密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research, 2018, 5(5): 501–509

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

区块链技术专刊

可更改区块链技术研究*

李佩丽[1,2], 徐海霞[1,2,3], 马添军[1,2,3], 穆永恒[1,2,3]

1. 中国科学院 信息工程研究所, 北京 100093
2. 中国科学院 数据与通信保护研究教育中心, 北京 100093
3. 中国科学院大学 网络空间安全学院, 北京 100049

通信作者: 徐海霞, E-mail: xuhaixia@iie.ac.cn

摘 要: 近年来, 区块链技术受到学术界和产业界的广泛关注和研究. 区块链具有透明性、去信任、可追溯、不可更改等特点, 吸引了不少企业开发基于区块链的应用. 区块链不可更改是指区块链上的历史数据一旦确认就不能被更改, 这一特点保证了区块链上历史数据的可靠性和完整性. 然而区块链的不可更改并非绝对, 在一些情况下, 如区块链应用平台存在程序漏洞、某一历史记录存在错误但没被及时发现等, 就有必要对出问题的历史记录做出响应和更改. 针对区块链可更改方面的研究工作较少, 埃森哲公司申请了可编辑区块链专利. 其主要用到变色龙哈希函数这一工具, 哈希函数的陷门由一个用户或多个用户共同掌管, 从而将修改区块的权限交给一方或多方. 因此, 他们的方案需预先选定一个更改者, 或由多个更改者进行交互完成更改. 本文针对联盟链, 设计了新的变色龙哈希函数, 使得在满足修改触发条件的情况下, 联盟链中的每个用户都有修改历史记录的权利. 我们提出了多方共同决策的区块链更改方法, 区块链的更改不依赖于一方, 也不需要多方交互完成更改, 只需要随机选出一个用户即可完成更改, 因此整个过程的交互次数较少.

关键词: 区块链; 可更改; 哈希函数; 秘密分享; 共识机制
中图分类号: TP309.7 **文献标识码:** A DOI: 10.13868/j.cnki.jcr.000259

中文引用格式: 李佩丽, 徐海霞, 马添军, 穆永恒. 可更改区块链技术研究[J]. 密码学报, 2018, 5(5): 501–509.
英文引用格式: LI P L, XU H X, MA T J, MU Y H. Research on fault-correcting blockchain technology[J]. Journal of Cryptologic Research, 2018, 5(5): 501–509.

Research on Fault-correcting Blockchain Technology

LI Pei-Li[1,2], XU Hai-Xia[1,2,3], MA Tian-Jun[1,2,3], MU Yong-Heng[1,2,3]

1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
2. Data Assurance & Communications Security Center, Chinese Academy of Sciences, Beijing 100093, China
3. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: XU Hai-Xia, E-mail: xuhaixia@iie.ac.cn

Abstract: In recent years, blockchain technology has received extensive attention from academia and industry. Blockchain has the characteristics of transparency, de-trusting, traceability, and unchangeability, attracting many enterprises to develop blockchain-based applications. The unchangeability of

* 基金项目: 国家重点研发计划 (2017YFB0802500)
Foundation: National Key Research and Development Program of China (2017YFB0802500)
收稿日期: 2018-07-16 定稿日期: 2018-09-19

blockchain means that the historical data on the blockchain cannot be changed once it is confirmed. This feature guarantees the reliability and integrity of the historical data on the blockchain. However, the unchangeablility of blockchain is not absolute. In some cases, such as a program loophole in the blockchain application platform, or a historic error not found in time, it is necessary to respond to and change the history of the problem. There is little research work on the blockchain changeability. Accenture applied for a patent about editable blockchain. It mainly uses the Chameleon Hash function. The trapdoor of the Hash function is managed by one node or multiple nodes, thus the edit authority is given to one or more parties. Therefore, the scheme needs to select a changer in advance, or interact with multiple changers to complete the change. This paper designs a new Chameleon Hash function for the alliance chain, so that each node in the alliance chain has the right to modify the history. We propose a blockchain error correction method for multi-party decision making. The blockchain modification does not depend on one party, and does not require multiple parties to complete the change. It only needs to randomly select a node to complete the change, so only a small number of interactions in the whole process is needed.

Key words: blockchain; changeable; Hash function; secret sharing; consensus mechanism

1 引言

区块链被认为是近年来最具革命性的新兴技术之一, 其首次出现在中本聪的论文《比特币: 一种点对点的电子现金系统》 [1] 中, 是比特币系统的底层支撑技术. 随后有很多专家和学者对区块链技术本身及其应用展开了深入的研究. 对于区块链没有统一的定义, 一般认为区块链是一种分布式的公开数据库, 具有防伪、不可更改、交易可追溯、去信任化等特点 [2]. 区块链中智能合约技术提供了一个公正、可自动执行的技术平台. 区块链的诸多优点吸引了不少企业开发基于区块链的应用 [3,4].

