计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(2):69−77 [doi: 10.15888/j.cnki.csa.008309]

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

基于可修改区块链的互联网码号资源管理方案①

樊松委[1,2], 陈　越[2], 刘　扬[1,2]

[1](郑州大学 软件学院, 郑州 450003)
[2](中国人民解放军战略支援部队信息工程大学, 郑州 450003)
通信作者: 樊松委, E-mail: fansongwei2022@163.com

摘　要: 为加强 IP 地址、自治域号等国际互联网码号资源的管理和控制, 国际互联网工程任务组提出了互联网码号资源公钥基础设施, 近年来有效解决路由劫持、路径篡改等问题, 为保证域间路由稳定运行发挥了巨大作用. 然而, 它在互联网码号资源管理模式中存在的安全问题也逐渐突显, 如单点故障、资源分配异常、证书撤销数据同步不及时造成验证失效等. 本文针对上述安全问题, 提出了一种基于可修改区块链的互联网码号资源管理方案, 并通过实验验证了该方案的有效性和可行性.

关键词: 资源公钥基础设施; 区块链; 互联网码号资源; 资源分配异常

引用格式: 樊松委,陈越,刘扬.基于可修改区块链的互联网码号资源管理方案.计算机系统应用,2022,31(2):69–77. http://www.c-s-a.org.cn/1003-3254/8309.html

Management Scheme of Internet Number Resources Based on Modifiable Blockchain

FAN Song-Wei[1,2], CHEN Yue[2], LIU Yang[1,2]

[1](School of Software, Zhengzhou University, Zhengzhou 450003, China)
[2](The PLA Strategic Support Force Information Engineering University, Zhengzhou 450003, China)

Abstract: To strengthen the management and control of Internet number resources such as IP addresses and autonomous system numbers (ASNs), the Internet engineering task force (IETF) proposes resource public key infrastructure (RPKI). In recent years, it has effectively solved the problems of route hijacking and path tampering and plays a crucial role in ensuring the stable operation of inter-domain routing. However, the security problems in the RPKI management mode are gradually highlighted, such as single point of failure, abnormal resource allocation, and verification failures caused by the poor synchronization of certificate revocation data. To tackle these problems, this study proposes a scheme for managing Internet number resources based on modifiable Blockchain. The experimental results show that the scheme is effective and feasible.

Key words: resource public key infrastructure (RPKI); Blockchain; Internet number resources; abnormal resource allocation

资源公钥基础设施 (resource public key infrastructure, RPKI)[1] 作为有效管理互联网码号资源 (Internet number resource, INR, 以下简称“资源”) 的专项技术, 已有多个国家和地区参与测试部署. 然而其在资源管理过程中存在各种安全问题[2], 如单点故障、资源分配异常、证书撤销数据同步不及时造成验证失效等. 针对上述问题, 本文提出了一种基于可修改区块链的互联网码号资源管理方案 (resource management by modifiable Blockchain, RMMB). 该方案将 RPKI 中的资源管理节点加入到区块链中以规避单点故障风险. 同时为了实现资源在区块链机制下高效的自动化管理, RMMB 在各 CA 节点部署动态变化的可分配资源池 (allocable

① 基金项目: 国家自然科学基金 (61502528)
收稿时间: 2021-04-14; 修改时间: 2021-05-11; 采用时间: 2021-05-28; csa 在线出版时间: 2022-01-17

resource pools, ARP), 并引入实现证书生成、撤销、验证等功能的智能合约到资源的管理之中. 实验表明该方案不仅可以解决单点故障与资源分配异常问题, 还消除了证书撤销数据同步存在的隐患, 使 RPKI 的资源管理变得更加安全高效.

1 问题描述和相关工作

1.1 RPKI 原理及运行机制

RPKI 是一种用来保障互联网基础码号资源安全使用的公钥基础设施, 它依托层次化体系对资源进行分配, 主要通过两种 X.509 证书来实现分配过程的可认证[3], 分别是认证权威 (certificate authority, CA) 证书和端实体 (end entity, EE) 证书. CA 证书用于实现互联网码号资源所有权的认证, EE 证书用于对路由源认证 (route origin attestation, ROA) 的签名验证. RPKI 由 CA、资料库 (repository) 和依赖方 (relying party, RP) 3 个基本模块构成, 其中 CA 机构负责将发布的证书和相关数据信息发送到资料库中, RP 负责同步资料库中的数据, 并对 ROA 进行验证. BGP 路由器则根据验证结果构建自己的过滤表项和缓存列表, 并通过这些表项来判断宣告路由是否有效来指导其路由决策[4], RPKI 的运行机制如图 1 所示.

