计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(12):9−17 [doi: 10.15888/j.cnki.csa.006674]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于区块链的供应链可信数据管理①

黄宇翔, 梁志宏, 黄　芯, 孙永科

(西南林业大学 大数据与人工智能研究院, 昆明 650224)
通讯作者: 梁志宏, E-mail: zhliang@swfu.edu.cn

摘　要: 为解决传统供应链中贸易数据潜在伪造、篡改等安全问题, 提出了一种基于区块链技术的供应链可信数据管理方案. 首先, 以智能合约将贸易规则代码化, 防范履约风险和提高贸易数据处理可信性; 其次, 采用 ZSS04 方案和抽样技术交互完成贸易数据完整性检验; 再次, 设计了适用于供应链的分布式共识机制, 以提高贸易数据存储可信性; 最后, 利用区块链技术所具有的原生特性实现非可信环境下的可信数据管理. 分析以及实验结果表明, 该方案能够为供应链中贸易数据管理提供新的思路和技术支持.

关键词: 供应链; 可信数据管理; 区块链; 智能合约; 密码学; 共识机制

引用格式: 黄宇翔,梁志宏,黄芯,孙永科.基于区块链的供应链可信数据管理.计算机系统应用,2018,27(12):9–17. http://www.c-s-a.org.cn/1003-3254/6674.html

Supply Chain Trustworthy Data Management Based on Blockchain

HUANG Yu-Xiang, LIANG Zhi-Hong, HUANG Bi, SUN Yong-Ke

(Institute of Big Data and Artificial Intelligence, Southwest Forestry University, Kunming 650224, China)

Abstract: In order to solve the security problems such as potential tampering and forgery in the traditional supply chain of transaction data, this study proposes a new trusted data management scheme based on blockchain technology. First, an intelligent contract suitable for supply chain is designed to code trade rules in order to prevent performance risks and improve the credibility of trade data processing. Then, ZSS04 scheme and sampling techniques are used to complete transaction data integrity check. Furthermore, a distributed consensus mechanism for supply chain is designed to improve the reliability of trade data storage. Finally, the trusted data management in untrusted environment is realized by using the native characteristics of blockchain. Results show that the scheme can provide new ideas and technical support for transaction data management in supply chain.

Key words: supply chain; trusted data management; blockchain; smart contract; cryptography; consensus mechanism

自 14 世纪欧洲文艺复兴, 资本主义萌芽确立, 全球贸易一直是人类历史上最强大的“财富创造者”. 国务院办公厅 2017 年 10 月发布《关于积极推进供应链创新与应用的指导意见》(国办发〔2017〕84 号) 中把"积极稳妥发展供应链金融"列为重点任务. 然而, 在推动供应链的快速发展的实践中, 须解决的关键问题是要保证交易数据的可信[1]: 换言之, 即在组织合作关系中, 在没有监视或控制的过程, 贸易主体间所提供的交易信息是真实可靠的. 多年来, 为了建立贸易中的信任机制, 信托机构和信托工具应运而生. 然而, 这种传统中心化的第三方数据管理方式缺乏技术可信度, 且存在效率低下、成本高昂和易受攻击等诸多不可控因素[2,3]. 这些问题都严重制约了供应链的健康发展. 因此, 解决供应链中交易数据可信问题在当今经济高速

① 基金项目: 国家自然科学基金 (61702442)
Foundation item: National Natural Science Foundation of China (61702442)
收稿时间: 2018-05-05; 修改时间: 2018-05-24; 采用时间: 2018-06-11; csa 在线出版时间: 2018-12-03

发展的时代尤为重要, 关系到社会主义市场经济的繁荣.

从数据管理的角度分析, 一个可信的数据库管理系统[4]是要从三个层面来保证系统的可信, 即存储的可信性、处理的可信性和外部访问的可信性, 如图 1 所示. 本文在研究供应链中可信数据管理的过程中, 重点关注于数据存储的可信性和处理的可信性, 一旦交易完成, 则不会出现被攻击者恶意篡改或交易信息丢失的情况.

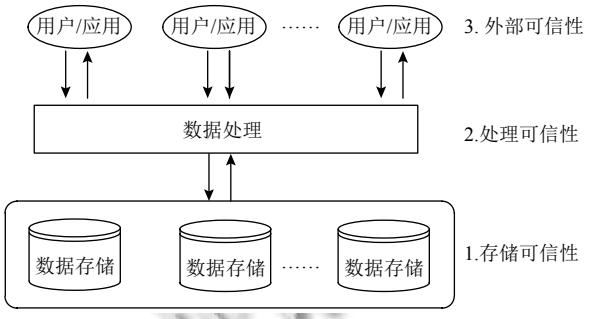

图 1 可信数据管理示意图

近年来, 学术界针对供应链可信数据管理的研究大部分主要集中在 B2B 与 B2C 模式上, 然而也面临着诸多交易信任问题. 例如, 文献[5]通过借助于大数据技术来捕获交易主体交易信息的证明依据, 但是该方法的缺点在于数据来源的局限性, 无法获得完整的交易链数据, 并且由于数据源的独立性, 攻击者借助大数据技术对交易数据进行非授权改动的可能性增大. 文献[6]提出将 B2B+B2C 供应链上的一部分放置在 ERP 中, 然而 ERP 是企业管理中非常复杂的信息管理系统, 无法真正有效确保交易数据的完全可信. 因此, 迫切需要从技术层面给出一种操作性强的供应链可信数据管理方法.

