计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(9):243−248 [doi: 10.15888/j.cnki.csa.006509]

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

去中心化的 SDN 接入身份认证研究①

秦　华[1], 刘　乐[2]

[1](北京工业大学 网络中心, 北京 100124)
[2](北京工业大学 信息学部, 北京 100124)
通讯作者: 刘　乐, E-mail: liule@emails.bjut.edu.cn

摘　要: 本文研究去中心化的网络接入身份认证问题. 依据非交互式零知识证明原理, 借鉴区块链技术中的共识思想, 对传统的拜占庭共识算法进行改进, 提出了利用已经接入网络的主机对申请接入网络者的公钥所属权进行认证, 并对认证结果达成共识的方案, 在此基础上在 SDN 网络中设计并实现了接入身份认证方案 BchainNAC.

关键词: 区块链技术; 去中心化; SDN; 接入认证; 拜占庭容错

引用格式: 秦华,刘乐.去中心化的 SDN 接入身份认证研究.计算机系统应用,2018,27(9):243–248. http://www.c-s-a.org.cn/1003-3254/6509.html

Study on Decentralized SDN Access Authentication

QIN Hua[1], LIU Le[2]

[1](Network Center, Beijing University of Technology, Beijing 100124, China)
[2](Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

Abstract: In this study, the decentralized network access authentication was focused on. Based on non-interactive zero-knowledge proof and technologies of blockchain, we improved practical Byzantine fault tolerance and designed a scheme that the hosts which have been connected to the network verify the host applying for access, by certificating the ownership of the public key. According to the scheme, BchainNAC is designed and implemented in the SDN network.

Key words: technology of blockchain; decentration; SDN; access authentication; Byzantine fault tolerance

随着移动互联网和 SDN 技术的普遍使用, 用户跨域接入网络的需求增长很快, 网络接入身份认证面临了巨大的挑战. 传统的接入认证方式依赖于认证服务器, 可以很好地支持用户在 SDN 域内移动, FlowNAC[1]与 SDFAC[2]均使用扩展后的 IEEE802.1x 协议, 由可信第三方设备根据用户提交的身份信息对接入者进行认证. 文献[3]在 SDN 网络的控制器中设计安全内核 SEK, 并加入了身份认证功能. 文献[4]则给出基于 OpenFlow 的身份认证控制器 SNAC.

跨域用户身份认证不仅依赖于两个域的认证服务器, 还依赖于两个域之间事先建立的信任关系. Eduroam[5]在所有相关的域中建立了一套域间信任机制, 实现用户使用原域的合法账号, 就能在全球已加入 Eduroam 联盟的其它域登录并访问网络. 用户的信息通过用户当前所在域的认证服务器转发给上一级认证服务器, 再转发到用户所在域的服务器, 最后再将应答发送给接入域. 文献[6]介绍并讨论了将 Eduroam 部署到 SDN 中的相关问题. 文献[7]提出分布式跨域认证系统. 在该系统中, 用户在经过各自域的认证服务器认证后, 由安全通信模块完成跨域的认证信息交互.

无论是本域接入还是跨域接入, 上述接入认证方案均由专门的认证服务器实现对用户身份的认证. 但是, 若既要满足用户认证上网, 又要方便用户随时随地接入网络, 在认证服务器之间事先建立认证信任关系

① 收稿时间: 2018-01-08; 修改时间: 2018-01-31; 采用时间: 2018-02-07; csa 在线出版时间: 2018-08-16

存在一定的技术复杂性和可行性, 无法很好地支持用户灵活跨域上网的认证需求, 因此需要研究新的网络接入认证方法.

本文研究去中心化的网络接入身份认证, 要求接入身份认证方法既不依赖于认证服务器, 同时又不用事先建立域间认证信任关系. 借鉴区块链技术[8–11], 本文提出了在无需相互信任的主机间设计共识机制实现身份认证, 同时记录认证结果支持认证结果可追溯的方案. 论文在 SDN 网络中设计并实现了去中心化的接入身份认证方案 BchainNAC.

1 问题描述

定义 1. 集合AH(Access Host) 为已成功接入网络的主机集合. AH集合内节点总数为n, 每一个节点按照接入网络的先后顺序设置编号, 其中任意一个节点的编号为$i \in \{1,n\}$, 节点记为AH_i. 本文中AH集合内所有节点组成验证者.

