2 基于区块链的微认证系统架构设计

2.1 设计目标

本研究将提出一个基于区块链的微认证系统架构, 架构中包含了可重用的微认证相关服务. 目标架构的需求如下:

(1) 微认证提供者在日常工作中可记录微认证对象的信息;

(2) 微认证提供者根据记录信息生成微证明;

(3) 向微认证验证者提供便捷、有效的微证明验证方式;

(4) 系统架构应保证数据隐私性、完整性;

(5) 确保系统底层架构的可用性.

2.2 架构设计

如图 3 所示, 系统架构采用 3 层结构: 用户界面层、服务层、与基础设施层. 其中, 服务层包括: 部署服务、账户管理服务、数据管理服务、与微认证管理服务; 基础设施层包括: 传统数据库和区块链网络 (及运行在区块链上的智能合约). 系统的目标用户包括: 微认证提供者 (如, 学校)、微认证对象 (如, 学生)、与微认证验证者.

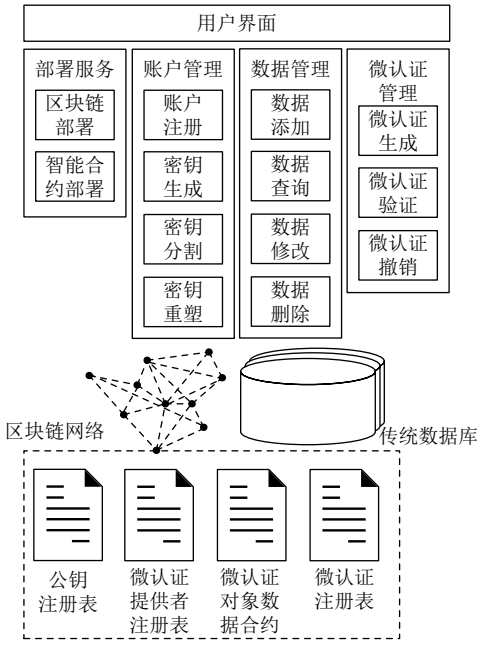

图 3 基于区块链的微认证系统架构示意图

2.2.1 服务层设计

部署服务包含了区块链部署与智能合约部署. 本研究中所采用的区块链平台为以太坊, 用户可通过输入参数 (如, 区块生成难度, 个人 IP 地址等) 自定义私有或联盟区块链网络进行部署. 当区块链部署完毕后, 用户即可以监控、查看区块链运行情况, 并部署所开发的智能合约.

账户管理服务包括账户注册服务、密钥生成服务、与密钥分割/重塑服务. 在本研究中, 区块链上的账户取代了传统数据库中所记录的账户密码. 用户可通过账户注册服务, 设置密码, 由底层区块链生成相对应的账户. 同时, 用户可选择生成非对称密钥, 以便于对数据进行加解密操作. 所生成的公钥将存储于智能合约中, 与用户的区块链账户对应, 便于其他用户进行查询. 通过密钥分割与重塑服务, 用户可将私钥 (或密码) 进行分割, 在丢失、遗忘密钥或密码后, 通过分割碎片进行重塑, 便能够重新控制相关的加解密公钥或者区块链账户.

通过数据管理服务, 微认证提供者可在日常工作中管理微认证对象的信息 (如, 学生的课程成绩), 并同时将其存储在链下数据库与链上智能合约中.

微认证管理服务包括微认证生成服务、微认证验证服务、与微认证撤销服务. 微认证提供者根据数据库中所记录的数据, 生成 JSON 格式的电子微认证. 微认证中包括: 提供者与认证对象的区块链账号、微认证识别号、生成日期、与证明内容. 在微认证验证阶段, 验证者可使用微认证验证服务以校验一个微认证的真伪性. 通过系统提供的在线验证通道, 验证者可输入所收到的微认证, 系统后台将自动识别微认证识别号与提供者区块链账号以进行验证, 并返回验证结果. 若微认证对象不再需要此电子证明, 或提供者发现微认证对象不符合证明内容, 提供者可撤销该微认证, 则后续不再支持验证者对该微认证进行验证. 需要注意, 微认证可通过各种方式 (如邮件) 在 3 种用户之间进行传输, 不包含在本研究范围内.