区块链不可更改的特点保证了历史交易数据的完整性和不可更改性, 是很多实际应用需满足的重要特性. 然而区块链上历史数据的不可更改并非绝对, 以太坊 [5] 所爆出的安全漏洞为此敲响了警钟. 以太坊区块链众筹项目 TheDAO [6] 在 2016 年 6 月份遭受过一场灾难性漏洞. TheDAO 合约的源码中存在着一个函数调用的漏洞, 使得攻击者可以将其资产池中的以太币非法转移给自己. 2016 年 7 月, 以太坊官方修改了以太坊的源码, 强行把 TheDAO 及其子 DAO 的资金转移到另外一个合约地址, 通过这种方式夺回被攻击者控制的 DAO 合约中的币, 但是这样却导致以太坊发生了分叉, 从而导致变成了两条链: 一条是原始的区块链, 一条是分叉出来的新的链. 近期以太坊又爆高危漏洞, 2018 年 5 月 ATN 技术人员收到异常监控报告, 显示 ATN Token 供应量出现异常, Token 合约由于存在漏洞受到攻击. 攻击者伪造合约账户进行恶意操作, 窃取了 ATN 的所有权, 并窃取合约地址持有的代币. ATN 基金会透露, 将销毁 1100 万个 ATN, 并恢复 ATN 总量, 同时将在主链上线映射时对黑客地址内的资产予以剔除, 确保原固定总量不变. TheDAO 事件和 ATN 事件给我们敲响了警钟. 区块链缺乏治理规则, 当遇到突发事件时, 没有调整规则, 只能通过软分叉或硬分叉来解决问题. 这容易导致意见分歧和混乱, 因此有必要建立区块链上的治理规则, 提高区块链的风险抵抗能力, 避免系统分裂.

区块链智能合约支持多样化的实际应用, 在大多情况下区块链上的数据应该满足不可更改性, 然而在一些特殊情况下 (突发意外、记录出错等), 我们也应该考虑区块链历史数据的可更改. 绝对的不可更改不利于区块链智能合约的错误纠察和及时止损. 对区块链更改机制的研究有助于区块链应用的健康发展.

区块链主要分为三类 [7]: 公有链、联盟链和私有链. 公有链对所有人开放, 任何人都可以参与; 联盟链只对特定的组织团体开放; 私有链对单独的个人或实体开放. 相对公有链, 联盟链中用户数量有限, 共识效率较高, 适合用于企业、组织之间. 目前有不少企业级的区块链项目被提出, 越来越多的公司根据自己的业务需求搭建联盟链平台. 本文针对联盟链, 研究其可更改技术, 使得当错误和突发情况出现时, 平台可以按照既定规则修改相应历史数据.

目前针对可更改的区块链研究工作较少. 埃森哲公司申请了可编辑区块链专利, 采用变色龙哈希 (Chamelelon Hash) 技术来更改一个链中的历史区块 [8]. 在他们的方案中, 只有拥有密钥的用户才可以使

用变色龙哈希来修改历史数据, 这个过程操作简单、效率较高, 不需要对后续无关区块进行改动. 变色龙哈希技术用于实现可编辑的区块链是一项重要的创新, 不过其存在中心化的问题. 因为变色龙哈希的陷门若由一个用户掌管, 就可能存在其任意修改区块历史的行为. 另外一个方法是将变色龙哈希的陷门分享给多个用户, 由多个用户共同协作来完成区块历史的编辑, 交互次数较多. 针对这个问题, 本文根据联盟链的特点, 研究采用多方共同决策的方法实现可更改的区块链. 为了避免更改功能被滥用, 或者引发链上数据混乱, 本文提出明确的触发准则以及修改原则. 一旦用户有更改需求, 首先该用户将更改需求发送给其他用户, 联盟链中每个用户都对于是否更改进行投票. 如果超过一半的用户投票同意更改某一历史区块, 则触发更改功能, 根据特定的共识机制随机选择一个用户对历史区块进行更改. 联盟链中每个用户地位平等, 都有参与投票的权利和被选为更改者的机会.

本文结构如下: 第 2 节介绍区块链相关知识和本文所用工具的相关概念; 第 3 节给出我们针对联盟链的可更改区块链的技术思路; 第 4 节介绍可更改区块链的具体构造; 第 5 节是对全文的一个总结.

2 背景知识

2.1 区块链

本文引用《区块链技术指南》[7] 一书中对区块链的定义. 区块链是基于区块链技术形成的分布式公共数据库 (或称公开账本). 其中区块链技术是指多个参与方之间基于现代密码学、分布式一致性协议、点对点网络通信技术和智能合约编程语言等形成的数据交换、处理和存储的技术组合.

为了方便的理解区块链技术, 我们先从介绍比特币入手. 在比特币出现之前, 数字货币系统需要可信的第三方机构来保证交易的安全有效, 例如银行、支付宝、微信等, 记账权则交给这些可信中心. 比特币是首个去中心化的数字货币, 可以解决双重支付和共识问题. 比特币系统不依赖于可信的中心管理员, 系统中的用户地位平等. 大家共同维护账本、验证交易, 并且竞争提出区块 (记账)[9].