定义 2. 集合EH(Entry Host) 为申请接入网络的主机集合, 其中任意一个节点记为EH_j, $j \in Z$. 本文中EH_j被看作证明者.

本文研究的主要问题是利用AH中的节点对EH_j进行身份验证. 主要研究内容如下:

(1) 在域和域之间无法事先建立信任关系或者是在接入域中无法事先为用户建立帐号等情况下, 利用接入申请者公开的身份信息进行身份验证, 本文利用接入申请者的公钥信息来进行身份认证.

(2) 为了不依赖于认证服务器解决跨域用户接入身份认证, 本文借鉴区块链技术的去中心化思想, 利用已经接入网络的主机对申请者的身份分别进行认证并达成共识.

(3) 为实现认证结果的可追溯性, 研究认证结果的存储方法. 本文利用区块链存储技术, 对认证结果进行存储, 支持认证结果可追溯.

2 基于非交互的零知识身份认证证明

假设申请接入网络的主机EH_j已生成私钥k, 并使用椭圆曲线算法计算得到公钥K, 且拥有包含公钥信息的数字证书C. 根据非交互式零知识证明[12]原理, AH以非交互式的方式验证EH_j身份的可信性. EH_j只需向AH集合提交包含自身公钥信息的证明m, 由AH集合内的AH_i验证m中认证凭证的所有权以及m中的公钥K与私钥k的对应关系.

m是EH_j提交的身份信息, 其取值是由 Elliptic Curve Cryptography (椭圆加密算法, 简称 ECC)[13]对EH_j的数字证书C进行数字签名生成. 为避免传输过程中产生错误, EH_j除了向AH提交证明m以外, 还需提交一个辅助证明参数A. A是由数字和字母组成的字符串, 是由$Base58check$[14]对公钥K与对公钥K进行 4 次哈希运算后生成的字符串的前 4 位进行连接操作所生成的字符串进行编码而成的. A的取值如下列公式 (1) 所示:

$$A= \text{Base58check}(\text{concat}(\text{left}(\text{SHA256}(\text{SHA256}(\text{RIPEMD160}(\text{SHA256}(K))))))) \quad (1)$$

其中, concat 函数[15]完成字符串的连接操作; left 函数[15]完成取字符串的前 4 位操作; SHA256位哈希函数; RIOPEMD160是信息摘要算法, 输出定长为 20 字节的值.

AH内的节点根据EH_j发送的m、A, 通过下面三项验证内容来证明EH_j的身份可信, 即:

(1) 验证申请信息的数据格式是否符合系统规则.

系统规则包括数据结构正确、输入输出表不为空、申请信息的大小不能小于最小字节等.

(2) 验证m是否有效.

EH_j的公钥K对数字签名解密, 并与原文的摘要进行比对验证.

(3) 验证A是否有效.

AH内的节点计算原始数据的校验码并与A作比对.

3 身份认证共识

AH集合内的节点分别通过上述方法验证EH_j的身份后, 还需要使用共识机制对验证结果达成共识, 集合AH才能对EH_j的身份做出最终的判断.

区块链基础架构的共识层中封装许多共识算法[16–19]. 本文改进实用拜占庭容错算法[20,21], 设计了AH集合对EH_j的身份达成共识的过程. 达成共识的过程包括共识节点对EH_j身份达成共识、共识节点存储共识结果.

为保证AH内节点对EH_j的接入申请达成共识, 需要所有参与达成共识的主机有共同的初始状态, 为实现共识结果可追溯, 需要存储每一次共识的结果, 因此, 借鉴区块链架构中共识层和数据层技术, 本文设计“区块+链”的结构来存储认证结果. 这种存储结构使认证结果具有如下两个特点: 第一, 每一个区块记录了上一个接入申请完成到该区块生成之前的接入结果, 保证

了数据的完整性, 实现认证结果的可追溯. 第二, 一旦新的区块生成并被加入到存储结果的最后一个位置后, 区块的接入认证记录再也不会改变或者删除, 保证了数据库的严谨性, 实现认证结果集体维护.

3.1 区块设计

区块是记录共识状态的数据单元, 由块头和块身组成. 区块块头中主要包括前一区块散列值、下一区块散列值、时间戳[22]、共识结果、区块高度, 各参数的主要作用分别为:

前一区块散列值、下一区块散列值: 实现新生区块与前一区块、下一区块的链接, 以便该区块与前后区块形成先后顺序, 在区块间形成链状结构.