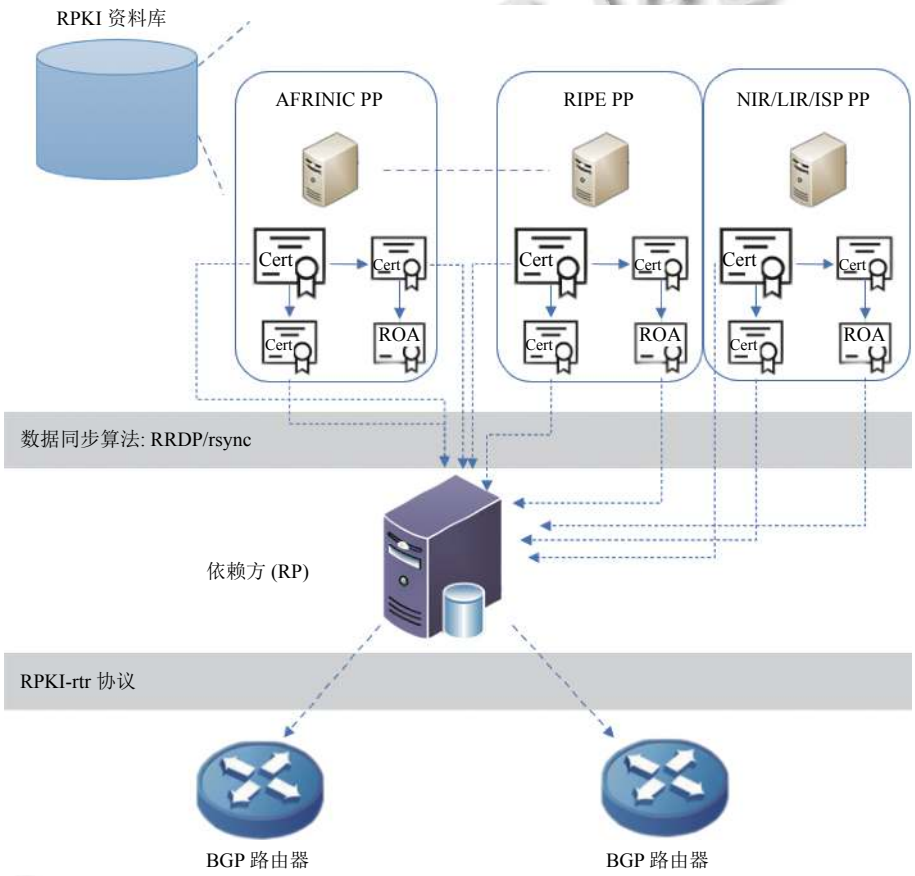

图 1 RPKI 运行机制

1.2 RPKI 存在的问题

RPKI 可以有效防御路由前缀劫持攻击, 然而其在资源管理模式中却存在较多的隐患, 导致 RPKI 的部署成本过高且难以较好实现预期目标[5], 总结起来有如下 3 个方面的问题: 单点故障、资源分配异常、证书撤销数据同步不及时造成验证失效. 单点故障主要是由于 RPKI 上层 CA 节点权力过大, 一旦上层节点被恶意攻击, 便会造成极大的安全隐患. 证书撤销方面的问题主要是由于 RPKI 采用的 CRL 机制[6], 其及时性有限、冗余度过高、存储及验证开销过大等缺陷会导致 CRL 列表同步不及时, 进一步造成过期证书被恶意利用等安全问题. RPKI 资源分配过程中出现的错误操作可能导致各种安全风险[7], 例如将未授权的资源分配给下级节点, 会导致该节点获取本不属于自己的路由信

息, 造成路由泄露; 恶意新增一条路由源认证信息, 会导致一条合法的路由被判定为无效; 重复分配已授权的资源, 可能会导致资源冲突或不可用等. Fu 等[8]针对 RPKI 中 CA 资源分配过程中可能出现的异常, 将其分为 3 种情况, 分别是未经授权资源分配、资源再次分配和资源转移.

未经授权资源分配是指 CA 节点将未经上级授权的码号资源分配给其下级节点的行为, 其又细分为完全未经授权资源分配和部分未经授权资源分配两种情况. 资源再次分配是指 CA 节点将已经分配给某下级节点的码号资源又重复分配给其他下级节点的行为, 其又细分为 Matching 类型 (重复分配给多节点的资源集合完全相同)、Subset 类型 (重复分配给多节点的资源集合属于包含关系) 和 Intersection 类型 (重复分配给多节点的资源集合存在交集). 资源转移是两个互联网机构通过第三方认证进行的资源交易过程, 在交易双方达成一致的前提下, 经过旧证书的撤销和新证书的颁发, 确立该资源新的所有权. 将该过程进行一定程度的简化, 如图 2 所示.

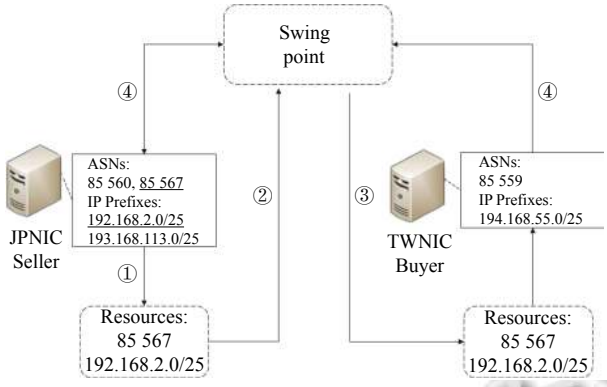

图 2 资源转移

JPNIC 与 TWNIC 双方均同意要对某部分资源进行转移, 过程如下: 两者选取一个 RPKI 层次结构中最低的 IR 节点, 称为 SP (swing point), 且该节点为双方的共同父节点, 并且同意作资源转移代理. JPNIC 创建一个证书并告知 SP, 描述要转移给 TWNIC 的资源; SP 向 TWNIC 发出一个新的扩展证书, 描述其资源管理情况以及新资源的详情, 当双方在确定技术和业务方面的转移都已经完成时会通知 SP, 待相关资源转移完成后 SP 会向 JPNIC 颁发新证书, 以更新 JPNIC 的资源管理情况.

1.3 相关工作

针对上述问题, 文献 [8] 提出滞后验证资料库的数字签名对象, 当出现异常时, 及时通知 CA 进行错误纠正, 但该方法需要一个错误纠正等待时间; 为了减少该等待时间, 文献 [9] 提出了一种“事前控制”的检测机制, 通过修改 rpki.net 提供的 RPKI-CA 工具在证书颁发之前对颁发的资源进行验证, 发现不符合规定的分配操作则不予执行. 但是其只能检测出资源未经授权分配以及 Matching 或 Subset 类型的资源再次分配, 且如果验证者被事先攻击, 则可绕过该检测机制使其无效, 故其存在单点故障和检测问题不全面的缺陷. 文献 [10] 针对这些不足提出了基于区块链的检测方案, 该机制通过将 CA 机构作为节点加入区块链来解决单点故障的问题, 利用了智能合约和哈希值数组的证书存储结构来保障机制的安全高效, 并通过实验证明了其安全性和可行性. 该机制虽然解决了单点故障的问题, 且能够进一步检测出 Intersection 类型的资源再次分配, 但是该方案未考虑出现证书撤销情况后检测技术的应对方案. 当资源所属权发生变更, 链上数据却依然保持变更之前的状态, 可能会导致原本正常的资源分配操作被智能合约检测为异常; 由于该机制只考虑对上传至区块链上的证书进行检测, 当出现链上证书过期的情况, 边界路由器却依旧信任, 这将会产生较大的安全风险.