比特币系统的分布式记账就是通过区块链技术来实现的. 在比特币系统中交易被存储在数据区块当中, 大约每 10 分钟就会产生一个区块. 每个区块包含区块头和区块体两部分. 其中交易以 Merkle tree 的形式存储在区块体部分, 区块头包含当前版本号、前一个区块的地址、时间戳、随机数和当前区块的哈希值以及交易 Merkle tree 的根. 区块是通过挖矿产生的. 而挖矿的过程是穷举随机数的过程. 矿工 (比特币用户) 将 10 分钟内的交易打包加上前一个区块的哈希值, 算出一个随机数, 使得这些值的哈希值满足某个条件. 如果矿工算出了满足条件的随机数则获得了这个区块的记账权, 随后矿工需要将其广播交给其他用户验证. 挖矿的矿工竞争获得最终记账权, 矿工需要付出大量的能源和时间, 以更大的概率获得一个区块的记账权. 通过这样的记账方式, 大家共同验证、维护统一的账本, 已经记录在区块链中的数据无法篡改.

比特币中区块链的简易结构如图1.

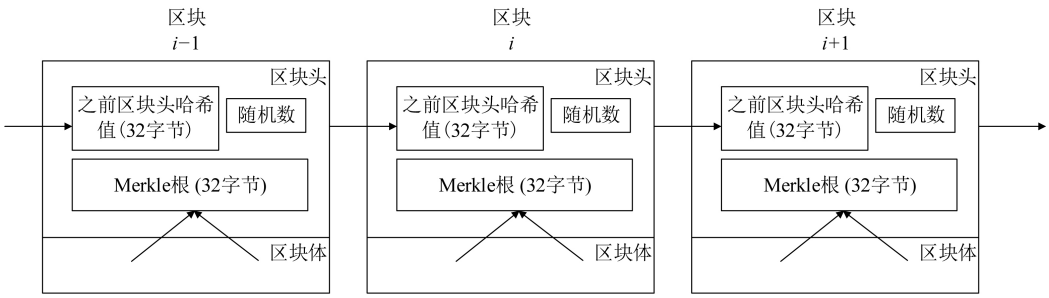

图 1 区块链结构图

Figure 1 Blockchain structure

比特币区块链具有透明性、公开验证、不可更改等特点. 其中不可更改性主要是由哈希函数的特性来保证的. 区块链中每个区块包含前一个区块的哈希值. 如果一个历史区块中数据被篡改, 根据哈希函数的抗碰撞性, 其哈希值也会相应变化, 这样就可以被用户发现.

本文对可更改区块链的研究同样从哈希函数入手, 为哈希函数设置陷门, 拥有陷门的用户可以修改区块中的内容但不改变整个区块的哈希值, 即容易找到碰撞. 这样后续无关区块不用发生任何改变. 与埃森哲利用变色龙哈希设计可编辑的区块链想法类似, 不同的是我们对变色龙哈希函数进行变化, 赋予多个用户修改历史记录的权限, 而且我们的方案中更改者的选择是随机的, 并不会预先确定.

2.2 工具

下面我们分别对本文用到的两个主要工具——变色龙哈希和可验证秘密分享进行介绍.

2.2.1 变色龙哈希 (Chamelelon Hash)

哈希函数 [10] 简单来讲就是能将任意长度的输入转换成一个固定长度的输出, 这个固定长度的输出称为原消息的散列值或哈希值. 通过原始输入消息可以很容易地计算出其哈希值, 通过输出 (哈希值) 很难还原出原始输入. 理想的哈希函数针对不同的输入产生不同的输出. 如果两个不同的消息产生了相同的哈希值, 则称发生了碰撞.

哈希函数满足如下特性:

(1) 抗碰撞性: 理想的哈希函数是无碰撞的, 但是实际的算法设计中很难做到. 对于哈希函数的抗碰撞性有弱抗碰撞性和强抗碰撞性两种.

(a) 弱抗碰撞性: 对于给定的一个消息, 要发现另一个消息使其碰撞在计算上是不可行的. 具体为: 对于任意的输入 m, 得到输出的结果 Hash(m), 很难找到另一个输入 m'(m' 不等于 m), 使得 m' 的 Hash 结果也为同样的输出, 即 Hash(m) = Hash(m').

(b) 强抗碰撞性: 对于任意的一对不同的消息, 使其碰撞在计算上是不可行的. 很难找到任意两个不同的消息 m 和 m', 使得 Hash(m) = Hash(m').

(2) 高灵敏度: 对于一个输入数据块, 哪怕只改动其一个比特位, 其哈希值的改动也会非常大.

与传统哈希函数不同, 变色龙哈希函数 (Chamelelon Hash) 可以人为设下一个陷门, 掌握了陷门就能轻松找到碰撞 [11]. 对于没有陷门的用户而言, 变色龙哈希函数依然满足抗碰撞性.

定义 1 一个变色龙哈希由四个算法 cham_hash = (Setup, KeyGen, Hash, Forge) 组成.