针对这些挑战, 本文基于区块链技术[7], 提出了一种新的供应链可信数据管理方案, 以保证交易数据的处理可信性和存储可信性. 在交易初始阶段, 贸易主体之间可以自由定制符合各方利益的交易规则, 引入交易中心, 并采用智能合约[8]将交易规则计算机代码化; 在交易确认阶段, 采用 ZSS04 短签名方案[9]对交易数据取验证标签, 可以利用同态标签[10]的同态性来保持交易数据结构的不变; 在交易验证阶段, 引入审核中心, 审核中心基于隐私保护数据持有性证明[11]和抽样技术[12]来完成交易数据完整性轻量级验证工作. 最后, 在登记上链阶段, 基于区块链的原生特性 (防篡改、无中心、强一致性等) 实现交易结果与相关交易证据在链上的高度钩稽, 以确保交易结果的高可信. 本文力图从技术上杜绝传统中心化数据管理模式下的数据易篡改、低可信, 确保供应链中各参与方的利益, 为稳定供应链生态提供技术支持.

本文第 1 节简要介绍双线性对、ZSS04 短签名方案、同态标签、智能合约和区块链; 第 2 节提出一种供应链可信数据管理方案; 第 3 节对供应链可信数据管理方案进行可行性分析和共识算法验证; 最后总结全文并对下一步工作进行展望.

1 预备知识

1.1 双线性对

定义 1. 假设G_1是一个阶为质数q的乘法循环群, G_2是一个阶为质数q的乘法循环群, G_1的一个生成元为p. 假设G_1、G_2上的离散对数问题都是困难的, 若满足以下三个性质, 则把映射$e:G_1\times G_1\to G_2$称之为双线性映射.

(1) 双线性性: 对于$\forall Q\in G_1, \{a,b\}\subseteq Z_q^*$有$e(ap,bQ)=e(p,Q)^{ab}$;

(2) 非退化性: $\exists\{p,Q\}\subseteq G_1$, 满足$e(p,Q)\neq 1$;

(3) 可计算性: 对于$\forall\{Q,R\}\subseteq G_1$, $\exists$多项式时间算法能够计算$e(Q,R)$.

1.2 ZSS04 短签名方案

假设乘法循环群G的阶为质数n.

(1) 离散对数问题 (DLP) 定义为: 令$a\in Z_n^*$, $\{g,g^a\}\subseteq G$, 求a;

(2) 计算 Diffie-Hellman 问题 (CDH) 定义为: 令$\{a,b\}\subseteq Z_n^*$, $\left\{g,g^a,g^b\right\}\subseteq G$, 求$g^{ab}$;

(3) 判定 Diffie-Hellman 问题 (DDH) 定义为: 令$\{a,b,z\}\subseteq Z_n^*$, $\left\{g,g^a,g^b,g^z\right\}\subseteq G$, 判定$z=ab \bmod n$是否成立.

假设$e:G\times G\to G_T$的双线性映射 (此处的G_T亦为阶为质数n的乘法循环群), 那么G中的 DLP 可归纳为计算G_T中的 DLP. 然而, G中的 DDH 不再困难: 令$\{a,b,z\}\subseteq Z_n^*$, $\left\{g,g^a,g^b,g^z\right\}\subseteq G$, 判定等式$e(g,g^z)=e\left(g^a,g^b\right)$是否成立即能解决$G$中 DDH 问题. 求逆 Diffie-Hellman 问题 (Inv-CDHP: $g,g^z\Rightarrow g^{1/z}$) 与平方 Diffie-Hellman 问题 (Squ-CDHP: $g,g^z\Rightarrow g^{z\times z}$) 是 CDHP 的两个变型.

ZSS04 短签名方案是由 Zhang 等人[9]所提的基于

双线性对的短签名方案, 其构造和安全性是基于求逆 Diffie-Hellman 问题 (Inv-CDHP), 这个问题等价于 CDHP. ZSS04 方案要比 BLS 方案更加有效, 并不需要特殊的散列函数. ZSS04 短签名方案由 4 个算法构成: 参数生成算法 *ParamGen*, 密钥生成算法 *KeyGen*, 签名生成算法 *Sign*, 签名验证算法 *Ver*, 具体如下:

(1) *ParamGen*: 系统参数为$\{G_1, G_2, e, q, p, H\}$;

(2) *KeyGen*: 挑选随机数$x \in Z_q^*$, 求$p_{pub} = xp$. 其中, p_{pub}为公钥, x为私钥;

(3) *Sign*: 给定私钥x, 消息内容m, 求$S = p(H(m)+x)^{-1}$, S即为签名;

(4) *Ver*: 给定公钥p_{pub}, 消息内容m和签名S, 证明等式$e\left(H(m)p + p_{pub}, S\right) = e(p,p)$是否成立.

1.3 同态标签

同态是一种映射关系$f: A \to B$, 即:

$$f(a*b) = f(a)\cdot f(b)$$

其中, $*$是 A 上的运算, $\cdot$ 是 B 上的运算.

由同态思想生成同态标签, 利用同态标签的同态性来完成贸易数据完整性检测. 若满足下列 2 个性质即为同态标签.

(1) 对$\forall$数据块d_i与d_j, $d_i + d_j$的标签信息$T\left(d_i + \mathrm{d}_j\right)$可由其本身标签信息得出, 即$T\left(d_i + d_j\right) = T(d_i) * T\left(d_j\right)$.

(2) 通过使用同态标签, 不用对所有的数据块进行验证, 只需使用少量特定数据块即可验证整个数据的完整性, 从而减少算力, 提高效率.

1.4 智能合约

智能合约[13]最早由密码学家 Nick Szabo 提出: 一个智能合约就是计算机协议, 它促进、检验或执行合约的协商、履行, 或者使合约条款不必要, 支持进行图灵完备的计算. 事实上, 其工作原理犹如计算机程序设计语言中的条件语句 if-then. 如果条件满足, 智能合约就会被自动触发, 则执行相应条款, 否则不执行.