而本文提出的基于可修改区块链的互联网码号资源管理方案同样利用了文献 [10] 中区块链技术的优点, 如去中心化、共识机制保持数据一致性、智能合约的应用等. 此外, 补充考虑了证书撤销机制, 利用区块的可修改来适应 RPKI 运行过程中数据的动态变化, 较好地解决了上述方案存在的问题.

2 可修改区块链技术

2.1 可修改区块链技术的提出背景

区块链技术去中心化、不可篡改、可追溯、公开透明等特性为很多应用提供了安全可靠的信任环境. 这也使得近些年区块链的关注度不断上升, 然而区块链上数据的绝对不可更改也给其发展带来限制, 近些年出现的一系列以太坊智能合约漏洞攻击事件为此敲响了警钟. 如 2016 年的 The DAO 攻击事件[11]、2017 年的 Parity 钱包被盗事件[12] 以及多起由整数溢出漏洞导致的合约攻击事件. 官方机构最终也只能通过软硬分叉的方式来解决问题, 这些事件不仅给用户带来极大的经济财产损失, 也给区块链技术带来负面影响. 此外, 区块链缺乏治理规则, 当遇到突发情况 (如代码漏洞、记录出错

等) 容易导致系统混乱. 因此, 在保证安全的前提下, 在特定情况下允许区块链上的数据被修改, 对于区块链的健康发展和其抗风险能力的提升具有积极意义.

2.2 可修改区块链技术方案的选择

目前关于可修改的区块链技术研究刚刚起步, 相关概念最早是由埃森哲公司提出, 其设计了利用变色龙哈希技术来更改区块链中的历史区块[13]. 该方案虽然带来了利用变色龙哈希技术来实现可修改区块链的启发, 但存在较大的中心化风险, 与我们使用区块链方案解决单点故障问题的初衷相违背.

Cheng 等[14] 提出了基于多项式的可修改区块链结构, 在每个块中通过拉格朗日插值方法组织数据段, 多项式函数用于保持块的顺序, 还可以通过选择适当的多项式幂和格式来调整修改的难度. 但是该方案的设计初衷是为了解决链上存在的金融欺诈交易信息类问题, 且需要人工对上链数据进行分类以确定其修改难度, 应用于 RPKI 的资源管理中契合度不高.

Lee 等[15] 提出了另一种构建可修改区块链的方法, 通过交易内容的哈希散列值和分散的对等网络节点的集体贡献来批准和修改交易, 其使用分层的多区块链模型作为在分散网络中修改交易的工具, 每个侧链的区块挖掘过程与主链以及其他侧链的区块挖掘过程几乎独立地进行. 该方案只进行了相关的理论安全分析没有进行实验验证其可行性, 且方案理论上涉及部署多条区块链, 在修改频繁的场景下可能导致跨链通信开销过大, 故不适用于 RPKI 的资源管理.

文献 [16] 基于 POSpace 共识机制的 SpaceMint 区块链系统, 通过在区块签名子块中引入机动因子的方式, 利用陷门单项函数, 在超过阈值数节点同意的前提下, 验证群组中的矿工节点利用各自的陷门生成新的机动因子, 完成对区块信息的更改, 其余区块不受影响, 整个区块链系统仍旧保持健壮性和安全性. 并通过实验验证了其可行性, 且该方案的细粒度更高, 可以只对区块内的某条交易进行修改, 适用于 RMMB 中 ARP 和链上证书撤销等应用场景. 综上所述, 本文最终选择了该项可修改区块链方案来应用于所提出的互联网码号资源管理方案.

2.3 基于 SpaceMint 的可修改区块链原理

Park 等提出了以空间共识, 即 POSpace 作为共识机制的 SpaceMint 区块链系统[17]. 不同于 POW 和 POS, 这种共识机制以节点付出的磁盘空间来作为代价证明, 通过节点对有向无环图的构造速度来衡量空间大小, 进而选取记账者. 为了使得在 POSpace 下区块链系统的安全性得以保障, SpaceMint 构建了全新的区块链结构, 如图 3 所示, 区块 i 包含 3 个部分: 证明子块 φ_i、签名子块 σ_i 和交易子块 τ_i. 其中, 证明子块 φ_i 和交易子块 τ_i 相当于传统区块结构中的区块头和区块体, 而新增的签名子块 σ_i 则打破了区块头、体之间的链接, 这样的结构可以抵御多种针对所需算力较小的区块链共识机制的系统攻击.

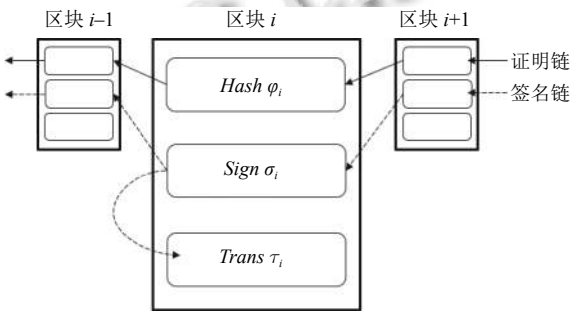

图 3 SpaceMint 区块链结构

图 3 中证明子块 φ_i=$Hash(I, \zeta_\varphi, (p_{ki}, \gamma_i, c_i, a_i))$, 符号具体含义如下: 当前区块号 I, 记账者对前一区块的证明子块 φ_{i-1} 的签名 ζ_φ, 记账者在竞争记账权时产生的承诺证明以及空间证明 $(p_{ki}, \gamma_i, c_i, a_i)$; 签名子块 σ_i 包含 $\{I, \zeta_\tau, \zeta_\sigma\}$, 符号具体含义如下: 当前区块号 I, 记账者对当前区块的交易子块 τ_i 的签名 ζ_τ, 记账者对前一区块的签名子块 σ_{i-1} 的签名 ζ_σ; 交易子块 τ_i 包含 $\{I, ctx\}$, 符号具体含义如下: 当前区块号 I; 交易信息列表 ctx; 该区块结构打破了区块头与区块体的直接联系, 为区块链的修改提供了可能.

