

基于区块链的高透明度 PKI 认证协议

陈立全^{1,2}, 李潇¹, 杨哲懿¹, 钱思杰¹

(1. 东南大学网络空间安全学院, 江苏 南京 210096; 2. 网络通信与安全紫金山实验室, 江苏 南京 211111)

摘要: 公钥基础设施 (PKI) 作为互联网空间安全基础设施的重要组成部分, 为互联网的信息传输提供必要的真实性、完整性、机密性和不可否认性。现有的公钥基础设施存在证书颁发机构权力过大、吊销查询困难等问题。随着区块链技术的发展, 可以利用区块链技术去中心化、透明度高、结构扁平等优点来解决上述公钥基础设施存在的问题, 提高整个互联网建立信任关系的能力和效率。因此, 提出基于区块链的高透明度 PKI 认证协议。该协议通过加入门限签名技术提出了改进的实用拜占庭容错共识算法 (TS-PBFT)。TS-PBFT 算法降低了原有实用拜占庭容错 (PBFT, practical Byzantine fault tolerance) 共识算法的通信复杂度, 减少了通信开销; TS-PBFT 算法在视图切换协议的主节点选举引入了外界监督机制, 增加了可监管性; TS-PBFT 算法在快速一致性协议中引入了批处理机制, 提升了共识过程的性能。该协议一方面在提出的 PBFT 算法的基础上引入了区块链技术, 提升了证书吊销查询的安全性, 并引入了计数布隆过滤器, 提升了证书查询的效率; 另一方面, 该协议在证书的生命周期管理中增加了证书审计流程, 对证书颁发机构的行为做出监管, 促使其提高安全标准, 达到限制其权力的目的。安全性分析和效率实验分析表明, 所提协议系统具有抵抗伪装申请证书攻击等安全属性, 与已有 PKI 协议相比在 TLS/SSL 握手耗时上具有优势。

关键词: 区块链; 拜占庭容错; 公钥基础设施; 认证协议

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2022052

Blockchain-based high transparent PKI authentication protocol

CHEN Liquan^{1,2}, LI Xiao¹, YANG Zheyi¹, QIAN Sijie¹

1. School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China

2. Purple Mountain Laboratories for Network and Communication Security, Nanjing 211111, China

Abstract: The public key infrastructure (PKI), as an significant component of the current Internet security infrastructure, guarantees the information transmission with the necessary authenticity, integrity, confidentiality and non-repudiation. However, the existing PKI also has shortcomings of excessive power of certification authority and difficulties in revoking and querying. Blockchain can be used to solve those problems by leveraging its advantages, such as decentralization, high transparency and flat structure. Furthermore, the ability and efficiency of the entire

收稿日期: 2021-08-09; 修回日期: 2022-03-14

通信作者: 陈立全, lqchen@seu.edu.cn

基金项目: 国家重点研发计划项目 (2020YFE0200600)

Foundation Item: The National Key R&D Program of China (2020YFE0200600)

引用格式: 陈立全, 李潇, 杨哲懿, 等. 基于区块链的高透明度 PKI 认证协议[J]. 网络与信息安全学报, 2022, 8(4): 1-11.

Citation Format: CHEN L Q, LI X, YANG Z Y, et al. Blockchain-based high transparent PKI authentication protocol[J]. Chinese Journal of Network and Information Security, 2022, 8(4): 1-11.

Internet to establish trust relationships may be improved. The transparent public key infrastructure (PKI) certification protocol based on the blockchain was proposed. The TS-PBFT algorithm was designed in the proposed protocol by adopting the threshold signature technology to the Practical Byzantine fault tolerance (PBFT) algorithm. The TS-PBFT algorithm reduced the communication overhead via reducing the communication complexity, strengthened the supervision via introducing external monitoring mechanism in the master node election of the view change protocol, and also improved the performance of the consensus mechanism via adding a batch processing mechanism. Moreover, a transparent blockchain-based PKI certification protocol was designed. The proposed protocol increased the security of certificate revocation and query, it also improved the efficiency of the certificate query by the introduction of counting bloom filters. Besides, the proposed protocol added audit function into the certificate lifecycle management. Accordingly, it can supervise the behavior of the certificate authority (CA), prompt it to improve security standards, and then achieve the purpose of limiting its authority. According to the security analysis and efficiency experiments, the proposed protocol was equipped with security properties, such as the resistance to spoofing certificate application attacks, and it achieved the best performance on TLS/SSL handshake time compared with existing PKI protocols.

Keywords: blockchain, Byzantine fault tolerant, public key infrastructure, authentication protocol

0 前言

公钥基础设施(PKI, public key infrastructure)是互联网空间安全基础设施的一个重要组成部分,是一种普适性的网络安全基础设施^[1]。PKI 的目的是在不可信网络中,以公钥密码学为理论基础,以数字证书为沟通媒介建立安全可靠的信任关系,使用户可以在互联网中方便地验证对方用户的身份,从而为互联网上信息的传输提供必要的真实性、完整性、机密性和不可否认性。

近 30 年以来,PKI 体系得到了快速的发展,尤其随着电子商务的兴起,PKI 体系进入了爆发性发展期。但是早期证书颁发机构(CA, certificate authority)多只注重扩大市场份额,而未将安全列入其最高优先级中,成为攻击者最大的攻击目标^[2]。2008 年,有人利用 StartCom 网站的漏洞,绕过了认证,申请并获得了未经授权域名的两张证书^[3]。2011 年, DigiNotar 成为第一家被完全入侵的 CA,导致伪造的证书被大范围使用,产生了非常严重的中间人攻击^[4]。

PKI 体系如今虽然已十分普遍和成熟,但一连串的事件说明其依旧存在缺陷。首先是 CA 权力过大,理论上 CA 有能力不经过域名拥有者的同意随意签发证书。其次是 PKI 体系中证书吊销与查询不关联,PKI 体系虽然提供了证书吊销列表(CRL, certificate revocation list)和在线证书状态协议(OCSP, online certificate status pro-

ocol) 服务用于查询证书吊销状态,但用户在查询过程中无法确保查询的证书就是 CA 查询的证书,换言之,用户无法脱离 CA 自行查询证书状态。最后是证书吊销查询服务的安全性和实用性极低,如 CRL 存在体积过大、更新慢和移动客户端不支持的缺点,OCSP 存在易受重放攻击和响应压制的问题。综上,如何设计行之有效的 CA 监管机制从而改善证书操作的透明度是当前亟待解决的问题之一。

