. 1−6

18 Ethereum Whitepaper [Online], available: https://ethereum.org/en/whitepaper/, April 5, 2020

19 Ripple interLedger protocol [Online], available: https://interledger.org/overview.html, April 5, 2020

20 Zhang Kai, Pan Xiao Zhong. Access control model based on user behavior trust in cloud computing. *Journal of Computer Applic-*

ations, 2014, **34**(4): 1051−1054
(张凯, 潘晓中. 云计算下基于用户行为信任的访问控制模型. 计算机应用, 2014, **34**(4): 1051−1054)

21 Wang Hai-Yong, Pan Qi-Qing, Guo Kai-Xuan. Access control model based on blockchain and user credit. *Journal of Computer Applications*, 2020, **40**(6): 1674−1679
(王海勇, 潘启青, 郭凯璇. 基于区块链和用户信用度的访问控制模型. 计算机应用, 2020, **40**(6): 1674−1679)

22 Huang J Q, Kong L H, Chen G H, Wu M Y, Liu X, Zeng P. Towards secure industrial iot: blockchain system with credit-based consensus mechanism. *IEEE Transactions on Industrial Informatics*, 2019, **15**(6): 3680−3689

23 Web3.js−Ethereum JavaScript API [Online], available: https://github.com/ethereum/web3.js/, April 5, 2020

24 Go Ethereum [Online], available: https://geth.ethereum.org/downloads/, April 5, 2020

25 Proof-of-authority [Online], available: https://www.poa.network/, April 5, 2020

26 Solidity document [Online], available: https://soliditycn.readthedocs.io/zh/latest/, April 5, 2020

张乐君 扬州大学教授. 分别于 2005 年获得哈尔滨工业大学硕士学位, 于 2008 年获得哈尔滨工程大学博士学位. 主要研究方向为计算机网络, 社会网络分析, 信息安全.
E-mail: zhanglejun@yzu.edu.cn
(**ZHANG Le-Jun** Professor at Yangzhou University. He received his master degree in computer science and technology at Harbin Institute of Technology in 2005 and his Ph.D. degree in computer science and technology at Harbin Engineering University in 2008. His research interest covers computer network, social network analysis, and information security.)

刘智栋 扬州大学信息工程学院硕士研究生. 主要研究方向为区块链应用与访问控制.
E-mail: yzu_liuzhidong@163.com
(**LIU Zhi-Dong** Master student at the College of Information Engineering, Yangzhou University. His research interest covers blockchain application and access control.)

谢 国 西安理工大学自动化与信息工程学院教授. 主要研究方向为智能信息分析与故障诊断. 本文通信作者.
E-mail: guoxie@xaut.edu.cn
(**XIE Guo** Professor at the College of Automation and Information Engineering, Xi′an University of Technology. His research interest covers intelligent information analysis and fault diagnosis. Corresponding author of this paper.)

薛 霄 天津大学智能与计算学部教授. 主要研究方向为服务计算, 计算实验, 群体智能.
E-mail: jzxuexiao@tju.edu.cn
(**XUE Xiao** Professor at the College of Intelligence and Computing, Tianjin University. His research interest covers service computing, computational experiment, and complex network.)