2.2.2 传统数据库设计

本研究采用传统数据库存储 3 类数据. 第 1 类数据是所部署的智能合约信息, 在进行调用时需要合约地址与二进制接口, 因此将此类智能合约相关信息存储以便于查看和管理. 第 2 类数据是智能合约模板, 本研究将设计、总结适用于智能合约的设计模式, 并应用在微认证系统中, 将其开发为合约模板, 把源码存储于数据库中, 方便用户在部署智能合约时能够进行选择与应用. 第 3 类数据是微认证对象数据, 将微认证对

象学习或考核过程中的原始数据记录下来存储于数据库中, 用于后续生成微认证.

2.2.3 区块链与智能合约设计

区块链为微认证系统提供了底层去中心化基础设施与智能合约运行环境, 微认证提供者与对象通过区块链账户在系统中进行业务流程交互. 本研究中, 区块链上主要应部署 4 类智能合约.

(1) 公钥注册表合约, 用户可将区块链账户与公钥作为键值对上传至注册表中, 在微认证验证过程中, 系统即通过微认证提供者的区块链账户查询到对应公钥进行验证序列的解密.

(2) 微认证提供者注册表, 在微认证系统中, 将能够开具微认证的工作人员的区块链账户存储在注册表中, 任意用户可输入一个区块链账户以查询其是否为有效工作人员. 只有有效工作人员才能向微认证对象生成能够进行有效验证的微认证.

(3) 微认证对象数据合约, 用于记录微认证对象的数据 (加密或哈希形式). 微认证对象数据合约继承了访问控制合约模板以应用相关机制, 规定只有该合约数据所覆盖的微认证对象可查询数据以进行数据完整性校验.

(4) 微认证注册表, 用于存储微认证验证信息. 微认证注册表也应用了访问控制机制, 即只有微认证提供者注册表中的区块链账户可以上传微认证验证信息, 以防止认证作假欺诈.

2.3 核心技术研究

本研究中涉及了区块链部署、区块链账户管理、链上链下数据存储、智能合约设计、与加解密技术方面的知识, 并将在本节中进行进一步的分析与探讨.

在进行区块链部署时, 系统后台脚本通过用户所输入的参数实现自动化部署与配置区块链客户端的任务. 智能合约的部署支持用户选择上传智能合约文件, 系统后台将自动编译并部署智能合约. 部署成功后, 该智能合约的链上地址与应用系统二进制接口将存储于数据库中, 以方便调用.

在链上智能合约设计的过程中, 确定合约中方法的权限控制是一个难点. 在智能合约进行编译与部署前, 以存储在数据库中的访问控制合约模板作为父类, 使目标合约继承该父类合约, 以获得访问控制机制. 图 4 展示了访问控制合约主要代码. 合约所有者默认为微认证提供者区块链账号, 在合约部署时被自动记录; 同时, 部署合约时即可定义能够访问合约的区块链账号[14]. 合约所有者能够更换访问控制的区块链账号以提高可扩展性, 如只有特定微认证对象能够访问存储了自身数据的合约进行数据完整性校验. “modifier”部分用于检测当前访问合约的是否为所定义的区块链账户, 若是, “_;”部分则为运行子类合约目标访问控制的代码.

```
contract AccessControl{
    address [] access;
    address owner;
    function AccessControl(address [] temAccess){
        owner = msg.sender;
        access = temAccess;
    }
    function changeAccess(address [] temAccess){
        if(msg.sender == owner){
            access = temAccess;
        }
    }
    modifier accessed(){
        for(uint i = 0; i < access.length; i++){
            if(msg.sender == access[i]){
                _;
                break;}}
    }
}
```

图 4 访问控制合约主要代码

由于目前区块链平台中并不支持找回或更改密码操作, 本研究出于去中心化的安全考量, 系统架构中也不支持将账户与密码存储于数据库中. 用户可通过分割与重塑区块链密码或加密密钥, 决定分割数量与重塑的阈值, 将所输入的密码/密钥分割成不同的部分, 并自行选择保存方式. 若用户忘记密码/密钥, 则可输入大于或等于阈值数量的密码碎片, 系统后台将通过所输入的碎片重塑出完整的密码/密钥[18].

由于链上数据的透明性与冗余性, 链上链下数据需要进行差异化处理[19]. 本研究中采用将大型原始数据存储在数据库中, 并将原始数据进行区分, 每一部分通过加密, 或者进行哈希计算, 将密文或哈希值存储于链上. 用户可通过对链上数据进行解密查看数据, 或者将链下原始数据与链上哈希值进行对比, 确保数据的完整性.