其可修改的原理如图 4 所示: 通过在区块签名子块中引入机动因子 G_i 的方式, 利用陷门单项函数, 在超过阈值数节点同意的前提下, 验证群组中的矿工节点利用各自的陷门生成新的机动因子, 完成对区块信息的更改, 其余区块不受影响, 整个区块链系统仍旧保持健壮性和安全性. 可修改区块链区块的生成和修改主要过程如过程 1 和过程 2.

过程 1. 可修改区块链新区块 i 生成

1) 通过 POSpace 共识机制选取记账者和挖矿能力排名在前 80% 的矿工;
2) 记账者计算区块 i 的证明子块 φ_i;
3) 记账者计算区块 i 的签名子块 $\sigma_{i,G}$, 引入机动因子 G_i;

$G_i(x_i^1,x_i^2,\cdots,x_i^n)=g_{P_i^1}(x_i^1)\|g_{P_i^2}(x_i^2)\|\cdots\|g_{P_i^n}(x_i^n)$, 其中 P 为矿工的公钥, g 为 ECC-200 函数, x 为矿工专属随机数 (每个矿工节点在初始化阶段均会得到系统随机分配的公私钥对和专属随机数), 机动因子由排名前 80% 的矿工共同生成. 将记账者对当前区块的交易子块 τ_i 的签名改为记账者对交易子块 τ_i 的哈希值与机动因子 G_i 异或结果的签名, 其余不变;
4) 记账者计算区块 i 的交易子块 τ_i;
5) 记账者将区块 i 打包并发布, 经全网验证通过, 区块上链.

过程 2. 可修改区块链区块 i 修改

1) 节点因某种原因要修改区块 i 中的某条交易信息, 随即生成修改请求, 全网广播;
2) 参与区块 i 生成的矿工对修改请求进行合法性认证;
3) 认证合法后, 根据新的交易子块 τ'_i, 利用如下公式:

$Hash(\tau_i)\oplus G_i(x_i^1,x_i^2,\cdots,x_i^n)=Hash(\tau'_i)\oplus G'_i(x'^1_i,x'^2_i,\cdots,x'^n_i)$

计算得到新的机动因子 G'_i;
4) 各矿工通过函数陷门计算出新的随机值 x', 随后在全网更新;
5) 区块 i 的记账者根据修改情况生成一条溯源信息放入交易池, 等待后续打包上链.

基于 SpaceMint 的可修改区块链通过单向陷门函数保证计算安全, 多数矿工集体参与保证修改的合法性, 即修改代表整体的意识. 虽然相较于传统区块链涉及了更多的计算, 增大了一定的安全风险, 但是其灵活性和可用性都大大增强, 这对于区块链技术的发展具有积极意义.

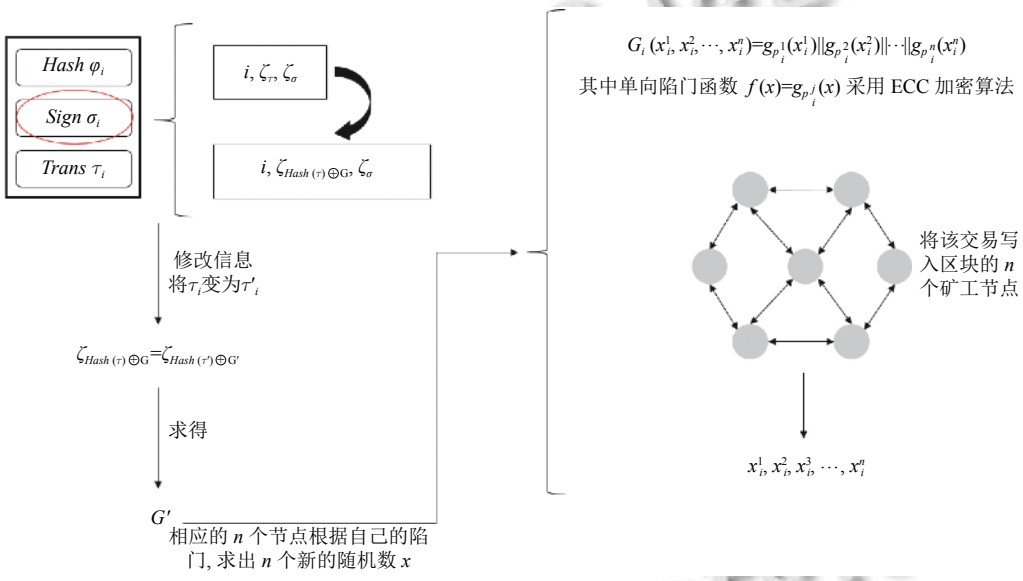

图 4 可修改的 SpaceMint 区块链原理

3 RMMB

3.1 RMMB 概述

RMMB 是一种基于可修改区块链, 依赖于 BGP 协议, 通过智能合约在系统内自动化执行的互联网码号资源管理方案. RMMB 主要分为两个模块: 数据模块和验证模块. 其中数据模块指的是通过智能合约的算法设计、可修改区块链的安全机制以及链上数据的保护机制等确保链上的证书数据真实、有效、合法. 验证模块主要涉及边界网关路由器的验证操作, BGP 路由器在接收到宣告报文后, 首先查询本地的路由表项, 若没有相关记录则向数据模块请求检验, 并通过数据模块的验证结果来更新本地路由表项以指导其路由决策. 数据模块为 BGP 路由器的验证操作提供了可信数据源, 进而有效解决非法违规的 BGP 报文所造成的路由前缀劫持问题.