- Setup(λ): 输入安全性参数 λ, 输出公共参数 pp;
- KeyGen(pp): 输入公共参数 pp, 输出公私钥对 (HK,CK), HK 为公钥, CK 为私钥又称为陷门;
- Hash(HK, m, r): 输入公钥 HK, 消息 m 和随机数 r, 输出变色龙哈希值 CH;
- Forge(CK, m, r, m'): 输入私钥 CK, 消息 m, 随机数 r, 消息 m', 输出另一个随机数 r', 满足 CH = Hash(HK, m, r) = Hash(HK, m', r').

变色龙哈希满足的安全性要求

- 抗碰撞 (collision resistance): 不存在一个有效算法在输入公钥 HK, 可以找到 (m_1, r_1) 和 (m_2, r_2), 其中 $m_1 \neq m_2$, 满足 Hash(HK, m_1, r_1) = Hash(HK, m_2, r_2).
- 陷门碰撞 (trapdoor collisions): 存在有效算法, 在输入陷门 CK 后, 对于任意的 m_1, r_1, 给定 m_2, 可以计算出 r_2, 满足 Hash(HK, m_1, r_1) = Hash(HK, m_2, r_2).
- 语义安全 (semantic security): 对于任意消息 m_1, m_2, Hash(HK, m_1, r_1) 与 Hash(HK, m_2, r_2) 的概率分布是不可区分的, 特别地, 当 r 为随机选择时, 从 Hash(HK, m, r) 无法得到关于 m 的任何信息.

变色龙哈希函数的实例

Hugo Krawczyk 和 Tal Rabin 在 2000 年提出了变色龙哈希方案 [11], 方案的具体描述如下:

- Setup(λ): 输入安全性参数 λ, 构造满足安全参数 λ 的大素数 p, q, 其中 p, q 满足 $p = kq + 1$, 选取乘法循环群 Z_p^* 中阶为 q 的元素 g, 输出公共参数 pp = (p, q, g);
- KeyGen(pp): 输入公共参数 pp, 在乘法循环群 Z_q^* 中随机选择指数 x, 计算 $h = g^x$. 最后得到私钥 CK = x, 公钥 HK = h;

- Hash(HK, m, r): 输入公钥 HK $= h$, 消息 m, 随机数 r, 其中 m, r 均为 Z_q^* 中的元素, 输出变色龙哈希值 CH $= g^m h^r \bmod p$;
- Forge(CK, m, r, m'): 输入私钥 CK $= x$, 消息 m, 随机数 r, 消息 m', 其中 m, r, m' 均为 Z_q^* 中的元素, 根据 CH $= g^m h^r = g^{m'} h^{r'} \bmod p$, 可得 $m + xr = m' + xr' \bmod q$, 继而可计算出 $r' = (m - m' + xr) \cdot x^{-1} \bmod q$.

2.2.2 可验证秘密分享

Shamir 和 Blakley 于 1979 年分别独立地提出秘密分享的概念, 并给出了 (k, n) 门限秘密分享方案 [12,13]. 在秘密分享方案中, 用户将需共享的秘密分成若干秘密份额也称子密钥、碎片, 并安全地分发给若干参与者掌管, 同时规定哪些参与者合作可以恢复该秘密. 之后 Chor 等人在 1985 年提出了可验证秘密分享的概念 [14]. 可验证秘密分享考虑到基本秘密共享方案中秘密分发者与参与者可能是不诚实的, 它在基本秘密共享方案的基础上, 增加一些公开承诺和验证算法, 来检测试图伪造秘密份额的用户, 包括秘密分发者和参与者 [15–17].

一个 (t, n) 可验证秘密分享方案需要满足两个要求:

(1) 可验证性: 在收到一份秘密份额后, 用户能够测试它是否是一个有效的份额. 如果一个份额有效, 则存在一个唯一的秘密作为秘密重构算法的输出, 秘密重构算法作用在任意 t 个有效份额上.

(2) 不可预测性: 对于多项式时间算法, 输入 $t-1$ 个秘密份额, 不能获得任何有关秘密的信息.

可验证秘密分享的实例

可验证秘密分享的一个典型例子是 Feldman 在 1987 年提出的一个非交互的方案 [15]. 它通过增加一个公开验证函数来扩展 Shamir 的方案, 是第一个不需要可信机构参与的非交互式可验证秘密分享方案. 该方案由四部分组成: 系统参数、秘密分发、验证算法和秘密重构.

(1) 系统参数: p 是一个大素数, q 为 $p-1$ 的一个大素因子, g 为 q 阶元, 三元组 (p, q, g) 是公开的, t 是门限值, n 是参与者数目, s 为要共享的秘密, 秘密空间与份额空间均为有限域 GF(p).

(2) 秘密分发: 随机选择一个 GF(p) 上的 $t-1$ 次多项式 $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$, 满足 $f(0) = a_0 = s$, 然后计算各秘密份额 $s_j = f(x_j) (\bmod q)$ 并秘密地发送给参与者, 其中 $j = (1, 2, \cdots, n)$, 同时公开函数 f 的系数的承诺 $c_i = g^{a_i} \bmod p$, 其中 $i = (0, 1, 2, \cdots, t-1)$.