目标系统架构应向验证者提供快速、高效、可靠的微认证验证服务, 避免过于繁复的验证手段. 在本架构设计中, 微认证生成时, 相关验证信息将存储于链上智能合约中, 包括: 提供者与证明对象的区块链账号、

微认证识别号、与微认证验证序列. 微认证验证序列是通过将微认证进行哈希计算后, 再由提供者私钥进行加密生成的. 微认证的验证过程如图 5 所示, 系统通过智能合约获取提供者公钥与链上微认证验证序列. 将微认证验证序列进行解密后所得的哈希值, 与用户所输入的微认证进行哈希计算后相对比, 进行双重验证, 已确保该微认证的有效性、权威性、与完整性.

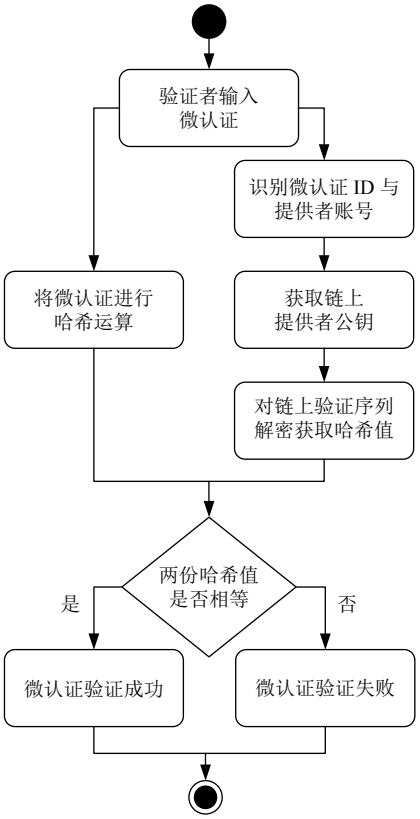

图 5 微认证验证流程示意图

2.4 传统微认证系统设计对比分析

与传统微认证系统架构设计相比, 本文方法主要在 3 个方面进行了设计创新.

(1) 区块链账户取代传统账户功能

传统微认证系统将用户的账户与密码存储于数据库中, 在登录等操作时将当前用户身份与数据库内记录进行对比. 本文方法以链上账户取代传统账户, 系统不存储用户密钥或密码, 避免了黑客攻击窃取用户账户资料的风险. 同时, 链上账户与智能合约可进行交互, 实现合约内的权限管理机制.

(2) 链上链下数据差异化处理

传统微认证系统架构只使用了数据库进行数据存储, 易遭受攻击, 形成“单点故障”. 本文方法将原始数据存于链下数据库, 哈希或加密后的数据存于链上, 通过链上链下数据对比, 确保了数据的一致性. 同时, 区块链的透明性与不可篡改性能够对微认证提供者的操作进行记录并回溯, 对系统内部人员的行为进行约束.

(3) 多节点分布式架构

底层区块链生态为本文方法提供了多节点的分布式架构. 相比于传统微认证系统部署运行在单一服务器上, 本文方法中涉及的所有目标用户均可选择在本地部署区块链节点, 提高了系统的可扩展性.

3 系统架构评估

3.1 系统原型开发

本研究中基于区块链的微认证系统原型使用 JavaScript 语言进行开发, 通过 Tomcat 进行部署. 为实现区块链部署服务, 所需要的部署文件通过安全外壳协议 (SSH) 进行传输, 包括部署脚本文件与创始区块文件. 系统原型采用 MySQL 作为传统数据库, 以太坊作为底层区块链网络. 链上智能合约通过 Solidity 语言进行开发, 并由 Web3 应用程序接口进行部署.

3.2 性能测试

本研究将系统原型部署在阿里云服务器上 (2 vCPUs, 8 GB RAM, 20 GB Disk) 进行性能测试. 实验第一部分为分别运行微认证生成服务与验证服务各一个小时, 记录了运行时间内微认证服务吞吐量的变化. 实验第二部分为分别运行 1000 次微认证生成服务与验证服务, 记录了运行时间总和.

图 6 展示了微认证生成服务与验证服务的吞吐量变化. 在实验过程中, 微认证生成服务吞吐量稳定地保持在 10 tps (每秒事务数) 左右, 而微认证验证服务吞吐量有些许波动, 大部分时间保持在 200 tps 左右, 最高不超过 250 tps. 图 7 与图 8 分别展示了运行 1000 次微认证生成服务与微认证验证服务所需总时间. 生成 1000 个微认证超过 19 000 s, 包含了生成时间以及将相关验证信息上传至区块链的时间.