区块链技术由中本聪提出,最早作为比特币的底层技术,其结合诸多现代的计算机技术对数据进行存储、交换和处理,是一种安全性极高的分布式存储技术^[5]。

区块链技术的快速发展,使大量的研究学者越来越重视区块链技术在 PKI 认证领域的潜力。Fromknecht 等^[6]提出了第一个基于区块链技术的 PKI 系统 Certcoin,该系统以比特币区块链系统为底层基础,使用 Certcoin 代替 CA 提供密钥查询和跨域认证功能,但区块链账本记录了用户身份从而造成隐私泄露问题。Axon 等^[7]改进了 Certcoin 系统,在其基础上增加了隐私保护功能。Matsumoto 等^[8]基于以太坊区块链架构设计了一种新的分布式 PKI 认证体系,该体系能够很好地解决传统 PKI 证书管理、撤销和查询通信量过大的问题。为了解决 CA 失责现象,Chen 等^[9]提出了基于区块链的公开高效证书审核 TLS 连接方案 CertChain,其核心思想是通过可靠性等级对 CA

进行分层。Kubilay 等^[10]在 CertChain 的基础上提出了一种新的基于区块链的带有证书透明度的模型 CertLedger, 该模型增加了抵抗分裂世界攻击的能力。马晓婷等^[11]基于联盟链实现了异构域之间的认证协议和重认证协议, 却未能提高证书吊销查询的安全性和证书查询的效率, 也未能对 CA 权力进行监管。

近年来, 区块链技术在其他领域的认证方案引起了研究人员的关注。王昊等^[12]设计了一种应用于智能交通系统中交叉口场景下的匿名车辆身份认证的方案, 该方案使用区块链记录了基于身份和哈希消息认证码的混合密码认证结果, 能够以较少的通信开销实现匿名车辆认证。张勳等^[13]设计了一种应用于移动自组织网的分布式认证方案, 该方案设计了具有轻量化的认证账本和共识算法的区块链技术, 能够应用于移动节点资源受限的场景。

从上述文献可知, PKI 体系中存在 CA 权力过大、证书吊销查询困难等问题, 同时区块链技术凭借其去中心化、透明度高、结构扁平等优点在认证领域有广阔的研究前景。本文提出了一种基于区块链的高透明度 PKI 认证协议, 主要贡献如下。

1) 提出了一种基于门限签名的高效拜占庭容错共识机制。该机制有 3 点创新: 一是引入门限签名技术, 降低了通信复杂度; 二是在视图切换协议的主节点选举中设计了可信度等级指标用于衡量单个节点的可信度, 增加了可监管性; 三是在快速一致性协议中引入了批处理机制, 提升了共识过程的性能。

2) 提出了一种基于区块链的高透明度的 PKI 认证协议。该协议基于提出的高效共识机制, 引入了区块链技术, 在 X.509 证书格式的基础上设计了新的 BcCert 证书, 增加了证书的透明度, 提升了证书吊销查询的安全性, 并引入计数布隆过滤器提升了证书查询效率。

3) 在提出的 PKI 认证协议中引入了外界监督机制, 在证书的生命周期管理中增加了证书审计流程, 对 CA 的行为做出监管, 促使 CA 提高其安全标准, 一定程度上限制了 CA 的权力。

本文提出的基于区块链的高透明度 PKI 认证协议, 是建立在设计的基于门限签名的高效拜占庭容错共识机制的基础上, 接下来阐述该共识机

制的设计过程。

1 基于门限签名的高效拜占庭容错共识机制

对拜占庭容错 (BFT, Byzantine fault tolerant) 算法的研究起源于 Lamport 等^[14]在 1982 年研究发表的论文 “Byzantine generals and transaction commit protocols”, 最早是用来解决在通信可靠的条件下, 网络环境中若存在恶意节点或者故障节点时如何达成一致共识的问题。

真正使拜占庭容错算法在实际应用中变得可用是在 1999 年, Castro 和 Liskov^[15]提出了实用拜占庭容错 (PBFT, practical Byzantine fault tolerance) 共识算法, 将算法复杂度从指数级别降为多项式级别。

原 PBFT 算法的设计初衷并不是针对区块链设计的, 因此将其直接应用到区块链相关应用中, 主要会存在以下缺点。

1) 三阶段的广播通信非常消耗带宽资源。假设节点总数为 N , 则在不考虑节点失效的情况下, 完成一次三阶段共识过程, 共需要进行传递的消息次数为 $Z(Z = 2N^2 - 2N)$ 。若请求频率较高, 势必带来极大的消息传递成本。

2) 视图切换协议中主节点选举策略过于简单。原 PBFT 共识机制的主节点选取采用轮转 (round robin) 的方式, 即 $p = v \bmod |R|$ 。若恶意攻击者掌握了相关消息, 则容易猜出之后主节点的位置进而发起攻击。

3) 不能实现共识请求的批量处理。原 PBFT 共识机制并不支持灵活的共识请求处理, 只能逐个依次处理共识请求。若请求并发量较大, 势必导致共识过程效率低。

虽然目前有较多针对 PBFT 的改进机制, 但没有彻底解决上述问题^[16-17]。本文主要研究基于区块链的高透明度 PKI 认证协议, 需要在区块链中存储身份认证相关的信息和 CA 审计信息, 便于在认证时查询区块链账本并使用相关信息, 同时根据审计信息对 CA 的权力做出限制。针对上述问题, 本文提出了一种基于门限签名的高效拜占庭容错共识 (TS-PBFT, threshold signature practical byzantine fault tolerant) 机制。

1.1 TS-PBFT 机制的创新点

1) 高扩展性: 能够容纳较大规模数量的节点同时参与共识过程。

2) 可监管性: 能够通过接收外部审计的方式影响主节点选举, 若节点具有不良的审计记录, 在选举时成为主节点的概率将大大降低。

3) 高效性: 在面对大量客户端请求时, 能够尽量及时处理请求, 不造成请求堆积。

本文提出的 TS-PBFT 机制主要在以下几个协议方面进行改进。

1.2 快速一致性协议

本协议确保了单个数据区块在唯一视图下的一致性, 具体流程如下。

(1) 请求阶段

客户端 C 向共识网络 Q 中的主节点 P 发送事务请求 r , 消息格式为 $\langle \text{Request}, o, t, C \rangle$, 其中 t 表示时间戳, o 表示要执行的操作。