1.5 区块链

区块链是一项全新的分布式记账系统[14], 多个独立节点共同参与维护, 具有不可更改、无中心、可追溯等原生特性. 区块链将数据分成不同的区块, 每个区块是由块头和块身这两部分组成, 块头存放前驱区块的哈希值, 块身则负责存储数据. 区块之间前后依次钩稽, 形成一条完整数据链, 如图 2 所示.

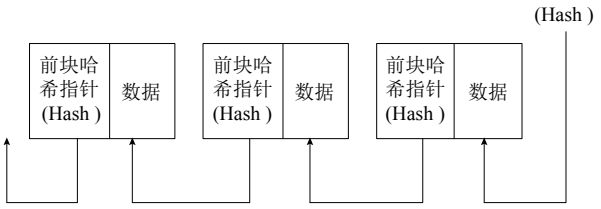

图 2 区块链示意图

区块链一般分为公有链、私有链和联盟链 3 类, 本文采用的是 Ethereum[15]项目下的联盟链. Ethereum 具有以下特性: 1) 智能合约支持图灵完备性, 设计了 256 位计算环境—以太坊虚拟机 (EVM), 并支持使用 Serpent, Solidity, LLL 类编程语言创建应用. 2) 通过叔块 (uncle block) 激励机制, 从而减少矿池, 使区块产生间隔由 10 minutes 缩短至 15 seconds, 并支持 PoW、PoS 共识算法. 3) 为避免外界恶意循环执行攻击, 通过 Gas 来控制代码执行的指令次数.

2 供应链可信数据管理方案设计

2.1 系统模型

方案由四个主体单元组成: 贸易主体 *P*, 交易中心 *S*, 审核中心 *B*, Ethereum 区块链商业网络 *C*. 系统模型如图 3 所示.

本方案的流程具体表述如下:

步骤 1. 方案中的每个主体单元先向证书机构 (Certificate Authority, CA) 完成认证工作, 并申请公钥数字证书.

步骤 2. 贸易主体 P 之间自由制定符合各方利益的交易规则, 交易中心 S 运用智能合约将交易规则计算机代码化, 当交易行为合法时, 智能合约将被触发, 交易生成. 采用 ZSS04 短签名方案对交易数据取验证标签, 并把验证标签、交易主体信息等与交易相关的内容发给审核中心 B.

步骤 3. 审核中心 B 通过贸易主体 P 的公钥来验证签名的真伪, 若为 true, 则向交易中心 S 发出证据挑战请求.

步骤 4. 交易中心 S 将证据发送给审核中心 B, 审核中心 B 验证双线性性等式成立与否. 若等式成立, 审核中心 B 和交易中心 S 将挑战证据、智能合约代码、相关交易信息以 JSON 的格式封装后发给贸易主体 P 进行签名.

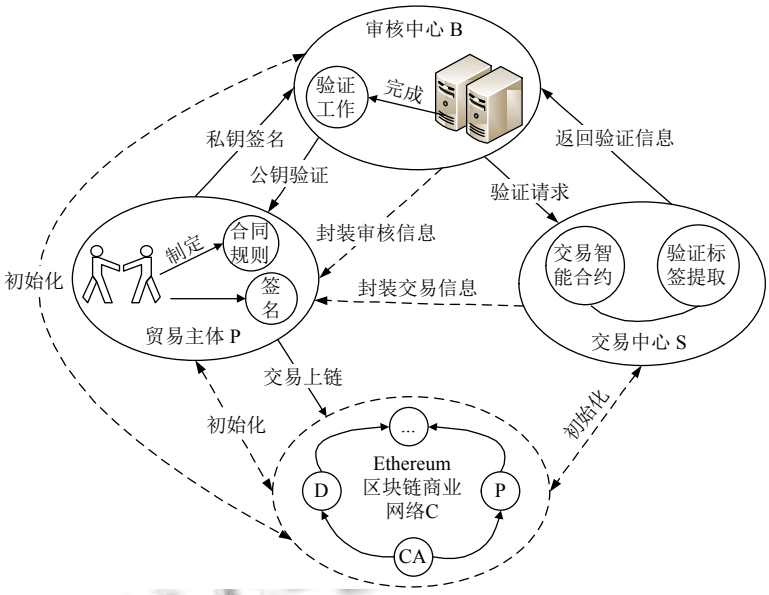

图 3 系统模型示意图

步骤 5. 将贸易主体 P 签名后的交易发送至 Ethereum 区块链商业网络 C, 通过分布式多节点共识机制算法将交易信息写入 Ethereum 区块链网络.

2.2 方案构造

2.2.1 参数

方案中的参数及其含义:

1) 贸易主体 *P*: 供应链上的一级、二级、三级等供应商/经销商、金融机构, 交易活动的主要参与者.

2) 交易中心 S: 负责交易的生成 (智能合约) 和一系列密钥生成工作, 为交易数据提供短暂存储.

3) 审核中心 B: 负责密钥验证和交易数据完整性审核事宜执行的领域专业知识机构.

4) Ethereum 区块链商业网络 C: 把审核中心 B 与交易中心 S 一起创建的数据登记上链. 这一网络中包含所有的共识节点单元 D、贸易主体 P、注册中心 CA 等共识主体, 各主体单元的身份核实是通过 CA 完成的, CA 并对审核通过的主体单元分发公钥证书.

5) 令G_1、G_2都是阶为质数q的乘法循环群, 且它们之间存在双线性映射$e: G_1 \times G_1 \to G_2$, p为G_1的一个生成元, 密码散列函数$H_1: \{0,1\}^* \to G_1$, 密码散列函数$H_2: \{0,1\}^* \to Z_q^*$.