(3) 验证算法: 各参与者在收到秘密份额后, 验证 $g^{s_j} = \Pi_{i=0}^{t-1} c_i^{x_j^i} \bmod p$, $(j = 1, 2, \cdots, n)$ 是否成立, 若成立则份额有效, 否则说明收到的秘密份额不正确.

(4) 秘密重构: 当 t 个参与者 $P_1, P_2, \cdots, P_t$ 合作恢复秘密时, 每一个参与者 P_j, 公开他的份额 s_j 给其他合作者, 每个合作者通过执行验证算法判断秘密份额的有效性. 由拉格朗日插值公式算得多项式函数 $f(x)$, 最后计算函数值 $f(0)$ 即为秘密 s.

3 可更改区块链的技术思路

前面介绍的变色龙哈希函数可以直接用于实现可编辑的区块链. 不过只有拥有陷门的那个用户才可以修改历史区块. 如果陷门拥有者恶意, 其可以任意修改历史区块信息或拒绝修改本应修正的区块. 因此方案的可靠性依赖于这个用户需是可信的, 这在一定程度上削弱了区块链分布式去中心化的思想.

本文针对联盟链的特点, 旨在设计去中心化的可编辑区块链, 使得所有参与者投票决定是否修改某一历史区块. 一个比较直接的想法是, 将变色龙哈希函数中的陷门进行秘密分享, 大于一半的用户可以恢复秘密. 如果有一半以上的用户同意修改历史区块, 例如将区块消息 m 改为 m', 则这些用户执行安全多方计算协议算出 r 对应的 r', 使得 Hash(m, r) = Hash(m', r'). 这个过程需要用户之间较多的交互, 增加了通信复杂度 [18].

为了避免使用安全多方计算这个比较强的密码工具, 让联盟链中所有用户都有编辑区块链的权力, 我们设计了新的变色龙哈希函数, 使得每个用户都有相应的陷门. 具体来说, 变色龙哈希函数的形式为 Hash$(m, \boldsymbol{r}) = g^m h_1^{r_1} h_2^{r_2} \cdots h_n^{r_n}$, 其中区块信息为 m, 随机数 $\boldsymbol{r} = (r_1, r_2, \cdots, r_n)$, $h_1 = g^{x_1}$, $h_2 = g^{x_2}$,

$\cdots, h_n = g^{x_n}$. 其中 $(x_1, x_2, \cdots, x_n)$ 为哈希函数的 n 个陷门, 分别被联盟链中的用户 $P_1, P_2, \cdots, P_n$ 拥有. 如果一个用户 P_i 要修改某一历史区块信息, 如将 m 改为 m', 则利用其陷门 x_i 就可以求得 $r_i^{'}$, 使得 $\mathrm{Hash}(m, (r_1, \cdots, r_i, \cdots, r_n)) = \mathrm{Hash}(m', (r_1, \cdots, r_i^{'}, \cdots, r_n))$. 这个哈希函数赋予了每个联盟链用户修改区块信息的能力.

那么到底由谁来修改区块? 这是我们设计方案的要解决的一个关键问题. 如果修改区块的用户事先已经确定, 那么就存在被攻击的风险. 因此修改区块的用户需是不可预测的、有一定的随机性. 这里我们采用分布式的随机数生成协议 (distributed random generation (DRG) protocol), 使多方共同产生一个随机数, 由这个随机数决定哪个用户来修改区块. 具体思路为: 假设有 $t(t > \frac{n}{2})$ 个用户同意修改, 每个参与修改的用户 P_i 产生一个随机数 ρ_i, 通过可验证的秘密分享 (verifiable secret sharing, VSS) 分享给其他用户. 每个用户验证所收到的分享值, 将分享值相加后广播, 最后由拉格朗日插值公式 (只需要够 t 个值即可) 算得 ρ 的值 (其中 $\rho = \rho_1 + \rho_2 + \cdots + \rho_t$). 这样每个用户都可以得到随机数 ρ 的值. 每个用户所选的随机数 ρ_i 对其他用户都是保密的. 对于 $i \in [t]$, 每个同意修改的用户计算哈希函数值 $\mathrm{Hash}(\mathrm{PK}_i, \rho)$. 我们规定这些哈希值中最小的哈希值所对应的公钥记为 PK_c 即为选中更改区块的那个用户. 被选中的用户 P_c 用自己拥有的陷门 x_c 计算修改后的区块 $(m', (r_1, \cdots, r_c', \cdots, r_n))$. 更改者对修改后的区块签名, 加上 t 个用户的投票 (表示同意将消息 m 改为 m'), 进行广播. 其他用户验证更改者是否是被选中的那个用户, 并验证投票数目和更改后的区块信息. 若验证通过则保存最新区块信息. 我们的方案通过随机选择更改区块的用户, 使得每个参与者都有被选择的机会, 且攻击者不能预先知道谁被选中, 从而提升了方案的安全性.