(2) 分发阶段

主节点 P 收到请求 $\langle \text{Request}, o, t, C \rangle$ 后分配一个序列号 s , 等待至少 $b \geq \text{batch}$ 个请求之后, 将收到的 b 个请求组装成事务组 R , 其中 $R = (r_1, \dots, r_b)$, v 表示视图号。最后将请求分发消息 $\langle \text{Request-Broadcast}, s, v, R \rangle$ 广播给所有 $3f + 2c + 1$ 个共识域节点。

(3) 提交确认阶段

备份节点 i 收到消息 $\langle \text{Request-Broadcast}, s, v, R \rangle$ 后进行检验, 如果满足下列所有条件就接收:

- ① 备份节点 i 的本地视图号等于 v ;
- ② 序列号 s 在区间 $[ls, ls + \text{win}]$;
- ③ 备份节点 i 之前从未接收过视图号为 v 、序列号为 s 的其他请求分发消息;
- ④ 事务组 R 中的操作合法且经过严格的权限控制检验。

一旦请求分发消息满足上述所有条件, 备份节点 i 首先接收该消息并计算哈希值 $h = H(s, v, R)$, 其中 $H(\cdot)$ 代表哈希函数 (如 SHA256、MD5 等); 接着对哈希值 h 使用子密钥进行签名 $\xi(h)_i$, 其中 $\xi(\cdot)_i$ 是指部分门限签名; 最后将共享签名消息 $\langle \text{Consistency-Share}, s, v, \xi(h)_i \rangle$ 发送给提交消息签名节点 C-Signer。

提交消息签名节点 C-Signer 收到备份节点 i

的消息 $\langle \text{Consistency-Share}, s, v, \xi(h)_i \rangle$ 后进行检验, 如果满足下列所有条件就接收:

- ① C-Signer 的本地视图号等于 v ;
- ② C-Signer 之前从未接收过视图号为 v , 序列号为 s 并且来自备份节点 i 的共享签名消息;
- ③ 验证通过备份节点 i 的部分门限签名消息 $\xi(h)_i$ 。

一旦 C-Signer 收到至少 $3f + c + 1$ 个满足上述所有条件的独立的共享签名消息, 则计算其整体门限签名 $\xi(h)$, 然后将提交确认证明消息 $\langle \text{Consistency-Proof}, s, v, \xi(h) \rangle$ 给所有备份节点。

备份节点 i 收到上述消息后进行检验, 通过之前收到的 $\langle \text{Request-Broadcast}, s, v, R \rangle$ 消息计算 $h = H(s, v, R)$, 然后验证提交确认证明消息内的 $\xi(h)$ 整体门限签名, 若验证通过, 则执行序列号为 s 的事务组 R 。

(4) 执行确认阶段

备份节点 i 执行完成序列号为 s 的事务组 R 中的所有事务后, 其状态从 $\chi_{(s-1)}$ 更新为 χ_s 。接着, 备份节点 i 计算其最新状态 χ_s 的哈希值 $d = H(\chi_s)$, 再对哈希值 d 进行部分门限签名 $\pi(d)_i$, 最后将共享签名消息 $\langle \text{Executed-Share}, s, \pi(d)_i \rangle$ 发送给执行消息签名节点 E-Signer。

执行消息签名节点 E-Signer 收到共享签名消息 $\langle \text{Executed-Share}, s, \pi(d)_i \rangle$ 后检验其部分门限签名, 若通过则记录到本地缓存中。一旦 E-Signer 收到至少 $f + 1$ 个来自不同备份节点的独立共享签名消息时, 则计算其整体门限签名 $\pi(d)$, 然后生成执行确认证明消息 $\langle \text{Executed-Proof}, s, \pi(d) \rangle$ 并将其广播给所有备份节点。备份节点 i 收到后对其签名 $\pi(d)$ 进行验证, 若通过则结束此次共识。

(5) 回复阶段

共识结束后, 执行消息签名节点 E-Signer 向每一个客户端 C 发送一个执行确认消息 $\langle \text{Execute-Done}, o, l, s, \chi_s, \pi(d), \text{Merkle} \rangle$ 。用来告知客户端 C 请求 o 成功执行。其中, χ_s 表示状态机最新状态, val 表示执行完 o 后的返回值, l 表示客户端 C 的请求 o 在事务组 R 内的序号, Merkle 表示序列号 s 内所有请求返回值所构成的默克尔树。

客户端 C 收到执行确认消息后对其签名 $\pi(d)$ 进行检验, 通过后对其做默克尔证明 $\text{verify}(o, l, s, \chi_s, \pi(d), \text{Merkle})$, 若区块链账本状态符合其期望值则确认消息。

若上述所有流程都在客户端定时器超时之前完成, 则共识流程顺利完成, 否则客户端重新发起共识请求。

1.3 线性一致性协议

(1) 触发条件

当提交消息签名节点 C-Signer 未在规定时间内收集到 $3f+c+1$ 条共享签名消息 $\langle \text{Sign-share}, s, v, \xi(h)_i \rangle$ 时, 说明系统中处理速度变慢, 共识性能下降, 这可能是因为非拜占庭错误节点数增多, 也可能是因为网络问题。不管是何种原因, 为了保证共识机制的效率, 系统此时会进入线性模式。在该模式下, 主节点 P 将兼具提交消息签名节点 C-Signer 的功能, 从而简化了系统模式, 提高了共识效率。

线性模式和快速模式的区别主要体现在分发阶段和提交确认阶段, 下面详细描述线性模式下分发阶段和提交确认阶段的实现原理。

(2) 分发阶段

C-Signer (即主节点 P) 在收到 $3f+c+1$ 条共享签名消息后, 生成请求分发消息 $\langle \text{Request-Broadcast-Slow}, s, v, \tau(h) \rangle$ 而非 $\langle \text{Consistency-Proof}, s, v, \xi(h) \rangle$, 然后广播给所有备份节点。注意, 此处的门限签名 $\tau(h)$ 不同于快速模式下的门限签名 $\xi(h)$ 。

(3) 提交确认阶段

备份节点 i 收到消息 $\langle \text{Request-Broadcast-Slow}, s, v, \tau(h) \rangle$ 后进行检验, 如果满足下列所有条件就接收:

- ① 验证通过 $\tau(h)$ 的数字签名;
- ② 备份节点 i 的本地视图号等于 v ;
- ③ 备份节点 i 之前从未在线性模式下, 接收过视图号为 v 、序列号为 s 的其他请求分发消息。

一旦请求分发消息满足上述所有条件, 备份节点 i 生成共享签名消息 $\langle \text{Consistency-Share-Slow}, s, v, \tau_i(\tau(h)) \rangle$ 并将其发送给 C-Signer (即主节点 P)。

当 C-Signer (即主节点 P) 收到 $2f+c+1$ 条共享签名消息后, 计算其整体门限签名 $\tau(\tau(h))$, 最后将提交确认证明消息 $\langle \text{Consistency-Proof-Slow}, s, v, \tau(\tau(h)) \rangle$ 广播给所有备份节点。

备份节点 i 收到上述消息后进行检验, 通过之前收到的 $\langle \text{Request-Broadcast}, s, v, R \rangle$ 消息计算 $h = H(s, v, R)$, 然后验证提交确认证明消息内的 $\tau(\tau(h))$ 门限签名, 若验证通过, 则执行序列号为 s 的事务组 R 。

1.4 检查点协议

在本节的共识机制中, 序列号为 s 的事务组 R 在区块链中可能存在下列 3 种状态。

① 事务提交: 指至少有一个非拜占庭错误节点已经提交事务。

② 事务执行: 指至少有一个非拜占庭错误节点已经执行完 $[1, s]$ 个事务组中的所有事务。

③ 事务稳定: 指至少有 $f+1$ 个非拜占庭错误节点已经执行完 $[1, s]$ 个事务组中的所有事务。

当序列号为 s 的事务组 R 稳定时 (即事务组 R 已经被所有节点执行), 可以使用垃圾回收机制收集所有小于序列号 s 的事务组。具体来说, TS-PBFT 共识机制会定期 (每隔 $\text{win}/2$ 时隙) 执行一次检查点协议来更新最后一个稳定序列号 ls 。

1.5 视图切换协议

(1) 触发条件

当以下任何一种情况发生时, 共识机制进入视图切换阶段:

- ① 视图切换定时器超时;
- ② 来自主节点 P 的消息签名验证失败;
- ③ 任意备份节点 i 收到 $f+1$ 个错误反馈消息。

每个备份节点 i 都会记录最近一个稳定序列号 ls , 之后每次执行完序列号在 $[ls, ls+\text{win}]$ 范围内的事务组后, 都会记录其状态摘要信息 $\chi_{ls}, \chi_{(ls+1)}, \dots, \chi_{(ls+\text{win})}$ 。

(2) 主节点选举

本文设计了可信度等级 (RL, reliability level) 指标来衡量单个节点的可信度, 并以此作为参考选举出主节点。

节点可信度等级指标的迭代计算方式如下。

$$RL_v = RL_{v-1} - \alpha if_{\text{leader}} - \beta \text{num}_{\text{audit}} + \gamma \quad (1)$$

其中, RL_v 表示节点在视图号为 v 下的可信度, 其初始值为固定值, if_{leader} 是在视图 $v-1$ 时是否当选主节点的布尔值, num_{audit} 表示在视图 $v-1$ 时该节点收到的审计报告数, α 、 β 、 γ 是调整系数。根据式(1), 若节点在视图 $v-1$ 时当选了主节点, 则在视图 v 时, 其可信度等级 RL_v 会有所降低, 这是为了避免同一个节点连续多次当选主节点; 若节点在视图 $v-1$ 时收到了不良的审计报表, 则在视图 v 时, 其可信度等级 RL_v 也会降低, 其降低的幅度由审计报表数所定, 这是为了对节点做出监管, 惩罚失责节点。

(3) 发起新视图

备份节点 i 将视图切换消息 $\langle \text{View-Change}, v, ls, \chi_{ls}, \chi_{(ls+1)}, \dots, \chi_{(ls+win)} \rangle$ 发送给新的主节点 P 。

新的主节点 P 收到至少 $2f+2c+1$ 个来自不同备份节点的视图切换消息后, 进入视图变更阶段, 设置新的视图号 $v^* = v+1$, 并且将之前收集到的 $2f+2c+1$ 条视图切换消息组 I 打包在新视图消息 $\langle \text{New-View}, v^*, I \rangle$ 中发送给所有备份节点。

(4) 接收新视图

在该阶段, 需要满足系统从老视图到新视图的安全切换。

备份节点 i 收到新视图消息 $\langle \text{New-View}, v^*, I \rangle$ 后提取视图切换消息组 I , 对其中的 $2f+2c+1$ 条消息按照序列号大小从低到高 $[ls, ls+win]$ 逐一处理。对于其中的每一个序列号为 $\star$ 的消息, 如果在本地缓存中能找到快速模式提交证明消息 $\langle \text{Consistency-Proof}, \star, v, \xi(h) \rangle$ 或者线性模式提交证明消息 $\langle \text{Consistency-Proof-Slow}, \star, v, \tau(\tau(h)) \rangle$ 的记录, 说明备份节点 i 之前已经经过执行其中的事务组, 认可其状态摘要签名。

2 基于区块链的透明 PKI 认证协议

在 TS-PBFT 机制的基础上, 本节设计了基于区块链的可信透明 PK (TB-PKI, transparent-advanced blockchain-based public key infrastructure) 认证协议。本协议的主要创新点如下。

1) 针对区块链的特点, 在 X.509 证书格式的基础上设计了新的 BcCert 证书, 增加了证书的透明度, 提升了证书吊销查询的安全性。

2) 引入计数布隆过滤器, 提升了证书查询

效率。

3) 将审计机制与共识机制相融合, 促使 CA 提高其安全标准, 一定程度上限制了 CA 的权力。

2.1 TB-PKI 认证协议系统模型

本节 TB-PKI 认证协议系统模型中, 参与的实体包括用户 (U, user), 证书申请者 (CR, certification register), 证书颁发机构 (CA, certification authority), 联盟区块链系统 (CBS, consortium blockchain system), 如图 1 所示。