2.2.2 方案构造过程

本方案是由 4 个阶段构造完成: 交易初始阶段、交易确认阶段、交易验证阶段和登记上链阶段.

1) 交易初始阶段

① 首先, 贸易主体 P、交易中心 S、审核中心 B 和其他共识主体需要在注册中心 CA 进行身份信息核实. 若通过, 则为其签发用于识别、认证网络主体的证书. 与此同时, 初始化一个 Ethereum 区块链商业网络 C.

② 贸易主体 P 之间可以自由制定符合自身利益的交易智能合约. 以面向供应链中先货后款质押采购的智能合约模型为例, 如图 4 所示. 将供应链公司与供应商的采购规则、供应链公司与经销商的销售规则使用 Solidity 语言编写智能合约, 一个合约是由一组代码 (合约函数) 和数据 (合约状态) 构成, 并写入区块链分布式网络体系中. 贸易主体 P 在注册账户时注册中心 CA 会通过构造函数为其初始化一个合约使用身份信息 identity, 在贸易主体 P 调用合约时, 该合约会先验证交易发起者的身份信息, 验证通过则进行合约操作. 当合约中的某一条件被触发时, 则自动执行相应的合约条款.

交易规则合约设计如下:

输入: *tx*, the object of transaction
输出: if success, return transaction data else throw exception

```
1. Procedure contract(tx)
2. if identity. sender = true then
3. tx = new Transaction();
4. tx.Content = tx.Content;
5. tx.Time = now;
6. tx.Id = tx.Id;
7. tx.Quantity = tx.Quantity;
8. return transaction data;
9. end if
10. end Procedure
```

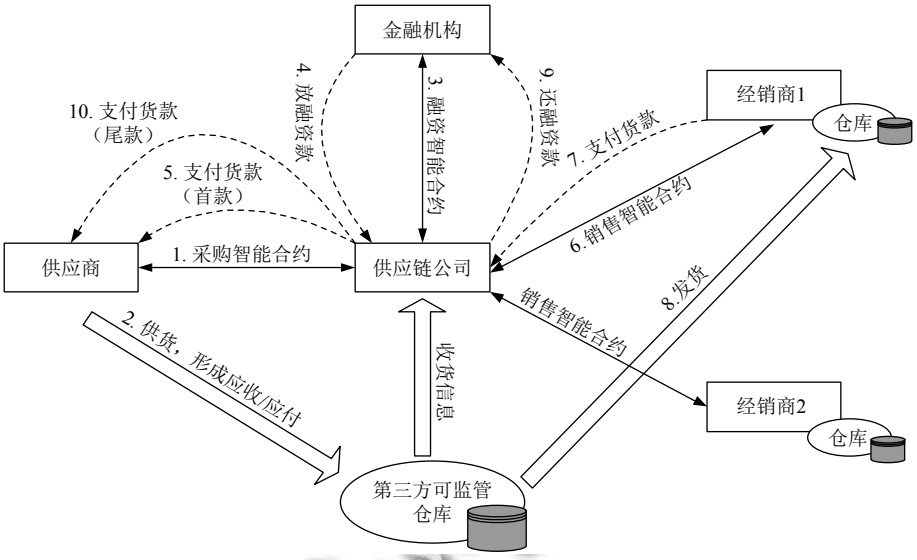

图 4 面向供应链中先货后款质押采购的智能合约模型

本方案的智能合约构造是基于图灵完备的 256 位计算环境—以太坊虚拟机 (EVM), 以 EVM 作为智能合约的运行环境, 可以进行多种类别的计算[16]. 智能合约运行于 EVM 中需要消耗一定数量的燃料 Gas, 故 Gas 则限定最大可运行计算指令. 设μ为当前网络状态, Λ为当前状态所剩余的 Gas, F为智能合约系统状态转移函数, $\bar{\mu}$为系统运行后的网络状态, $\bar{\Lambda}$为系统执行后的剩余的 Gas, ε为合约终止执行的条件列表, λ为记录序列, φ为合约执行后返回剩余的 Gas, T为合约输出的交易数据结果. 整个智能合约完成之后的状态转换函数表达式如式 (1):

$$\left(\bar{\mu},\bar{\Lambda},\varepsilon,\lambda,\varphi,T\right)=F\left(\mu,\Lambda,\lambda\right) \tag{1}$$

2) 交易确认阶段

① 交易中心 S 将交易数据 T 按属性分成i块, 即 $\{t_1,t_2,\cdots,t_i\}\subseteq Z_q^*$, q 是一个质数, 交易数据集合 $T=\left\{j|t_j,1\leqslant j\leqslant i\right\}$. 贸易主体 P 挑选一个随机的签名密钥对$(rpk,rsk)$, $x\in Z_q^*$, $y\leftarrow G_1$, 并计算公钥$p_{pub}\leftarrow xp$. 贸易主体 P 公开 $Param$: rpk, p_{pub}, p, i, j, y, $e(y,p_{Pub})$, 但保密 $Param$: x和rsk.

② 在交易中心 S 中, 对交易中的数据t_j取验证标签$\gamma_j\leftarrow p(x+H_2(\omega j)y^{t_j})^{-1}\in G_1$, 其中, $1\leqslant j\leqslant \mathrm{i}$, ωj是随机数$RN(RN\in Z_q^*$, 用于标识交易数据$T)$ 与j的连接. 数据验证标签记作集合$\xi=\left\{j|\gamma_j,1\leqslant j\leqslant \mathrm{i}\right\}$.