因为每个联盟链用户 P_i 都有哈希函数的陷门, 都可以计算 m 对应的新的随机数 $\boldsymbol{r}'$ 使得 $\mathrm{Hash}(m, \boldsymbol{r}) = \mathrm{Hash}(m', \boldsymbol{r}')$, 即都可以改变历史区块信息, 但是用户所发布修改后的区块需满足下面两个条件才能被接受:

(1) 系统中有超过一半的用户 (t 个) 同意修改, 即包含一半以上用户的投票;

(2) 用户的确是被选定修改的那个用户, 这可以通过计算差值公式得到.

4 可更改区块链方案构造

我们针对联盟链 (用户数目已知, 假设为 n 个用户) 设计了区块链的可更改方法, 使得联盟链用户共同投票决定是否更改某一历史区块.

4.1 区块链和哈希函数

本文我们不考虑联盟链的本身共识机制如何设计, 只关注区块链的区块内容和历史区块修改规则. 联盟链中的区块链结构和第二章提到的比特币区块链结构类似, 每个区块包含区块体和区块头两部分. 区块头包含 Merkle 树的根、前一个区块头的哈希值和一个随机数. 区块与区块之间前后通过哈希值相连接. 区块头随机数以外的所有内容记为消息 m, 随机数记为 $\boldsymbol{r}$. 其中 $\boldsymbol{r}$ 表示为 $(r_1, r_2, \cdots, r_n)$. 对 $m, \boldsymbol{r}$ 做哈希, 计算 $\mathrm{Hash}(m, \boldsymbol{r}) = g^m h_1^{r_1} h_2^{r_2} \cdots h_n^{r_n}$, 其中 $h_1 = g^{x_1}$, $h_2 = g^{x_2}$, $\cdots$, $h_n = g^{x_n}$. 陷门 $(x_1, x_2, \cdots, x_n)$ 分别被联盟链中的用户 $P_1, P_2, \cdots, P_n$ 掌管. 假设联盟链的用户数量为 n, 用户 $P_1, P_2, \cdots, P_n$ 的公钥分别是 $(\mathrm{PK}_1, \mathrm{PK}_2, \cdots, \mathrm{PK}_n)$.

我们的变色龙哈希函数的具体构造为:

- $\mathrm{Setup}(\lambda)$: 输入安全性参数 λ, 构造满足安全参数 λ 的大素数 p, q, 其中 p, q 满足 $p = kq + 1$, 选取乘法循环群 Z_p^* 中阶为 q 的元素 g, 输出公共参数 $\mathrm{pp} = (p, q, g)$;
- $\mathrm{KeyGen}(\mathrm{pp})$: 输入公共参数 pp, 随机选择指数 $x_1 \in Z_q^*$, $x_2 \in Z_q^*$, $\cdots$, $x_n \in Z_q^*$, 计算 $h_1 = g^{x_1}$, $h_2 = g^{x_2}$, $\cdots$, $h_n = g^{x_n}$. 私钥 $\mathrm{CK} = (x_1, x_2, \cdots, x_n)$, 公钥 $\mathrm{HK} = (g, h_1, h_2, \cdots, h_n)$;
- $\mathrm{Hash}(\mathrm{HK}, m, \boldsymbol{r})$: 输入公钥 $\mathrm{HK} = g, h_1, h_2, \cdots, h_n)$, 消息 m, 随机数 $\boldsymbol{r} = (r_1, r_2, \cdots, r_n)$, 输出变色龙哈希值 $\mathrm{CH} = g^m h_1^{r_1} h_2^{r_2} \cdots h_n^{r_n} \bmod p$;
- $\mathrm{Forge}(\mathrm{CK}_i, m, \boldsymbol{r}, m')$: 输入私钥 $\mathrm{CK}_i = x_i$, 消息 m, 随机数 $\boldsymbol{r} = (r_1, r_2, \cdots, r_n)$, 消息 m', 根据 $\mathrm{CH} = g^m h_1^{r_1} h_2^{r_2} \cdots h_n^{r_n} = g^{m'} h_1^{r_1} \cdots h_i^{r_i^{'}} \cdots h_n^{r_n} \bmod p$, 可得 $m + x_1 r_1 + \cdots + x_i r_i + \cdots + x_n r_n =$

$m' + x_1 r_1 + \cdots + x_i r_i^{'} + \cdots + x_n r_n \bmod q$, 可计算 $r_i^{'} = (m - m' + x_i r_i) \cdot x_i^{-1} \bmod q$.

我们设计的变色龙哈希函数, 对于没有陷门 $\mathrm{CK} = (x_1, x_2, \cdots, x_n)$ 的用户而言仍然满足抗碰撞性. 对于拥有陷门 x_i 的用户 P_i 可以运行 Forge 算法找到碰撞使得 $\mathrm{Hash}(\mathrm{HK}, m, \boldsymbol{r}) = \mathrm{Hash}(\mathrm{HK}, m', \mathbf{r}')$.

4.2 可更改区块链构造

当出现突发情况需要对历史某一区块信息进行更改时, 方案包含 3 个阶段: 投票阶段、随机选择阶段和更改确认阶段, 下面描述其具体流程.