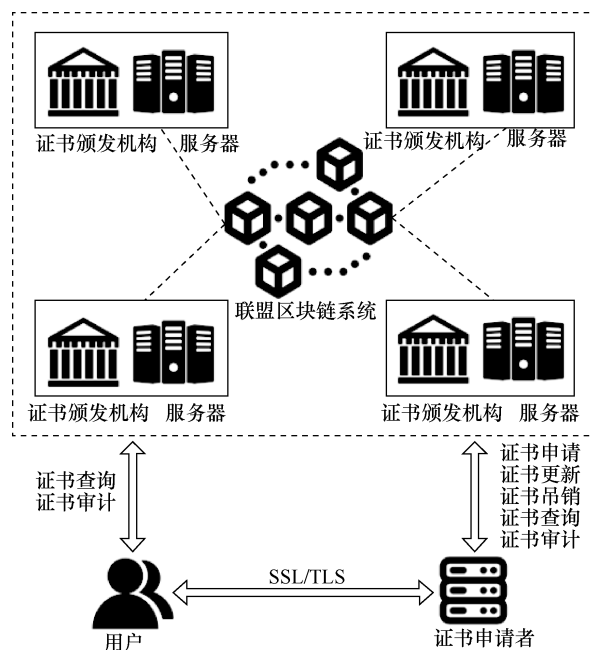


图 1 TB-PKI 认证协议系统架构
Figure 1 Framework of TB-PKI authentication protocol system

2.2 区块链证书

TB-PKI 认证协议设计了一种针对联盟区块链场景的证书, 由区块链各节点达成共识后记入区块链账本中, 该证书可作为授权域不可篡改的可信凭证。传统的 X.509 证书与本文区块链证书的差别如图 2 所示。

本文设计的 BcCert 证书保留了传统 X.509 证书中的版本号、证书序列号、发行商名称、证书主体名、证书主体公钥信息和扩展中大部分主要字段。在此基础上, 新增了与区块链相关的请求类型、请求时间、证书哈希和上一区块高度字段。与此同时, 去除了签名、签名算法和扩展中的 CRL 分发点以及颁发机构信息访问字段。

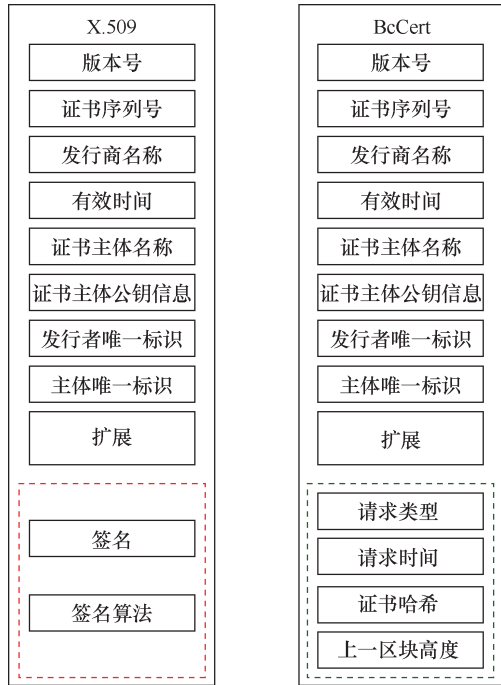


图2 X.509 证书与 BcCert 证书
Figure 2 X.509 certificate and BcCert certificate

新增字段主要用于记录与数字证书相关的信息，操作类型、修改时间、哈希值和上一次区块高度。

删除字段主要包含签名、签名算法、扩展字段中的 CRL 分发点和扩展字段中的颁发机构信息访问。删除这些字段是因为本节模型中，用户对证书的信任源于整个联盟区块链系统而不是单个 CA，同时本节模型不需要额外借助第三方机构即可完成证书状态查询。

2.3 区块结构设计

在 TB-PKI 认证协议中，区块链存储的数据有两种。第一种是证书生命周期记录，第二种是证书审计记录。两者的区块头基本相同，唯一的区别是证书生命周期记录的区块头多了双计数布隆过滤器字段，该字段用于证书的快速查询。两者的区块体也不同，证书生命周期记录的区块体由多个 X.509 证书组成，而证书审计记录的区块体由多个审计报告组成，如图 3 所示。

2.4 生命周期管理

(1) 证书注册

CR 向 CA 发起注册请求 $\text{RegReq} = \{\text{Info}_{\text{CR}}, \text{Reg}\}$ ，其中 Info_{CR} 表示证书申请者的身份信息，Reg 表示请求操作为“注册”。

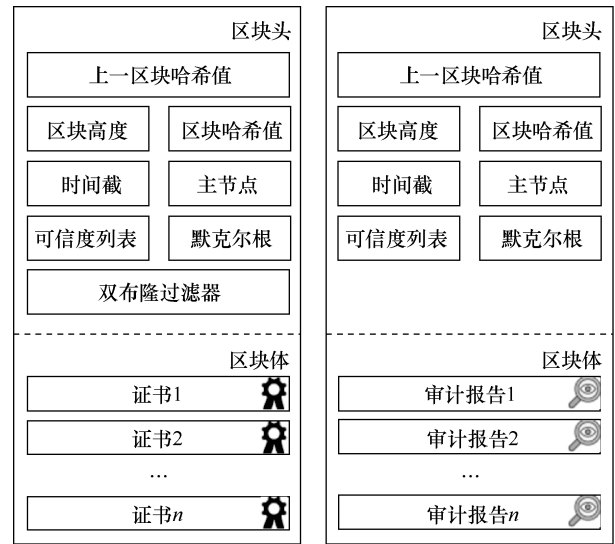


图3 TB-PKI 协议区块数据结构设计
Figure 3 Blockchain structures of TB-PKI protocol

收到请求后，CA 先通过审查机制验证 CR 的身份，若身份属实，则 CA 在本地生成对应的区块链数字证书 $\text{BcCert}_{\text{CR}}$ ，接着将 $\text{BcCert}_{\text{CR}}$ 发送到联盟区块链系统 CBS 中共识存储。

CBS 中的主节点收到请求后，将 $\text{BcCert}_{\text{CR}}$ 放入区块体中，并将其哈希值插入计数布隆过滤器 ACBF 中，等待一定数量的证书请求后，组装区块并写入区块链账本中，共识存储完成后，返回证书在区块链账本中的位置 h 。