由于区块链中每个区块所能包含的事务数量有限, 同时, 以太坊中的“工作量证明”共识机制 (POW)[20] 导致了其存在区块间隔 (将近 12 s 生成一个区块), 因此上传微认证信息需要耗费一定时间. 相较之下, 微认证的验证则有更快的速度, 验证 1000 个微认证共需不

到 100 s. 虽然微认证验证服务包括从智能合约中获取信息, 但由于其不改变链上数据的状态, 此类查询操作并不需要通过共识机制被包含进区块中, 能够运行得更加有效率. 因此, 微认证验证服务的吞吐量比生成服务高出许多.

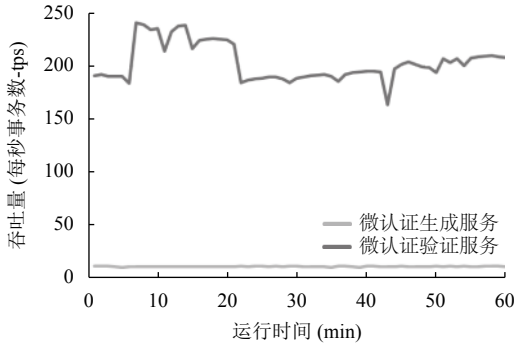

图 6 微认证服务吞吐量变化

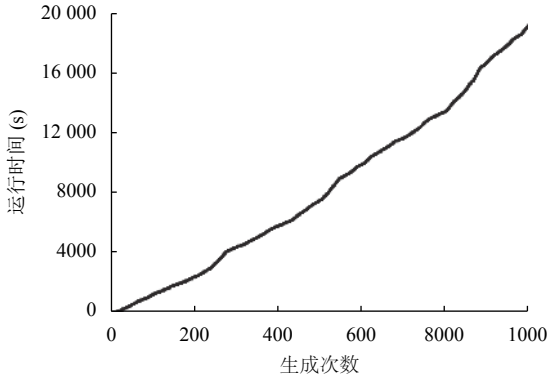

图 7 微认证生成服务运行时间

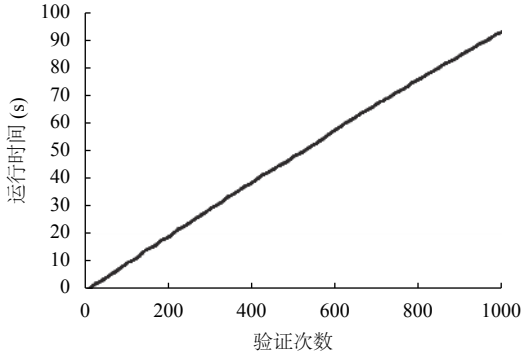

图 8 微认证验证服务运行时间

实验数据表明, 基于区块链的微认证系统基本达到了预期性能. 为进一步提高性能表现和可伸缩性, 在后续研究中将使用其他区块链平台 (如, Parity、Hyperledger Fabric 等) 进行系统开发. 同时, 未来研究也将尝试汇集多个微认证一并哈希打包在一条事务内进行上传[21], 通过减少链上事务数量的方法以达到更高效的性能表现.

3.3 安全性评估

本文系统架构拟达到的隐私安全保护为: 系统架构应保证数据隐私性、完整性; 同时系统底层架构的可用性应得到保障.

通过将区块链引入微认证系统, 改善了现有的中心化架构, 同时提高了微认证的安全性. 微认证对象可自主选择目标微认证验证者 (如毕业生应聘时可将微认证附于简历中), 即未授权者不能够获取证明, 从而防止微认证对象身份信息的泄露. 同时, 若微认证有被不恰当使用的现象发生, 微认证对象可向提供者提出申请, 撤销该证明并进行重新生成与颁发.

本研究中通过将原始数据存储于链下数据库, 加密或哈希数据存储于链上的方式, 微认证验证过程中也使用了数据解密与哈希对比两个步骤, 提供了对数据一致性的保护. 区块链网络需要运行在多个主机节点之间, 每个节点都存有完整的区块副本. 因此, 若出现攻击者对其中的节点进行攻击, 试图篡改数据, 系统也能够很快地通过其余节点进行恢复. 同时, 区块链将记录微认证对象数据与微认证验证信息的上传, 因此, 任何违规的行为 (如, 制作虚假微认证) 都可以追溯到具体的区块链账户, 并进行相应处罚.