③ 为了使交易数据T标识符RN不会被恶意更改, 计算交易数据T的标签$tag=RN\|Sign_{rsk}(RN)$, 其中 $Sign_{rsk}(RN)$ 使用私钥rsk来对RN进行的签名.

3) 交易验证阶段

① 交易中心 S 将验证所需的数据 $(\xi,\ rpk,\ \mathrm{i},p_{pub})$ 发送到审核中心 B. 审核中心 B 接收验证数据后, 若外界再对交易数据进行非法篡改则会在证据挑战阶段被检测到, 从而保障交易数据 T 的安全性和完整性.

② 审核中心 B 使用公钥rpk来验证 $Sign_{rsk}(RN)$, 如果验证 success, 则输出RN信息; 如果 false, 则中止验证.

③ 审核中心 B 通过单向散列函数: g, 在交易数据T中的i个数据块里随机抽取m个块索引$\{x_1,\cdots,x_m\}$:

$$x_k=\left(\left(g^k(tag)\right)\bmod \mathrm{i}\right)+1,\{k|1\leqslant k\leqslant m\}. \tag{2}$$

其中,

$$g^k(tag)=\begin{cases} g(tag),k\in\{1\} \\ g\left(g^{k-1}(tag)\right),k\in[2,m]\end{cases}. \tag{3}$$

挑战请求:

$$\mathrm{Challenge}=\left\{\left(j,\Gamma_j\right)\middle|x_1\leqslant j\leqslant x_m,\Gamma_j\in Z_q^*\right\}. \tag{4}$$

式 (3) 发送给交易中心 S 后, 其计算:

$$\Phi=I+\zeta\cdot\sum_{j=x_1}^{x_m}t_j\Gamma_j \bmod q. \tag{5}$$

$$\gamma=\prod_{j=x_1}^{x_m}\gamma_j^{\Gamma_j}. \tag{6}$$

式 (4) 中$I \in Z_q^*$, $E = e\left(y, p_{pub}\right)^I \in G_2$, $\zeta = H_2(E) \in Z_q^*$.

最后, 交易中心 S 将γ, Φ, E作为验证凭证返还至审核中心 B, 审核中心 B 验证等式 $\zeta = H_2(E) \in Z_q^*$是否成立, 并通过式 (7) 验证接收的数据正确性.

$$\prod_{j=x_1}^{x_m} e\left(H_2(\omega j) y^{t_j} p + p_{pubj}, \gamma_j\right) = e(p, p) \tag{7}$$

4) 登记上链阶段

审核中心 B 验证通过后, 其便将交易数据、贸易主体信息和验证数据、参数等一起经过 ZSS04 短签名方案签名后发送至区块链商业网络 C. 审核中心 B 和交易中心 S 封装已验证通过的交易Tra_i, 将交易Tra_i发送给贸易主体 P 进行核对, 核对无误后用自己的私钥进行签名. 之后, 将交易Tra_i在区块链商业网络 C 中进行广播, 所有节点接收并验证交易, 经过优化的共识算法 (CVBFT) 共识之后将交易Tra_i写入区块链网络中, 如图 5 所示.

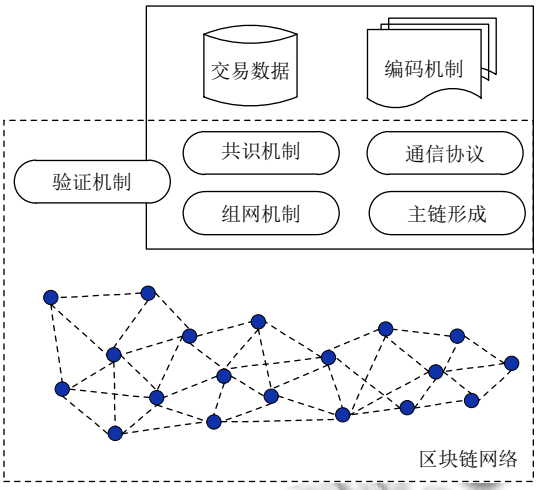

图 5 区块链网络示意图

至此, 在供应链中一笔交易数据上链存储工作完成.

2.3 优化的共识机制

区块链技术最大的优势在于能够在高度分散的去中心化系统中通过激励机制, 使各节点积极参与验证区块数据的真实性共识工作[17]. 但该机制在供应链贸易可信数据管理的应用中存在明显的不足: 主要集中在以工作量证明 (PoW) 为代表的共识算法具有高耗、低效等性能瓶颈, 在比特币交易系统中 TPS 仅有 6.67, 一个区块的产生需要 10 minutes, 并且需要花费高达 1 个小时的时间全网确定 1 次交易, 这些性能问题是难以满足供应链贸易数据管理应用的实际需求. 本文在研究工作量证明 (PoW)、权益证明 (PoS)、股份授权证明 (DPoS) 等共识算法的基础上, 对拜占庭容错算法 (PBFT) 进行优化, 提出了适用于供应链可信数据管理的信用投票机制 (CVBFT): 每个节点通过其主链上的公证单元被引用的次数作为投票的凭据进行投票, 得票最高的 10 个节点被选举为公证节点并提供公证单元; 经过一段时间后, 对数据进行刷新, 再进行新一轮的投票选举. 经过优化的共识算法 (CVBFT) 过程如图 6 所示.