CA 将证书 $\text{BcCert}_{\text{CR}}$ 和证书所在区块高度 h 作为结果返回给证书申请者 CR。

CR 在区块链账本上查询 $\text{BcCert}_{\text{CR}}$ 状态，若已生效，则该阶段结束。

(2) 证书更新

CR 向 CA 发起更新请求 $\text{UpdReq} = \{\text{BcCert}_{\text{CR}}, h, \text{Upd}\}$ ，其中 $\text{BcCert}_{\text{CR}}$ 表示证书申请者之前获得的证书， h 表示证书所在高度，Upd 表示请求操作为“更新”。

收到更新请求后，向联盟区块链系统查询 $\text{BcCert}_{\text{CR}}$ 是否存在区块链账本中。

CBS 收到请求后，执行 $\text{BCheck}(\text{Hash}(\text{BcCert}_{\text{CR}}), h)$ ，然后将查询结果返回给 CA。

CA 收到后判断证书是否在区块链账本中，若不存在，则直接拒绝更新操作；若存在证书，则在本地重新生成区块链数字证书 $\text{BcCert}_{\text{CR}}^*$ ，并将其中的“上一次区块高度”字

段填充为更新之前的证书所在区块高度 h ，接着将 $BcCert_{CR}^*$ 发送到联盟区块链系统 CBS 中共识存储。

CBS 中的主节点收到请求后，将 $BcCert_{CR}^*$ 放入区块体中，并将其哈希值插入计数布隆过滤器 ACBF 中，同时在区块链账本找到之前的 $BcCert_{CR}$ ，并将其哈希值插入计数布隆过滤器 RCBF 中，等待一定数量的证书请求后，组装区块并写入区块链账本中，共识存储完成后，返回证书在区块链账本中的位置 h^* 。

CA 将证书 $BcCert_{CR}^*$ 和证书所在区块高度 h^* 作为结果返回给证书申请者 CR。

CR 在区块链账本上查询 $BcCert_{CR}$ 状态，若其有效时间已更新，则该阶段结束。

(3) 证书吊销

CR 向 CA 发起吊销请求 $RevReq=\{BcCert_{CR}, h, Rev\}$ ，其中 Rev 表示请求操作为“吊销”。

收到吊销请求后，CA 向联盟区块链系统 CBS 查询 $BcCert_{CR}$ 是否有效。

CBS 收到请求后，执行 $BCheck(Hash(BcCert_{CR}), h)$ ，然后将查询结果返回给 CA。

CA 收到后判断证书是否在区块链账本中，若已经失效或者找不到记录，则直接拒绝吊销操作；若有效，则将 $BcCert_{CR}$ 和状态修改请求发送给联盟区块链系统 CBS。

CBS 中的主节点收到请求后，将 $BcCert_{CR}$ 的哈希值从计数布隆过滤器 ACBF 中删除，并将其哈希值插入计数布隆过滤器 RCBF 中，等待一定数量的证书请求后，组装区块并写入区块链账本中，共识存储完成后，返回区块链账本中最新区块的位置 h^* 。

CA 将证书最新区块高度 h^* 作为结果返回给证书申请者 CR。

CR 在区块链账本上查询 $BcCert_{CR}$ 状态，若已吊销，则该阶段结束。

(4) 证书验证

U 发起请求，与 CR 建立 SSL/TLS 连接。

CR 在握手协议中提供具有相关区块高度值的证书 $\{BcCert_{CR}, h\}$ 。

U 检查 $BcCert_{CR}$ 的签名有效期，并在本地证

书库中查询颁发证书 CA 的合法性，若上述检查都通过， U 将 $\{Hash(BcCert_{CR}), h\}$ 发送给联盟区块链系统 CBS 查询其完整生命周期记录。

CBS 收到请求后，执行 $BCheck(Hash(BcCert_{CR}), h)$ ，查询结束后，返回验证结果。

U 收到验证结果，若结果显示证书验证通过，则说明证书合法有效，正式建立 SSL/TLS 连接，否则，断开 SSL/TLS 连接。

(5) 证书审计

U 或 CR 若发现 CA_i 签发的证书异常或者出现了其他负面审计问题，可以向任意证书颁发机构 CA 发起审计请求 $AdtReq=\{Audit_{CA_i}, Adt\}$ ，其中 $Audit_{CA_i}$ 表示对 CA_i 的审计反馈， Adt 表示请求操作为“审计”。

CA 审核确认之后将 $Audit_{CA_i}$ 发送到联盟区块链系统 CBS 中共识存储。

CBS 中的主节点收到请求后，将 $Audit_{CA_i}$ 放入区块体中，等待一定数量的审计请求后，组装区块并写入区块链账本中，共识存储完成后，返回证书在区块链账本中的位置 h 。

CA 将审计报告所在区块高度 h 作为结果返回。

U 或者 CR 在区块链账本上查询是否存在 $Audit_{CA_i}$ ，若存在，则该阶段结束。

3 方案分析

3.1 安全性分析

(1) 缓解伪装申请证书

在 TB-PKI 认证协议中，引入外界监督机制。若域名的真正拥有者发现其证书被他人冒名申请，则可以向 CA 发起证书审计请求，并要求其吊销虚假证书。CA 确认后向联盟区块链系统发起请求，将该审计报告写入区块链账本，同时将区块链账本中的虚假证书状态修改为吊销状态，最后通过算法降低颁发该虚假证书 CA 的可信度。

虽然该种攻击无法从根源上避免，但通过该审计反馈机制，影响了失责 CA 在之后共识过程中的话语权，也在区块链账本中留下了不可篡改的记录。总体来说，TB-PKI 认证协议能够促使 CA 提

高其安全性，进而降低发生该类攻击的可能性。

(2) 抵抗恶意修改证书

在 TB-PKI 认证协议中，证书操作并不是由某一个证书签发机构所决定，而是需要整个联盟区块链系统内的成员达成共识才能决定。TB-PKI 认证协议使用的共识机制是 TS-PBFT，节点分布为 $n=3f+2c+1$ ，其拜占庭容错率为 $f>n/3$ ，即除非恶意攻击者能同时入侵 $1/3$ 以上个 CA，才有可能恶意修改证书。而同时入侵 $1/3$ 以上个 CA 近乎不可能。因此，TB-PKI 认证协议更加安全。