图 5 展示了 RMMB 的整体框架. 图右侧展示的是数据模块, 在应用 RMMB 的系统内, 所有 CA 节点都要加入一一对应的区块链网络, 每个 CA 节点的基本配置为专属公私钥对、原始数据集、可分配资源池. 其中专属公私钥对的主要目的是为了验证证书发布者的身份信息; 原始数据集在系统建设之初就会得到确定, 一般情况不会发生改变; 可分配资源池包含了 3 部分信息, 分别是 CA 身份标识 (对应 CA 公钥的哈希摘要值)、可分配的 IP 地址资源、可分配的 AS 码号资源. 上述数据均保存在由可修改区块链技术构造的区块中.

同时, 智能合约根据相关条件判断, 自动执行发布者身份验证, 证书内容验证, 区块内容合法修改等步骤, 确保所有链上数据的真实有效. 图左侧展示的是 RMMB

验证模块. 在 BGP 协议中, 相邻路由器会互相传达接收到的相关 BGP 报文, 在图中 BGP 路由器接收到来自相邻路由器且本地路由表项没有记录的更新报文后, 会立刻将相关验证申请信息发送给数据模块, 智能合约在捕捉到信息后, 会启动链上搜索迅速得出该宣告报文合法性的验证结果, 随后利用系统的私钥签名后返回, BGP 路由器签名验证通过后按照规则更新路由表. 由于更新路由表操作需要一定的时间, 在面对本地无法判断的更新报文时, BGP 路由器管理者根据情况将其设置为按默认路由转发或作丢弃处理.

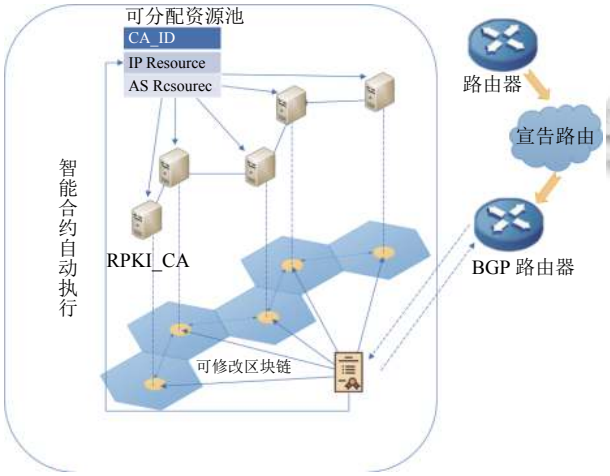

图 5 RMMB 整体框架

3.2 智能合约

RMMB 智能合约系列主要包含 3 方面的执行内容: 验证 CA 节点请求的真实性和合法性、调用底层区块链区块生成和修改的权限、检索链上数据. 应用场景涉及 CA 节点证书生成、证书撤销以及 BGP 路由器发起的证书验证等过程. 智能合约设计思路如下:

智能合约 1. CA 节点证书生成

1) 对 CA 节点 m 提交的资源证书生成请求 (请求分配给下级 CA 节点 n 的 IP 资源为集合 a, AS 资源为集合 b) 进行签名验证;
2) 验证成功后, 通过 CA 身份标识获取 m 的 ARP 信息 (可分配的 IP 资源为集合 c, AS 资源为集合 d), 假设 n 的 ARP 为空值;
3) 对比判断是否满足如下条件: $a \subseteq c \cap b \subseteq d$;
4) 若满足, 则同意该请求, 调用区块链区块生成或修改权限, 将 m 的 ARP 修改 (可分配的 IP 资源为集合 c-a, AS 资源为集合 d-b), 将 n 的 ARP 修改 (可分配的 IP 资源为集合 a, AS 资源为集合 b);
5) 生成相应证书放入交易池, 等待后续打包上链.

智能合约 2. CA 节点证书撤销

1) 对 CA 节点 m 提交的证书撤销请求 (请求撤销下级 CA 节点 n 的 IP 资源为集合 a, AS 资源为集合 b) 进行签名验证;
2) 验证成功后, 通过 CA 身份标识获取 m 的原始数据集 (IP 资源为集合 c, AS 资源为集合 d), m 的 ARP 信息 (可分配的 IP 资源为集合 e, AS 资源为集合 f), n 的 ARP 信息 (可分配的 IP 资源为集合 g, AS 资源为集合 h);
3) 判断是否满足如下条件: $a \subseteq c \cap b \subseteq d$;
4) 若满足, 则同意该请求, 调用区块链区块生成或修改权限, 将 m 的 ARP 修改 (可分配的 IP 资源为集合 e+a, AS 资源为集合 f+b), 将 n 的 ARP 修改 (可分配的 IP 资源为集合 g-a, AS 资源为集合 h-b);
5) 将链上的目标证书执行撤销操作, 即修改为空值.

智能合约 3. BGP 路由器证书验证

1) 对 BGP 路由器提交的证书验证请求进行解密;
2) 解密成功后, 提取请求中相应的特征值;
3) 利用特征值在链上检索相对应的证书;
4) 通过链上是否存在对应的目标资源证书来判断 BGP 路由器收到的报文是否合法, 并生成路由表项;
5) 将验证结果返回给 BGP 路由器, 指导其路由决策.

从智能合约 3 中提取算法 1, 设此时链上的证书数量为 n, 智能合约首先对某 BGP 路由器的验证请求进行解密, 接着获取请求中格式标准的<AS, IP>, 若上述过程出现问题, 均返回 False; 随后遍历检索所有证书, 若存在与<AS, IP>匹配的证书, 且此时其未处于正在被修改的状态, 返回 True, 若其处于被修改的状态则等待其修改完成后再进行匹配, 若匹配则依然返回 True, 否则返回 False. 经过对算法 1 的分析, 可以得出算法的复杂度主要与链上证书数量 n 有关, 故该算法 $T(n)=O(n)$, $S(n)=O(n)$, 其中, T 和 S 分别表示时间复杂度和空间复杂度.