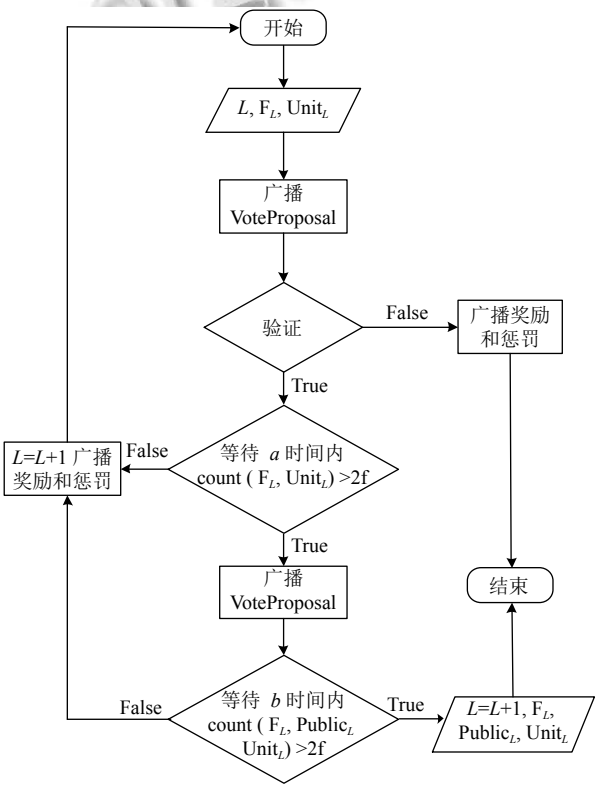

图 6 CVBFT 算法流程图

3 方案可行性分析

3.1 正确性分析

引理 1. 证明式 (7) 的正确性.

证明: 由于 $\gamma_j \leftarrow p(x + H_2(\omega j) y^{t_j})^{-1} \in G_1$, $\gamma = \prod_{j=x_1}^{x_m} \gamma_j^{\Gamma_j}, I \in Z_q^*$, 故有式 (8).

首先, 令 $A=H_2(\omega j) y^{t_j} p + p_{pubj}$; $B=H_2(\omega j) y^{t_j} p + p_{pubk}$; $C=\sum_{j \neq k} I_j \left(H_2(\omega j) y^{t_j} p + p_{pubj}\right)$; 当且仅当$j \neq k$时, 则有:

$$\begin{aligned}&\prod_{j=x_1}^{x_m} e\left(H_2(\omega j)y^{t_j}p+p_{pubj},\gamma_j\right)\\&=\prod_{j\neq k} e\left(A,I_j p\right).e\left(B,(H_2(\omega j)y^{t_j}p+\gamma_k)^{-1}(p-C)\right)\\&=e(C,p).e(p,-C).e(p,p)=e(p,p)\end{aligned} \quad (8)$$

综上所述, 式 (6) 成立, 证毕.

3.2 共识算法验证

3.2.1 功耗

功耗是衡量一个系统对资源所利用的能力, 也是衡量系统对资源消耗的重要指标, 本文使用每轮共识的消耗 (Consume Per Trun, CPT) 来表示. 基于 DAG 的区块链应用中的功耗是指每一轮公证人共识节点选举所需要的时间Δ_L内, 对 CPU 的使用率, 如式 (9). 其中Δ_L为当前 L 轮共识的时间间隔, $ConsensusConsume_{\Delta_L}$ 为该时间间隔内 CPU 的使用率.

$$CPT_{\Delta_L}=ConsensusConsume_{\Delta_L} \quad (9)$$

通过运行共识算法, 取Δ_L为 60 s, 然后测试 CPU 在这段时间内的使用率. 每次选举都测试 10 轮, 取 10 轮的平均值作为各不同选举的公证人共识节点数量的CPT_{Δ_L}. CVBFT 共识算法与 POW 共识算法功耗比较如图 7 所示.

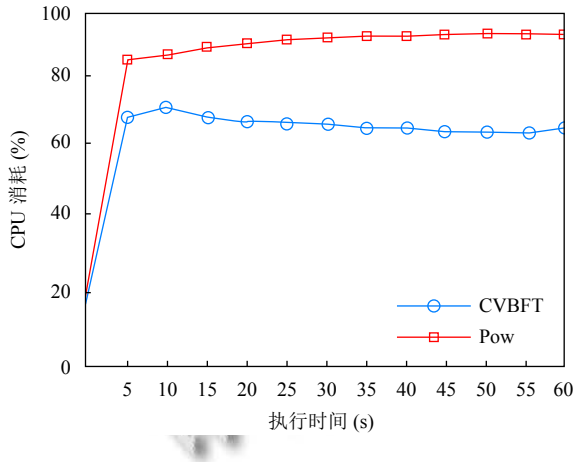

图 7 CVBFT 算法和 PoW 算法的 CPU 消耗比较图

从比较的结果可以看出, 基于 POW 工作量证明的共识算法在运行过程中, CPU 的使用率接近 95%, 完全占据了系统的资源使用. 而 CVBFT 共识算法在在每一轮选取公证人共识节点的过程中 CPU 的使用率仅占 65% 左右. 由此可以看出, 算法的改进起到了降低功耗的作用.

3.2.2 时延

时延的指标可以衡量整个系统所在的网络的通信性能和共识算法完成任务的运行时间. 在基于 DAG 的区块链应用中, 具备高并发是该技术的核心, 故低时延是保证整个系统平稳运行的关键. 本文使用式 (10) 表示时延.

$$Delay_L=\text{VoteProposal}_L+\text{Votecondirm}_L \quad (10)$$

其中, $Delay_L$为当前第 L 轮的公证人共识节点选举产生选举人所用的时间间隔 (时延), VoteProposal_L为共识节点广播投票阶段到其中的共识节点确认投票信息又到全网广播确认阶段的过程, Votecondirm_L为共识节点最终确认选举公证人共识节点的过程.

本文选取的公证人共识节点数量分别为 4, 5, 6, 7, 8, 9. 根据不同选取的公证人共识节点数量来测试时延, 测试共识轮次为 10 轮, 结果如图 8 所示.