时间戳: 记录当前区块的写入时间, 为数据增加时间维度, 使其具有极强的可验证性和可追溯性.

共识结果: 记录本次共识的结果.

区块高度: 记录当前已存储的区块链中区块的个数, 用h来表示, $h \geq 0$, 指网络中已成功进行接入身份认证的主机数量.

3.2 共识机制

针对每一台申请加入网络的主机EH_j, 集合AH中参与共识的节AH_i, 均需要对EH_j的接入申请进行验证, 并使用共识机制对验证结果达成共识.

集合AH内的节点都需要参与对EH_j的身份验证、对验证结果达成共识的过程, 然而集合AH内的节点有可能无法参与并提交认证信息. 因此, 集合AH是一个拜占庭系统[23], 本文设计的共识机制需要具有一定的容错能力, 保证共识机制的安全性和可用性. 针对实用拜占庭容错算法的应用场景与本文研究的AH集合的拜占庭系统存在的不同, 本文从以下两个方面改进拜占庭容错算法, 解决网络中接入身份认证的共识问题.

(1) 实用拜占庭容错算法中的节点既可以成为发起共识的申请者, 也可以成为达成共识的参与者, 而本文研究的网络中集合AH所在的拜占庭系统仅包含参与共识过程的验证者. 因此, 本文中的共识算法需为每一次共识过程选取一个主节点, AH内的其他节点为副本节点. 主节点负责发起一次共识, 监控AH对本次共识的共识结果, 并通知EH_j及AH中的其他节点共识结果.

(2) 实用拜占庭容错算法只需要对申请者的请求达成共识, 而本文中的共识算法除了需要对申请者的请求达成共识之外, 为实现共识结果可追溯, 存储共识结果还需要对生成认证结果存储区块达成共识. 因此, 本文中的共识算法需要分别对EH_j身份的认证结果、同意生成认证结果区块的结果达成共识.

3.3 改进的拜占庭容错算法

假设网络初始化后已经存在一定数量的主机. 依据拜占庭容错算法可知, 只有参与共识的错误节点数f不超过$\frac{n-1}{3}$才能达到系统正常运行的安全性和可用性需要.

定义 3. 视图是一次共识从开始到结束所使用的全部数据的数据集合. 每个视图分配一个编号v, 编号从 0 开始且$v < n$. 在AH集合对EH_j的身份达成共识的过程中, 可能会发起多次共识过程, 视图编号v的值逐渐递增, 直至在该视图下对EH_j的身份达成共识.

假设每次产生区块的时间间隔为t, AH从接收EH_j的接入申请到对认证达成共识的过程如图 1 所示, 改进的拜占庭容错算法的步骤如下文.

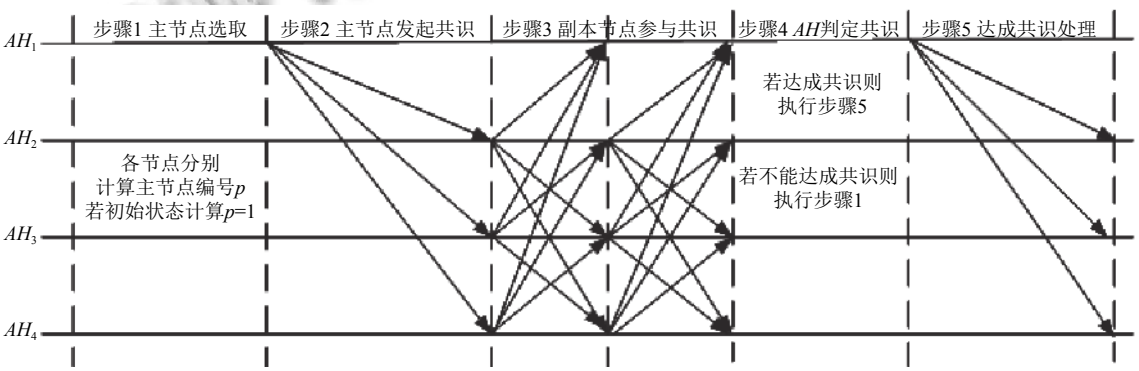

图 1 共识过程

步骤 1. AH选取主节点.