算法 1. 链上证书验证

输入: 系统公钥加密的验证请求 Cryptograph
输出: <AS, IP>的验证结果<True/False>

```
1) if Decrypt (Cryptograph, Private_key) is invalid then
2)    return False
3) if Get (<AS, IP>) is invalid then
4)    return False
5) for i=0 to n do
6)    if <AS, IP> in Certificate on the blockchain then
7)       if this Certificate in modifying then
8)          wait until the end and to step 6
9)       else return True
10) else return False
```

4 实验与分析

本节进行仿真实验, 模拟基于可修改区块链的互联网码号资源管理方案并用相关实验结果与同类方案

进行对比分析. 可修改区块链技术所涉及的密码学参数在 Visual Studio 2017 环境下使用 C++语言进行计算, 计算涉及到的函数有 SHA-256、DSA-512、ECC-200 等. 使用 Intel(R)Core(TM)i7-7500U CPU(2.70 GHz, 16 GB memory) 模拟挖矿节点, 区块链的结构生成以及智能合约的算法设计用 Python 3.7 来实现, 数据存储采用 Key-Value 形式.

设计 5 个节点进行 POSpace 挖矿竞争, 通过各节点存储有向无环图顶点的效率模拟空间大小证明, 其最终竞争结果按照从高到底排名为 4、5、2、1、3, 验证群组阈值设定为 80%. 实验场景设计如下: APNIC 所管理的资源范围为{ASNs: 85 550-85 580; IP Prefixes: 192.168.2.0/24、193.168.113.0/24、194.168.55.0/24}, 需给 JPNIC 分配资源{ASNs: 85 560、85 567; IP Prefixes: 192.168.2.0/25、193.168.113.0/25}, 给 TWNIC 分配资源{ASNs: 85 559; IP Prefixes: 194.168.55.0/25}.

4.1 可行性实验

根据上述设计, 以 JPNIC 视角为例, 假设 JPNIC 在资源分配之前 ARP 为空值. APNIC 节点开始提交资源证书生成请求, 首先智能合约会对该请求进行签名验证; 验证通过后, 判断资源分配{ASNs: 85 560、85 567}⊆{ASNs: 85 550–85 580}合法, 同时 IP 资源也满足条件; 此时开始调用区块生成或者修改权限来对 JPNIC 和 APNIC 的 ARP 进行修改, 假设此时通过生成新区块实现, 以区块 826 为例, 记账者为矿工 4, 此次挖矿排名交易信息 ctx={ranking, 8CD85AE8, 4, 5, 2, 1, 3}, 其包含证明子块 φ_{826}、签名子块 $\sigma_{826,G}$ 和交易子块 τ_{826}, 由实验测得: φ_{826}=e5258fd7afac42f5e586b85aef00c7de7cd780b5224fe9b32561558533376108, τ_{826}={826, ctx}, 其中 ctx 可以包含 25–30 条交易信息, $\sigma_{826,G}$={826, 3C5F70EDDD51270F1854D6455338CF5FF04CBBC3, 8D87C8A1CDE7F78CB4F5AB1C78790890A955AD8}; 最后生成相应的 X.509 证书, 与上述交易信息一起进入交易池中, 等待后续矿工打包上链.

在上述资源证书生成后, JPNIC 的 ARP 更改为{ASNs: 85 560、85 567; IP Prefixes: 192.168.2.0/25、193.168.113.0/25}, 随后经过 JPNIC 与 TWNIC 进行协商, JPNIC 决定将资源{ASNs: 85 567; IP Prefixes: 192.168.2.0/25}转移给 TWNIC, 如图 2 所示. 则此时需要 JPNIC 将已经颁发的证书执行撤回操作, JPNIC 节点开始提交证书撤销请求, 首先智能合约会对该请求进行签名验证; 验证通过后, 判断将要撤回的资源属于 JPNIC 的原始数据集, 撤回请求合法; 此时开始调用区块生成或者修改权限来对 JPNIC 的 ARP 进行修改, 假设此时通过修改信息所在的区块实现, 首先 JPNIC 节点向全网广播一条交易更改请求, ReviseTx={826, ARP_Change, (JPNIC_ARP; ASNs: 85 560、85 567; IP Prefixes: 192.168.2.0/25,193.168.113.0/25), (JPNIC_ARP; ASNs; 85 560; IP Prefixes: 193.168.113.0/25)}, 对应的矿工节点 4、5、2、1 收到更改请求后, 验证其是否合法, 验证通过后开始进行修改操作, 生成新的交易子块 τ'_{826}, 然后根据公式计算出新的机动因子 G', 随后 4 名矿工节点使用各自的私钥进行 ECC-200 解密, 求出新的专属随机值 x', 并在全网更新; 最后将修改情况生成一条溯源信息放入交易池, 等待后续打包上链. 以节点 4 为例, 根据新的机动因子求得 $\tau'_{P(4/826)}$=5FF033D5F280D9FD62F0396E8D50D7691450CC9C40505B6010, 然后利用其私钥进行 ECC-200 解密, 得到新的专属随机数 $x'_{(4/826)}$=9E5A4A279E8818C784FE6BC8DE8859BE9B1A5E7D50285BC3. 上述步骤完成后, 开始执行证书撤销操作, 智能合约再次调用区块修改权限对包含证书的区块进行修改, 将区块内的证书信息修改为空值, 在不影响区块其他内容的前提下, 数据已按照要求合法修改.

通过不断模拟生成编号 1–5 的区块, 我们得到了这些区块生成的平均时间开销大约为 3.846 s, 接着对编号 1–5 的区块进行了修改测试, 得出区块修改平均时间为 1.221 s. 根据图 6 可以看出 RMMB 单个区块生成的时间基本维持在 4 s 以内, 区块修改时间为区块生成时间的 1/3 左右.