实验结果表明, 选取公证人数量的不同并不会对算法的时延造成太大的影响.

3.3 可信性分析

本节将从处理可信性、存储可信性这两方面来对供应链可信数据管理方案进行分析.

性质 1. 该供应链可信数据管理方案满足数据处理可信性.

证明: 首先, 在方案的初始阶段, 贸易主体之间自由定制符合各方利益的交易规则, 并采用智能合约将交易规则计算机代码化. 智能合约既包含执行逻辑, 又包含执行条件, 当条件满足时, 即执行逻辑. 由数据管理的视角来看, 智能合约类似于数据管理系统中的存储过程和触发器. 但又不同于传统数据管理系统中的事务, 不仅智能合约所处理的结果需要保存在区块链里面, 而且其本身亦要存储区块链中. 在交易的确认、验证阶段, 交易中心 S 将交易数据定义为集合$T=\left\{j|t_j,1\leqslant j\leqslant i\right\}$, 贸易主体 P 挑选一个随机的签名密钥对 (rpk,rsk), 并计算公钥$p_{pub}\leftarrow xp$, 并对交易数据取验证标签, 记作集合$\xi=\left\{j|\gamma_j,1\leqslant j\leqslant i\right\}$. 交易中心 S 将验证所需的数据 ($\xi$, rpk, i, p_{pub}) 发送到审核中心 B. 审核中心 B 接收验证数据后, 若外界再对交易数据进行非法篡改则会在证据挑战阶段被检测到, 从而保障交易数据 T 的安全性和完整性. 综上, 所提方案满足可信数据管理的处理可信性.

性质 2. 该供应链可信数据管理方案满足数据存储可信性.

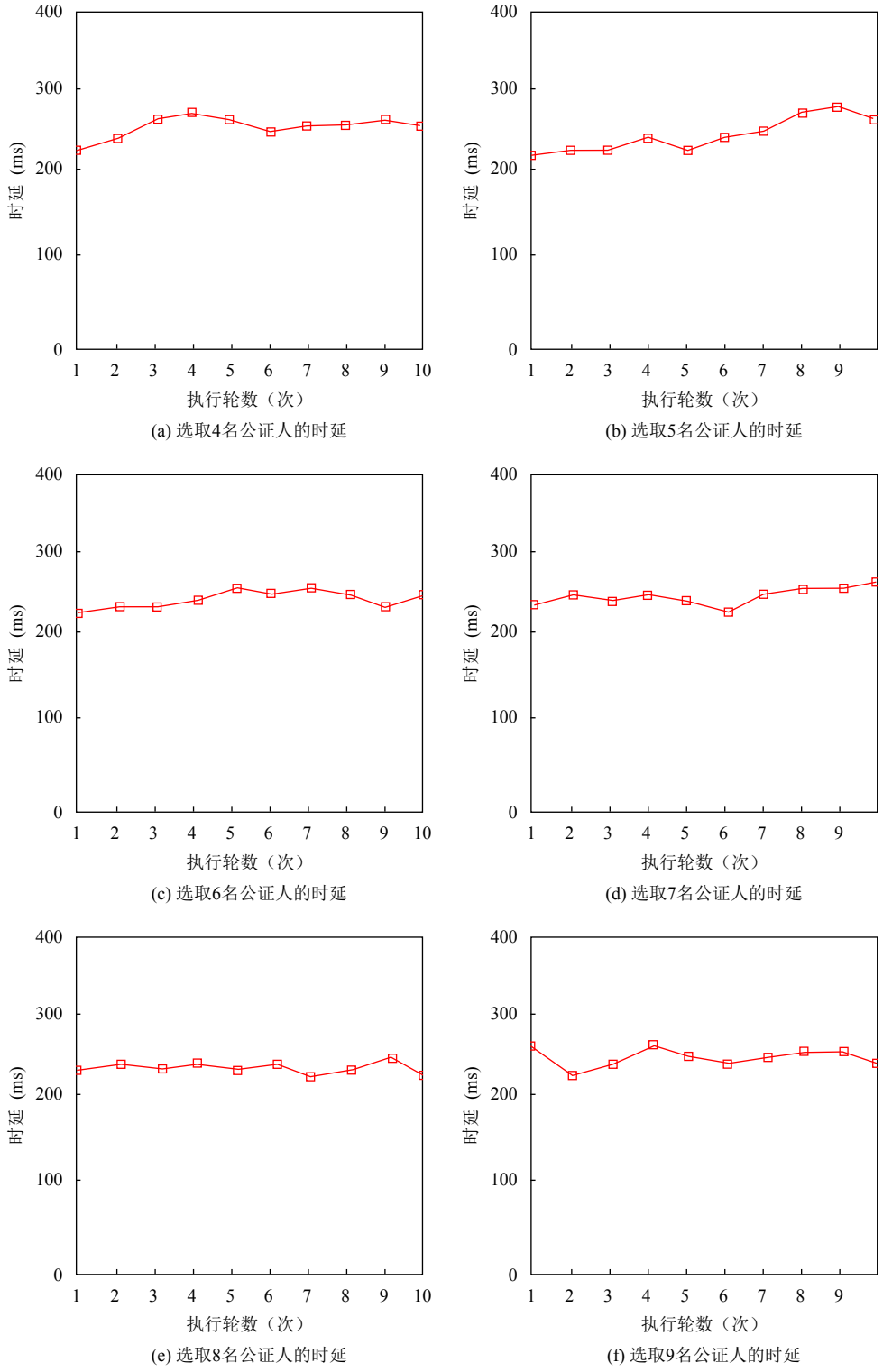

图 8　选取 4 到 9 名公证人的时延对比图

证明: 存储可信性本质是解决分布式共识问题. 在登记上链阶段, 交易Tra_i在区块链商业网络 C 中进行广播, 所有节点接收并验证交易, 经过优化的共识算法(CVBFT) 共识之后将交易Tra_i写入区块链中. 区块链中区块与区块之间通过密码散列函数依次顺序钩稽. 如果攻击者篡改或伪造交易记录, 则需构造一条长于

当前主链的链. 因此, 所需要的计算力远大于合法区块链的计算力, 这样的篡改得不偿失. 所以, 链上的数据安全存储也得到有效保障.