4 结论与展望

本文介绍了一种基于区块链的微认证系统架构, 为现有的微认证系统提供了一种改进方案, 并分析了引入区块链技术后系统的性能与安全性. 试验表明, 由于区块链自身性能限制, 上传数据至区块链的过程需要耗费一定时间, 但验证微认证的过程具有很高的效率. 同时, 微认证对象能够更好地掌握微认证的使用情况, 通过区块链数据的透明性与不可篡改性, 微认证欺诈现象也能够得到有效缓解. 本文系统架构只是初步研究, 还有很大的改进空间, 下一步的工作是优化系统架构, 并进一步研究基于区块链的自主身份管理领域.

参考文献

1 Walport M. Distributed ledger technology: Beyond block chain. UK Government Office for Science. https://www.gov.uk/government/news/distributed-ledger-technology-beyond-block-chain. (2016-01-19).

2 Staples M, Chen SP, Falamaki S, *et al*. Risks and opportunities for systems using blockchain and smart contracts [Technical report]. Sydney, Australia: CSIRO, 2017.
3 Swan M. Blockchain: Blueprint for a New Economy. Sebastopol, CA, USA: O'Reilly Media, Inc, 2015.
4 押男, 徐盟盟. 微认证: 非正式学习成果的认定方式. 高等继续教育学报, 2018, 31(5): 17–21, 75. [doi: 10.3969/j.issn.1006-7353.2018.05.004]
5 魏非, 闫寒冰, 祝智庭. 基于微认证的教师信息技术应用能力发展生态系统构建研究. 电化教育研究, 2017, (12): 92–98.
6 Acree L. Seven lessons learned from implementing micro-credentials [Technical report]. Raleigh, NC: Friday Institute for Educational Innovation at the NC State University College of Education, 2016.
7 Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Consulted, 2008. https://bitcoin.org/en/bitcoin-paper.
8 Tschorsch F, Scheuermann B. Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys & Tutorials, 2016, 18(3): 2084–2123. [doi: 10.1109/COMST.2016.2535718]
9 Omohundro S. Cryptocurrencies, smart contracts, and artificial intelligence. AI Matters, 2014, 1(2): 19–21. [doi: 10.1145/2685328.2685334]
10 Alharby M, Van Moorsel A. Blockchain-based smart contracts: A systematic mapping study. arXiv preprint arXiv: 1710.06372, 2017.
11 Eberhardt J, Tai S. On or off the blockchain? Insights on off-chaining computation and data. Proceedings of the 6th European Conference on Service-Oriented and Cloud Computing. Oslo, Norway. 2017. 3–15.
12 Bartoletti M, Pompianu L. An empirical analysis of smart contracts: Platforms, applications, and design patterns. Proceedings of 2017 International Conference on Financial Cryptography and Data Security. Sliema, Malta. 2017. 494–509.
13 Zhang P, White J, Schmidt D C, *et al*. Applying software patterns to address interoperability in blockchain-based healthcare apps. arXiv preprint arXiv: 1706.03700, 2017.
14 Liu Y, Lu QH, Xu XW, *et al*. Applying design patterns in smart contracts. Proceedings of the 1st International Conference on Blockchain. Seattle, WA, USA. 2018. 92–106.
15 Tian F. An agri-food supply chain traceability system for China based on RFID & blockchain technology. Proceedings of 2016 13th International Conference on Service Systems and Service Management. Kunming, China. 2016. 1–6.
16 Liang XP, Zhao J, Shetty S, *et al*. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. Proceedings of 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications. Montreal, QC, Canada. 2017. 1–5.
17 Reyna A, Martín C, Chen J, *et al*. On blockchain and its integration with IoT. Challenges and opportunities. Future Generation Computer Systems, 2018, 88: 173–190. [doi: 10.1016/j.future.2018.05.046]
18 Allen C, Brock A, Buterin V, *et al*. Decentralized public key infrastructure. A White Paper from Rebooting the Web of Trust, 2015.
19 Zyskind G, Nathan O, Pentland A. Decentralizing privacy: Using blockchain to protect personal data. Proceedings of 2015 IEEE Security and Privacy Workshops. San Jose, CA, USA. 2015. 180–184.
20 Zheng ZB, Xie SA, Dai HN, *et al*. An overview of blockchain technology: Architecture, consensus, and future trends. Proceedings of 2017 IEEE International Congress on Big Data. Honolulu, HI, USA. 2017. 557–564.
21 Liang XP, Zhao J, Shetty S, *et al*. Towards data assurance and resilience in IoT using blockchain. Proceedings of MILCOM 2017–2017 IEEE Military Communications Conference. Baltimore, MD, USA. 2017. 261–266.