(3) 抵抗使用伪造证书

在 TB-PKI 认证协议中，由于区块链具备分布式多节点的特点，即使恶意攻击者使用响应压制的方式，也无法影响用户获取证书吊销状态。同时由于区块链账本上记录了证书的整个颁发过程，可以通过查询最新区块的区块头快速判断证书是否有效。若恶意攻击者在 SSL/TLS 连接中使用了伪造证书，由于这些伪造证书并没有被写入联盟区块链系统中，用户在检查证书状态时可以发现异常，进而拒绝之后的 SSL/TLS 连接。因此，TB-PKI 认证协议在对抗伪造证书的安全性上也更胜一筹。

(4) 抵抗分布式拒绝服务攻击

在 TB-PKI 认证协议中，多个成员组成了联盟组织，可以将 CA 或者某个用户的访问请求转发到联盟平台的其他成员上，从而减轻单个成员的访问负担。

在传统的 PKI 体系中，可信第三方提供的 CRL 服务或者 OCSP 服务往往是单机服务，分布式拒绝服务攻击可以对系统造成严重的安全威胁。而 TB-PKI 认证协议相比传统的 PKI 体系，在分布式拒绝服务攻击下可以提供可持续的服务。

3.2 性能分析

TB-PKI 认证协议的区块结构与比特币区块结构比较如表 1 所示。在安全方面，区块头中存在可信度列表字段，用于记录联盟区块中各个 CA 的可信度，从而可以对 CA 行为进行公平公正的审计，此外，在区块体部分设计了 BcCert 证书结构，使证书的状态和颁发过程可以被完整记录，提供透明的状态查询，具有更高的安全性；在效率方面，区块头中主要增加了可信度列表和双计

数布隆过滤器的字段，区块头结构设计了双计数布隆过滤器，分别用于记录有效证书和记录吊销证书，提高了查询效率。协议中设计的区块结构为了保证安全和效率增加了对应字段，可能比比特币区块需要更多空间用于数据的存储。

表 1 TB-PKI 认证协议的区块结构与比特币区块结构比较
Table 1 Comparison of blockchain structure used in TB-PKI authentication protocol and bitcoin

区块结构	可信度列表	双计数布隆过滤器	BcCert 证书
TB-PKI 协议	具备	具备	具备
比特币	不具备	不具备	不具备

接下来，通过实验对 TB-PKI 协议中证书注册、证书更新和证书吊销的耗时进行测试。同时将 TB-PKI 认证协议的单次完整握手时间和其他基于 PKI 认证协议改进方案进行比较。

实验环境和参数设置如图 4 所示，本次实验基于 Docker 虚拟化技术实现，每个节点都独立运行在规格相同的 Docker 虚拟机中。区块链系统节点为 6 个，不设置故障节点。

软硬件规格	物理机（6 核 12 线程，16 GB 内存，MacOS 操作系统）
	Docker 虚拟机（单核 CPU，1 GB 内存，Linux 操作系统）
使用的算法和库函数	数字签名算法：RSA
	门限签名算法：BLS
	哈希函数：SHA256
	SSL/TLS 库：openssl
	共识算法：TS-PBFT

图 4 实验环境和参数设置

Figure 4 Experiment environment and parameter setting

(1) 证书生命周期耗时

每次实验设置 1 个客户端，以 1 000 次/秒的频率分别申请进行证书注册、证书更新和证书吊销。取 100 次实验的平均值作为最终结果。

证书申请、更新、吊销过程均以申请者发起请求为始，以最终在联盟区块链系统中查询到最新证书状态为止。其中，共识阶段以 CA 发起共识请求为始，以联盟区块链系统生成新的区块为止；查询证书阶段以证书申请者收到

CA 完成共识阶段之后返回的证书和区块位置为始, 以在联盟区块链系统中查询到证书最新状态为止。上述证书申请过程耗时和起止时刻如表 2 所示, 证书更新过程耗时和起止时刻如表 3 所示, 证书吊销过程耗时和起止时刻如表 4 所示, 其中, 时刻指时间刻度。

表 2 证书申请过程耗时和起止时刻

Table 2 Duration of the certificate application stage and corresponding starting and ending time point
单位: ms

阶段	耗时	起始时刻	终止时刻
证书申请	8 103	0	8 103
共识阶段	5 001	44	5 045
查询阶段	90	8 013	8 103

表 3 证书更新过程耗时和起止时刻

Table 3 Duration of the certificate updating stage and corresponding starting and ending time point
单位: ms

阶段	耗时	起始时刻	终止时刻
证书更新	8 379	0	8 379
共识阶段	5 970	163	6 133
查询阶段	99	8 280	8 379

表 4 证书吊销过程耗时和起止时刻

Table 4 Duration of the certificate revocation stage and corresponding starting and ending time point
单位: ms

阶段	耗时	起始时刻	终止时刻
证书吊销	7 562	0	7 562
共识阶段	5 020	103	5 123
查询阶段	23	7 539	7 562

从表 2、表 3 和表 4 可以看出, 在证书的申請、更新和吊销过程中, 共识过程需要的时间远多于查询过程需要的时间。证书更新过程的耗时比证书申请过程的耗时需要的时间稍长, 是因为在证书更新过程中需要多查询一次证书的状态。证书吊销过程中的查询阶段需要的时间较少, 因为此次证书查询只需要直接找到最新的区块即可, 不需要通过区块高度定位区块。

(2) 单次握手时间

在一次完整的 TLS/SSL 握手中, 用户和证书申请者建立连接, 证书申请者提供证书给用户, 用户先在本地验证证书合法性, 然后在联盟区块

链系统中查询证书的最新状态是否有效, 若有效, 则建立连接。

本实验设置 1 个客户端和 1 个服务端, 使其建立 TLS/SSL 连接, 并在连接建立后马上断开并清除缓存, 为了不失一般性, 取 1 000 次实验的平均值作为最终结果。与此同时, 将结果和 CRL、OCSP、CCSP^[18]、ARPKI^[19]、TriPKI^[20]做对比。需要注意的是, 本实验设定每次 TLS/SSL 连接, CRL 都需要重新下载最新的撤销证书列表。