3.4 复杂度分析

以下将对方案中交易智能合约生成、交易审核以及与相关交易数据上链三个阶段进行计算复杂度和通信复杂度分析. 其中, 方案的通信复杂度指标主要是依据事务处理过程中的通信轮数, 该方案复杂度分析结果如表 1 所列.

表 1 方案复杂度分析

	通信复杂度	计算复杂度
智能合约生成	O(1)	O(qt)
交易审核	O(1)	O(t)+O(v)
数据上链	O(1)	O(1)

交易智能合约生成阶段的计算复杂度是由交易中心生成合约的 O(qt), 通信复杂度为 O(1), 其中 q 是一笔交易中的交易数量, 而 t 则是涵盖的交易种类. 交易审核阶段的计算复杂度主要是两部分构成, 一部分是交易中心 O(t), 另一部分是审核中心 O(v), 通信复杂度为 O(1), 其中 v 为按种类进行随机抽取的数据块数. 数据上链阶段的计算复杂度是由贸易主体 P、审核中心 B 和交易中心 S 等一系列的验证签名构成, 为 O(1), 通信复杂度为 O(1).

4 结论与展望

本文通过智能合约的加入, 使贸易中交易双方或多方即可如约履行自身的义务, 实现从外挂合约到内置合约的转变, 有效管控履约风险. 鉴于传统供应链存在交易本身真实性难以验证、信任问题突出等, 通过提出适用于供应链的共识机制, 构建一种以低时延、低成本、低功耗建立的信任机制, 实现了贸易中交易数据的可信存储, 数据可信度得以提升, 并降低行业中的风控成本. 在本文工作基础上, 下一步将基于区块链构建“债转平台”, 以债权凭证为载体, 降低融资成本, 以解决供应商对外支付及上游客户的融资需求.

参考文献

1 Xu NR, Liu JB, Li DX, *et al*. Research on evolutionary mechanism of agile supply chain network via complex network theory. Mathematical Problems in Engineering, 2016, 2016: 4346580. [doi: 10.1155/2016/4346580]

2 Sarr I, Naacke H, Gueye I. Blockchain-based model for social transactions processing. Proceedings of the 4th International Conference on Data Management Technologies and Applications. Colmar, Alsace, France. 2015. 309–315. [doi: 10.5220/0005519503090315]

3 王海龙, 田有亮, 尹鑫. 基于区块链的大数据确权方案. 计算机科学, 2018, 45(2): 15–19, 24. [doi: 10.11896/j.issn.1002-137X.2018.02.003]

4 钱卫宁, 邵奇峰, 朱燕超, 等. 区块链与可信数据管理: 问题与方法. 软件学报, 2018, 29(1): 150–159. [doi: 10.13328/j.cnki.jos.005434]

5 Tan H. Research and design on B2B E-commerce supply chain management system. Applied Mechanics and Materials, 2013, 380-384: 4771–4774. [doi: 10.4028/www.scientific.net/AMM.380-384.4771]

6 Hwang HJ, Seruga J. An intelligent supply chain management system to enhance collaboration in textile industry. International Journal of U- and E-Service, Science and Technology, 2011, 4(4): 47–62.

7 Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. https://nakamotoinstitute.org/bitcoin/, [2008-10-31].

8 Watanabe H, Fujimura S, Nakadaira A, *et al*. Blockchain contract: Securing a blockchain applied to smart contracts. Proceedings of 2016 IEEE International Conference on Consumer Electronics. Las Vegas, NV, USA. 2016. 467–468. [doi: 10.1109/ICCE.2016.7430693]

9 Zhang FG, Safavi-Naini R, Susilo W. An efficient signature scheme from bilinear pairings and its applications. Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography. Singapore. 2004. 277–290.

10 胡德敏, 余星. 一种基于同态标签的动态云存储数据完整性验证方法. 计算机应用研究, 2014, 31(5): 1362–1365, 1395. [doi: 10.3969/j.issn.1001-3695.2014.05.018]

11 Wang C, Chow SSM, Wang Q, *et al*. Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on Computers, 2013, 62(2): 362–375. [doi: 10.1109/TC.2011.245]

12 王苏南, 李印海, 罗兴国. 基于分层抽样算法的异常攻击流量检测. 计算机工程, 2012, 38(12): 105–109. [doi: 10.3969/j.issn.1000-3428.2012.12.031]

13 Nick S. Smart contracts: Building blocks for digital markets. http://www.alamut.com/subj/economics/nick_szabo/smartContracts.html, [1998-08-16].

14 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. [doi: 10.16383/j.aas.2016.c160158]

15 Buterin V. A next generation smart contract &decentralized application platform. http://fermatslibrary.com/s/ethereum-a-next-generation-smart-contract-and-decentralized-application-platform, 2013.

16 鲁静, 宋斌, 向万红, 等. 基于区块链的电力市场交易结算智能合约. 计算机系统应用, 2017, 26(12): 43–50. [doi: 10.15888/j.cnki.csa.006109]

17 Gramoli V. From blockchain consensus back to Byzantine consensus. Future Generation Computer Systems, 2017. [doi: 10.1016/j.future.2017.09.023]