在证书验证实验中, 我们模拟了智能合约在链上对目标证书检索的过程, 得到了从不同证书数量规模的区块链上检索到目标证书的时间开销, 将 RMMB 作为一个黑匣子, 从 BGP 路由器的角度来检测系统的效果. 如图 7 所示, 在链上证书数量由 1 000 到 10 000 递增的过程中, 系统验证的时间开销也随之增加. 可以看出在链上证书数量为 10 000 时, 单次平均验证时间在 1 s 以内.

接下来进行 RMMB 适用性分析. 在当前 RPKI 机制下, BGP 路由器在获取最新的 ROA 路由过滤表时, 需要 RP 先通过同步协议同步 RPKI 资料库的最新数据, 随后根据最新数据提取出有效的 ROA 记录生成路

由过滤表项, 最后再通过 RTR 协议将表项下发至各个 BGP 路由器[18]. 而现阶段 RP 软件同步 RPKI 资料库的默认刷新间隔通常在 1 min 到 1 h 不等[19]. BGP 路由器每间隔 1 h 左右通过 RTR 协议从 RP 获取一次最新数据[20], 也就是说 BGP 路由器可以接受的系统更新时间间隔在 1 h 左右.

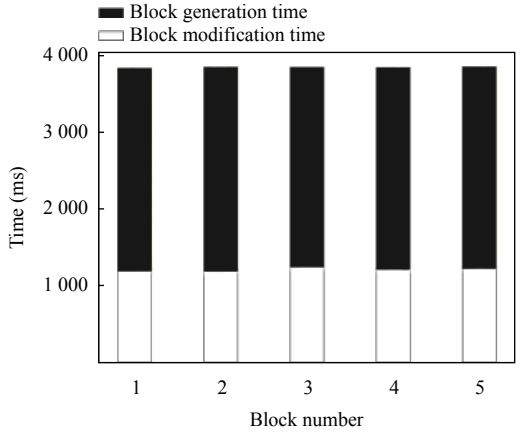

图 6 区块生成与修改时间开销

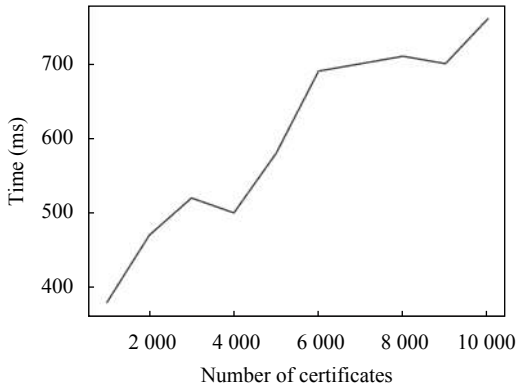

图 7 证书验证时间开销

而在 RMMB 中, 区块链节点同步存储着证书信息, 系统数据自动维护更新. 对 BGP 路由器而言, 只需要向系统发起验证申请即可快速更新本地路由表项, 属于增量式的触发更新. 截至 2021 年 5 月 16 日, RPKI 资料库中约有 107 697 个文件 (包括.cer 文件和.roa 文件), 此时 RPKI 全球部署率为 29.03% (数据来源于 https://rpki-monitor.antd.nist.gov/). 结合验证算法的时间复杂度及实验数据, 粗略推测链上证书数量达到 107 697 时, 单次平均验证时间在 15 s 之内. 假设在最坏情况下, 即<AS, IP>匹配到的证书恰好在修改状态下的时间开销 $T = t_1 + 2t_2 + t_3 + t_4 + t_5 + t_6 + t_7$ (其中链上证书验证时间t_1约为 15 s, BGP 路由器与 RP 之间的信息传输时间t_2约为 0.2 s. 加解密以及签名验证使用 RSA-1024 算法, 测得 BGP 路由器公钥加密的平均时间t_3与系统私钥签名的平均时间t_4都约为 0.052 s, BGP 路由器私钥解密的平均时间t_5与系统公钥验证的平均时间t_6都约为 0.018 s, 区块修改时间t_7约为 1.221 s), 计算得 BGP 路由器平均每次获取路由表更新的时间开销 T=16.76 s. 随着 RPKI 部署率的不断提升, RMMB 链上证书数量也会不断上升, t_1也会随之增加. 根据上述单次平均验证时间预测, 当其全球部署率达到 50% 时, t_1约增长到 25.86 s, 假设其余时间消耗均不变, 此时 T=27.62 s.

通过以上数据分析得出, 对 BGP 路由器而言, 相比于现行 RPKI 机制下路由表的周期性更新, 通过本文所提出 RMMB 机制下更新的时间开销不论是目前还是在未来一段时间均在合理的范围之内. RMMB 能够安全高效地帮助 BGP 路由器与最新且可靠的数据源达到同步, 具有较强的可行性.

4.2 有效性分析

对 RPKI、文献 [9]、文献 [10] 和 RMMB 四种方案在上文所提出的 3 方面问题, 即单点故障、证书撤销数据同步不及时、资源分配异常, 进行功能性分析. 由于所有方案都是围绕着 RPKI 进行改进, 故将 RPKI 作为对照, 其所有类型的问题在理论上都存在; 对于单点故障问题, 由于文献 [9] 所提出的检测机制未改变 RPKI 层次化的体系结构, 故依旧存在单点故障风险, 一旦 CA 节点被入侵, 检测机制可能面临失效. 文献 [10] 和本文所提方案都是基于区块链的分布式架构, 改变了 RPKI 层次化的体系结构, 均很好地解决了单点故障问题. 对于证书撤销方面可能存在的安全风险, 文献 [9] 和文献 [10] 的主要侧重点在于对资源异常分配的检测和管理, 其中文献 [9] 依旧延续 RPKI 的证书撤销规则, 依旧存在证书撤销同步不及时造成的安全问题. 文献 [10] 只考虑了将区块链作为 RPKI 资料库, 证书经过安全验证才能合法上链, 却未能考虑证书上链持久化存储后的一些列情况, 如过期证书撤销、身份信息发生变更证书的处理等, 存在较大的安全隐患. RMMB 则全面考虑了上述问题, 设计了安全高效的智能合约, 并将证书撤销涉纳入到链上数据管理中, 很好地解决了证书撤销数据不同步的问题.