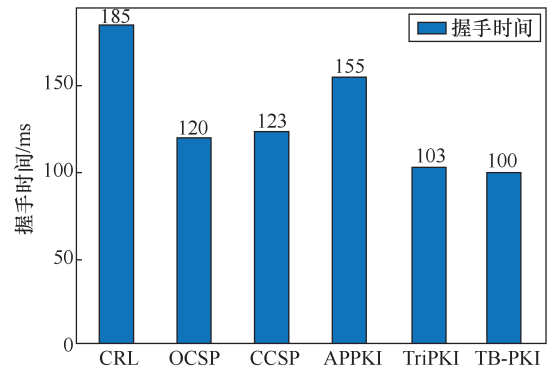


图 5 握手时间性能对比

Figure 5 Comparison of handshake performance

从图 5 可以看出, CRL 方案耗时 185 ms, OCSP 方案耗时 120 ms, CCSP 方案耗时 123 ms, ARPKI 方案耗时 155 ms, TriPKI 认证协议耗时 103 ms, 而本文方案耗时 100 ms。由此可以看出, 本文设计的 TB-PKI 认证协议在 TLS/SSL 握手中相比其他方案具有一定的优势。

4 结束语

本文提出了一种基于门限签名的高效拜占庭容错共识机制, 将通信复杂度降低到 $O(n)$ 级别; 完善了主节点选举方式; 加入了批处理机制, 从而大幅提升了共识过程的性能。与此同时, 本文提出了一种基于区块链的可信透明 PKI 认证协议。该协议不仅增加了证书的透明度, 提升了证书吊销查询的安全性, 而且将审计机制与共识机制相融合, 促使 CA 提高其安全标准, 一定程度上限制了 CA 的权力。与现有证书查询方案相比, 本文方案能够有效提升证书查询效率, 降低握手时间。

参考文献:

- [1] MATHUR S, ARORA A. Internet of things (IoT) and PKI-based

- Security Architecture[M]//Industrial Internet of Things and Cyber-Physical Systems: Transforming the Conventional to Digital. 2020: 25-46.
- [2] SERRANO N, HADAN H, CAMP L J. A complete study of PKI (PKI's Known Incidents)[J]. SSRN Electronic Journal, 2019 (47).
- [3] KAMINSKY D, PATTERSON M L, SASSAMAN L. PKI layer cake: new collision attacks against the global X. 509 infrastructure[C]//International Conference on Financial Cryptography and Data Security. 2010: 289-303.
- [4] PRINS J R, CYBERCRIME B U. DigiNotar certificate authority breach "operation black tulip"[J]. Fox-IT, 2011, (11): 18.
- [5] DAVIDSON S, DE FILIPPI P, POTTS J. Blockchains and the economic institutions of capitalism[J]. Journal of Institutional Economics, 2018, 14(4):639-658.
- [6] FROMKNECHT C, VELICANU D, YAKOUBOV S. Certcoin: a namecoin based decentralized authentication system 6.857 class project[J]. Unpublished Class Project, 2014.
- [7] AXON L, GOLDSMITH M. PB-PKI: a privacy-aware blockchain-based PKI[C]//SECRYPT. 2017: 311-318.
- [8] MATSUMOTO S, REISCHUK R M. IKP: turning a PKI around with decentralized automated incentives[C]//2017 IEEE Symposium on Security and Privacy (SP). 2017: 410-426.
- [9] CHEN J, YAO S, YUAN Q, et al. Certchain: public and efficient certificate audit based on blockchain for tls connections[C]//IEEE INFOCOM 2018-IEEE Conference on Computer Communications. 2018: 2060-2068.
- [10] KUBILAY M Y, KIRAZ M S, MANTAR H A. CertLedger: a new PKI model with certificate transparency based on blockchain[J]. Computers & Security, 2019, 85: 333-352.
- [11] 马晓婷, 马文平, 刘小雪. 基于区块链技术的跨域认证方案[J]. 电子学报, 2018, 46(11): 2571-2579.
- MA X T, MA W P, LIU X X. A cross domain authentication scheme based on blockchain technology[J]. Acta Electronica Sinica, 2018, 46(11):2571-2579.
- [12] 王昊, 吴天昊, 朱孔林, 等. 交叉场景下基于区块链技术的匿名车辆身份认证方案[J]. 网络与信息安全学报, 2020, 6(5): 27-35.
- WANG H, WU T H, ZHU K L, et al. Anonymous vehicle authentication scheme based on blockchain technology in the intersection scenario[J]. Chinese Journal of Network and Information Security, 2020, 6(5): 27-35.
- [13] 张勛, 马欣. 基于区块链的轻量化移动自组网认证方案[J]. 网络与信息安全学报, 2020, 6(4): 14-22.
- ZHANG X, MA X. Lightweight mobile Ad Hoc network authentication scheme based on blockchain[J]. Chinese Journal of Network and Information Security, 2020, 6(4): 14-22.
- [14] LAMPORT L, FISCHER M. Byzantine generals and transaction commit protocols[R]. Technical Report 6. 1982.
- [15] CASTRO M, LISKOV B. Practical byzantine fault tolerance[C]//OSDI. 1999: 173-186.
- [16] HAO X, YU L, ZHIQIANG L, et al. Dynamic practical byzantine fault tolerance[C]//2018 IEEE Conference on Communications and Network Security (CNS). 2018: 1-8.
- [17] LI F, LIU K, LIU J, et al. DHBFT: Dynamic hierarchical byzantine fault-tolerant consensus mechanism based on credit[C]//Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data. 2020: 3-17.
- [18] CHARITON A A, DEGLERI E, PAPADOPOULOS P, et al. CCSP: a compressed certificate status protocol[C]//IEEE INFOCOM 2017-IEEE Conference on Computer Communications. 2017: 1-9.
- [19] BASIN D, CREMERS C, KIM T H J, et al. ARPki: Attack resilient public-key infrastructure[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 382-393.
- [20] CHEN J, YAO S, YUAN Q, et al. Checks and balances: a tripartite public key infrastructure for secure Web-based connections[C]//IEEE INFOCOM 2017-IEEE Conference on Computer Communications. 2017: 1-9.

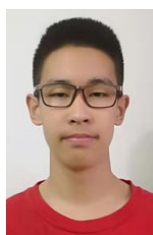
[作者简介]



陈立全 (1976-), 男, 广西玉林人, 东南大学教授、博士生导师, 主要研究方向为密码与安全协议、区块链技术。



李潇 (1996-), 女, 江苏镇江人, 东南大学博士生, 主要研究方向为物联网安全、区块链。



杨哲懿 (2005-), 男, 北京人, 主要研究方向为区块链、物联网安全。



钱思杰 (1995-), 男, 浙江绍兴人, 东南大学硕士生, 主要研究方向为区块链技术、数据安全技术。