(1) **投票阶段**

(a) 一段时间后, 区块链中若有用户 $P_s (s \in [n])$ 发起更改请求 R_s, 请求将某一历史区块链中的内容 m 改为 m', 用户对请求签名得到 σ_s, 广播 (R_s, σ_s) 给联盟链中的用户, 开启投票阶段.

(b) 联盟链中其他用户收到请求后, 若同意修改, 则用户对 P_s 的请求 R_s 签名并广播.

(c) 用户 P_s 收集到大于一半的用户 (假设用户数目为 $t > \frac{n}{2}$) 的签名后, 广播这 t 个签名.

(2) **随机选择更改区块的用户**

(a) 参与投票的 t 个用户记为 $(P_1, P_2, \cdots, P_t)$, 每个用户 P_i 选择一个随机数 ρ_i, 做 $\mathrm{Feldman}(t, n)$ 可验证的秘密分享 (VSS) 分享给其他用户. ρ_i 的分享值记为 $(s_{i,1}, s_{i,2}, \cdots, s_{i,n})$.

(b) 每个用户 P_i 验证所收到的分享值, 验证通过后将收到的分享值 $s_{1,i}, s_{2,i}, \cdots, s_{t,i}$ 相加得到 $S_i = s_{1,i} + s_{2,i} + \cdots + s_{t,i}$, 将 S_i 广播.

(c) 每个用户收到至少 t 个 S_i 后, 由拉格朗日插值算得随机数 ρ 的值, $\rho = \rho_1 + \rho_2 + \cdots + \rho_t$.

(d) 每个用户计算哈希值 $h_i = \mathrm{Hash}(\rho, \mathrm{PK}_i)$, 对于所有的 $i \in \{1, 2, \cdots, t\}$. 然后对这 t 个哈希值进行排序, 其中最小的哈希值, 记为 h_c, 所对应的公钥 PK_c 即为当前选中修改历史区块的那个用户.

(3) **更改并确认阶段**

(a) 要将某一历史区块消息 m 变更为消息 m', 被选中的用户 PK_c, 计算 $r_c' = (m - m' + x_c r_c) \cdot x_c^{-1} \bmod q$. 修改后的区块头内容为 $(m', (r_1, \cdots, r_c', \cdots, r_n))$, 除了 m 和 r_c 变为 m' 和 r_c', 其他随机数都没有变.

(b) 用户 PK_c 广播 $(m', (r_1, \cdots, r_c', \cdots, r_n))$、其他用户的投票 (即在投票阶段用户对请求 R_s 的签名)、秘密分享阶段算得的随机数 ρ 和对前面这些内容的签名.

(c) 其他用户验证用户 PK_c 是否对应的哈希值 $\mathrm{Hash}(\rho, \mathrm{PK}_c)$ 最小、PK_c 的签名和用户的投票. 若验证通过, 则验证 $\mathrm{Hash}(m', (r_1, \cdots, r_c', \cdots, r_n))$ 是否等于 $\mathrm{Hash}(m, (r_1, \cdots, r_c, \cdots, r_n))$, 验证通过则记录更改后的历史区块并标记, 标记的内容包含上一步用户 PK_c 广播的所有信息.

4.3 说明

我们方案设置投票环节, 赋予联盟链中所有用户参与投票的权利. 我们假设联盟链中超过一半的用户是诚实的, 如果大于一半的用户同意修改则可以对历史数据进行修改并达成共识. 我们的创新点在于给每个用户都赋予更改的权利, 这和比特币系统每个人都有成为矿工的权利类似. 最后到底由谁来修改不是预先确定的, 通过引入随机性来决定由谁来更改区块, 这样攻击者就不能提前针对某一用户采取攻击行为.

5 总结

区块链以其透明性、去信任化、可追溯、不可更改等特点吸引了学术界和产业界的广泛关注和研究, 已有不少企业着手开发基于区块链的应用 [19–21]. 然而区块链并非完美, 基于区块链设计的应用平台可能因为代码漏洞、人为操作失误等原因导致历史记录存在问题, 影响区块链平台的正常使用和用户的切身利益. 针对这一问题, 本文研究区块链的可更改机制, 设计了针对联盟链的多方共同决策的区块链更改方法,

使得区块链在特殊情况下可更改. 在我们的方案中, 联盟链中每个用户都有投票决策的权利, 并且都有机会被选为更改者. 对可更改区块链的研究有助于促进区块链的广泛应用和健康发展.

References

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

[2] YUAN Y, WANG F Y. Blockchain: The state of the art and future trends[J]. Acta Automatica Sinica, 2016, 42(4): 481–494. [DOI: 10.16383/j.aas.2016.c160158]
袁勇, 王飞跃. 区块链技术发展现状与展望 [J]. 自动化学报, 2016, 42(4): 481–494. [DOI: 10.16383/j.aas.2016.c160158]

[3] XIE H, WANG J. Study on block chain technology and its applications[J]. Netinfo Security, 2016, 2016(9): 192–195. [DOI: 10.3969/j.issn.1671-1122.2016.09.038]
谢辉, 王健. 区块链技术及其应用研究 [J]. 信息网络安全, 2016, 2016(9): 192–195. [DOI: 10.3969/j.issn.1671-1122.2016.09.038]

[4] SWAN M. Blockchain: Blueprint for a New Economy[M]. Sebastopol, CA, USA. O'Reilly. 2015: 22–24.