资源分配异常细分为多种情况, 文献 [9] 由于只进行资源未经授权分配检测、Matching 类型资源再次分配和 Subset 类型资源再次分配检测, 所以未能解决

Intersection 类型资源再次分配问题; 在资源转移的情况下, 由于被转移资源属于未授权资源类型, 故文献 [9] 可以检测出异常, 能够解决资源转移的问题. 文献 [10] 和本文所提出的方案均全面考虑了资源未经授权分配和资源再次分配的各种情况, 故均能成功解决上述问题; 在资源转移的情况下, 文献 [10] 中关于被转移资源的证书已颁发至区块链上, 且该机制未考虑出现此情况下的应对方案, 所以未能解决此问题, 而 RMMB 通过 ARP 在区块链上的合法动态改变, 能够检测出资源异常, 使该问题得到了解决. 将最终结果汇总如表 1 所示, 可以看出 RMMB 相较于其他方案在解决各种问题方面都具有更好的表现.

表 1 有效性分析结果

方案是否解决以下问题	RPKI	文献[9]	文献[10]	本文
单点故障	×	×	√	√
证书撤销数据不同步	×	×	×	√
资源未经授权分配	×	√	√	√
Matching&Subset类型资源再次分配	×	√	√	√
Intersection类型资源再次分配	×	×	√	√
资源转移	×	√	×	√

5 总结

本文提出了将可修改区块链作为底层技术并结合智能合约的互联网码号资源管理方案 RMMB. 通过可分配资源池机制的设计, 将上述技术的优势有机结合起来. 在应对互联网码号资源管理存在的问题时, 不仅较传统 RPKI 方案更加安全, 和现有方案相比也具有更高的实用性. 该方案建立在 RPKI 的框架下, 为边界 BGP 路由器提供了及时同步且可靠的资料库, 所需要的时间成本也在合理范围之内. 下一步的工作方向是在应用层面, 利用真实的历史数据来研究该方案对于路由劫持、路径篡改等实际问题的解决情况.

参考文献

1 马迪. RPKI 概览. 电信网技术, 2012, (9): 30–33.

2 Osterweil E, Manderson T, White R, *et al*. Sizing estimates for a fully deployed RPKI. Technical Report, Califorlia: Verisign Labs, 2012.

3 Lepinski M, Kent S. An infrastructure to support secure internet routing: IETF RFC 6480. IETF Trust, 2012: 1–24.

4 Bush R. Origin validation operation based on the resource public key infrastructure (RPKI): IETF RFC 7115. IETF Tust, 2014: 1–11.

5 陈迪, 邱菡, 朱俊虎, 等. 区块链技术在域间路由安全领域的应用研究. 软件学报, 2020, 31(1): 208–227. [doi: 10.13328/j.cnki.jos.005867]

6 Yu SC, Wang C, Ren K, *et al*. Attribute based data sharing with attribute revocation. Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. Beijing: ACM, 2010. 261–270.

7 Kent S, Ma D. Adverse actions by a certification authority (CA) or repository manager in the resource public key infrastructure (RPKI): IETF RFC 8211. IETF Trust, 2017: 1–26.

8 Fu Y, Yan ZW, Liu XW, *et al*. Scenarios of unexpected resource assignment in RPKI. IETF Trust, 2015: 1–17.

9 刘晓伟, 延志伟, 耿光刚, 等. RPKI 中 CA 资源分配风险及防护技术. 计算机系统应用, 2016, 25(8): 16–22. [doi: 10.15888/j.cnki.csa.005313]

10 彭素芳, 刘亚萍. 基于区块链的 RPKI 中 CA 资源异常分配检测技术. 网络空间安全, 2019, 10(7): 38–44. [doi: 10.3969/j.issn.1674-9456.2019.07.007]

11 Mehar M, Shier C, Giambattista A, *et al*. Understanding a revolutionary and flawed grand experiment in blockchain: The DAO attack. Journal of Cases on Information Technology, 2019, 21(1): 19–32. [doi: 10.4018/JCIT.2019010102]

12 The Parity Wallet Hack Explained. OpenZeppelin blog. https://blog.openzeppelin.com/on-the-parity-wallet-multisig-hack-405a8c12e8f7/. (2017-07-19).

13 Krawczyk H, Rabin T. Chameleon signatures. Proceedings of the Symposium on Network and distributed system security symposium. San Diego: NDSS, 2000. 143–154.

14 Cheng LC, Liu JQ, Su CH, *et al*. Polynomial-based modifiable blockchain structure for removing fraud transactions. Future Generation Computer Systems, 2019, 99: 154–163. [doi: 10.1016/j.future.2019.04.028]

15 Lee NY, Yang JH, Onik MMH, *et al*. Modifiable public blockchains using truncated hashing and sidechains. IEEE Access, 2019, 7: 173571–173582. [doi: 10.1109/ACCESS.2019.2956628]

16 任艳丽, 徐丹婷, 张新鹏, 等. 可修改的区块链方案. 软件学报, 2020, 31(12): 3909–3922. [doi: 10.13328/j.cnki.jos.005894]

17 Park S, Kwon A, Fuchsbauer G, *et al*. SpaceMint: a cryptocurrency based on proofs of space. 22nd International Conference on Financial Cryptography and Data Security. Nieuwpoort: Springer, 2018. 480–499.

18 苏莹莹, 李丹, 叶洪琳. 资源公钥基础设施 RPKI: 现状与问题. 电信科学, 2021, 37(3): 75–89.

19 Kristoff J, Bush R, Kanich C, *et al*. On measuring RPKI relying parties. Proceedings of the ACM Internet Measurement Conference. New York: ACM, 2020. 484–491.

20 Bush R, Austein R. The resource public key infrastructure (RPKI) to router protocol: IETF RFC 6810. IETF Trust, 2013: 1–27.