[5] https://ethereum.org/

[6] BUTERIN V. Critical update re: DAO vulnerability[EB/OL]. https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/. June 17, 2016.

[7] ZOU J, ZHANG H N, TANG Q, et al. Guidelines for Blockchain Technology[M]. Beijing: China Machine Press. 2016: 97–99.
邹均, 张海宁, 唐屹, 等. 区块链技术指南 [M]. 北京, 机械工业出版社. 2016: 97−99.

[8] KRAWCZYK H M, RABIN T D. Chameleon hashing and signatures[P]. US Patent 6108783, 2000.

[9] QIN B, CHEN L C H, WU Q H, et al. Bitcoin and digital fiat currency[J]. Journal of Cryptologic Research, 2017, 4(2): 176−186. [DOI: 10.13868/j.cnki.jcr.000172]
秦波, 陈李昌豪, 伍前红, 等. 比特币与法定数字货币 [J]. 密码学报, 2017, 4(2): 176–186. [DOI: 10.13868/j.cnki.jcr.000172]

[10] SURHONE L M, TENNOE M T, HENSSONOW S F. Hash Function[M]. 2010.

[11] KRAWCZYK H M, RABIN T D. Chameleon signatures[C]. In: Network and Distributed System Security Symposium (NDSS 2000). San Diego, CA, USA. 2000: 143–154.

[12] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 24(11): 612–613. [DOI: 10.1145/359168.359176]

[13] BLAKLEY G R. Safeguarding cryptographic keys[C]. In: Proceedings of the National Computer Conference, Montvale, NJ, USA. 1979: 313. [DOI: 10.1109/AFIPS.1979.98]

[14] CHOR B, GOLDWASSER S. Verifiable secret sharing and achieving simultaneity in the presence of faults[C]. In: Proceedings of 26th IEEE Symposium on Foundations of Computer Science. Portland, OR, USA. 1985: 383–395. [DOI: 10.1109/SFCS.1985.64]

[15] FELDMAN P. A practical scheme for non-interactive verifiable secret sharing[C]. In: Proceedings of 28th IEEE Symposium on Foundations of Computer Science. Los Angeles, CA, USA. 1987: 427–438. [DOI: 10.1109/SFCS.1987.4]

[16] PEDERSEN T P. Non-interactive and information-theoretic secure verifiable secret sharing[C]. In: Advances in Cryptology—CRYPTO 1991. Springer Berlin Heidelberg, 1991: 129–140. [DOI:10.1007/3-540-46766-1_9]

[17] BENOR M, GOLDWASSER S, WIDGERSON A. Completeness theorems for noncryptographic fault-tolerant distributed computation[C]. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. Chicago, IL, USA. 1988: 1–10. [DOI: 10.1145/62212.62213]

[18] CRAMER R, DAMGÅRD I, MAURER U. General secure multi-party computation from any linear secret-sharing scheme[C]. In: Advances in Cryptology—EUROCRYPT 2000. Springer Berlin Heidelberg, 2000: 316–334. [DOI: 10.1007/3-540-45539-6_22].

[19] WANG X, WENG J, ZHANG Y, et al. Blockchain system for creating digital assets based on reputation value[J]. Netinfo Security, 2018, 18(5): 59–65. [DOI: 10.3969/j.issn.1671-1122.2018.05.007]
王醒, 翁健, 张悦, 等. 基于信誉值创建数字资产的区块链系统 [J]. 信息网络安全, 2018, 18(5): 59–65. [DOI: 10.3969/j.issn.1671-1122.2018.05.007]

[20] HE P, YU G, ZHANG Y F, et al. Survey on blockchain technology and its application prospect[J]. Computer Science, 2017, 44(4): 1–7. [DOI: 10.11896/j.issn.1002-137X.2017.04.001]
何蒲, 于戈, 张岩峰, 等. 区块链技术与应用前瞻综述 [J]. 计算机科学, 2017, 44(4): 1–7. [DOI: 10.11896/j.issn.1002-137X.2017.04.001]

[21] LIN X F. Blockchain technology applictions in finacial industry[J]. China Finance, 2016, 2016(8): 17–18.
林晓轩. 区块链技术在金融业的应用 [J]. 中国金融, 2016, 2016(8): 17–18.

作者信息

李佩丽 (1988–), 河北人, 博士. 主要研究领域为安全协议、公钥密码、区块链隐私保护.
lipeili@iie.ac.cn

徐海霞 (1973–), 河北人, 博士, 副研究员. 主要研究领域为安全多方计算、数字货币、区块链隐私保护.
xuhaixia@iie.ac.cn

马添军 (1993–), 山西人, 博士研究生. 主要研究领域为区块链隐私保护、公钥密码、智能合约.
matianjun@iie.ac.cn

穆永恒 (1994–), 河北人, 硕士研究生. 主要研究领域为数字货币、区块链共识机制.
muyongheng@iie.ac.cn