AH内所有节点从各自本地存储的区块中读取h, 计算主节点, AH_i通过:

$$P = (h + v) \bmod n \tag{2}$$

随机决定主节点AH_p, p为主节点的编号. AH_i通过比对p与本节点编号, 判断本节点是否为主节点.

步骤 2. 主节点发起共识.

(1) 主节点AH_p验证m、A并用ECC对m的散列值进行数字签名获得m_{σ_p}, m_{σ_p}表示主节点已验证m、A.

(2) 主节点提取本地存储的区块信息生成包含存储区块版本号、前一区块散列值、h等信息的$block$, 并根据对m、A的验证结果判断是否对$block$使用ECC进行数据签名获得$block_{\sigma_p}$. 如果验证结果为同意接入, 则计算$block_{\sigma_p}$, 否则不计算$block_{\sigma_p}$.

(3) 主节点AH_p向AH中广播包含$<h, v, p, block, block_{\sigma_p}>$的消息, 发起共识.

步骤 3. 副本节点参与共识.

(1) 副本节点AH_i $(i \neq p)$ 收到主节点发送的消息后, 验证m、A, 并用ECC对m的散列值进行数字签名获得m_{σ_i}.

(2) 副本节点$AH_i(i \neq p)$ 根据对m、A的验证结果判断是否对$block$使用ECC进行数据签名获得$block_{\sigma_i}$. 如果验证结果为同意接入, 则计算$block_{\sigma_i}$, 否则不计算$block_{\sigma_i}$.

(3) 副本节点AH_i $(i \neq p)$ 向AH广播包含$<h, v, i, m_{\sigma_i}, block_{\sigma_i}>$的信息.

步骤 4. AH判定共识及操作.

(1) AH内任一节点依据收到的m_{σ_i}、$block_{\sigma_i}$的数量判定共识结果. 若m_{σ_i}、$block_{\sigma_i}$的数量满足关系: $n - f \leq m_{\sigma_i}$的数量$\leq n$, 且$block_{\sigma_i}$的数量$\geq n-f$, 共识结果为同意EH_j接入网络, 并同意生成区块, 此时节点在本地生成共识结果区块, 并将新生成的区块保存在本地; 若m_{σ_i}、$block_{\sigma_i}$的数量满足关系: $n - f \leq m_{\sigma_i}$的数量$\leq n$, $block_{\sigma_i}$的数量$\leq n-f$, 共识结果为拒绝EH_j接入网络, 并拒绝生成区块, 此时节点不做其他操作.

(2) 如果在当前视图下AH在经过$2^{v+i} * t$的时间间隔仍未达成共识或者接收到非法接入认证申请后, 可以更换视图, 直到AH成共识为止. 更换视图的方法为AH_i发起视图更换请求, 视图编号加 1, 当AH_i至少收到$n - f$个来自不同的副本节点返回的响应时, 则返回步骤 1, 重新开始新一轮达成共识的过程.

步骤 5. AH对共识的处理.

(1) AH内已确定共识结果的节点依据共识结果处理此次共识.

(2) 主节点AH_p在获知共识结果后, 处理共识结果, 将共识结果发送给EH_j, 并在AH中广播共识结果.

(3) 在收到主节点AH_p发送的共识结果时仍未知此次共识结果的节点接收主节点AH_p的共识结果, 依据共识结果处理此次共识.

4 实验及结果分析

依据以上对去中心化的网络接入身份认证问题的研究, 本文在 SDN 网络[24–27]中设计并实现了去中心化的接入身份认证方案 BchainNAC. 搭建如图 2 所示的原型环境.

实验环境如下: 一台 OpenDaylight 控制器[28,29]、一台普通交换机、3 台 OpenFlow 交换机、4 台主机. 在交换协议开发平台上部署 Open VSwitch[30,31](开放式虚拟交换机, 简称 OVS) 模拟 SDN 交换机. 使用网桥建立 OpenFlow 交换机与控制器的连接, 由普通交换机作为中转实现控制器与多台 OpenFlow 交换机连接. 主机$h1$、$h2$、$h3$、$h4$为已经连入 SDN 的主机. 主机$h5$为接入认证的申请者. 每一台主机的配置为: CPU 的个数为 1, 核数为 1; 内存容量为 1 GB.

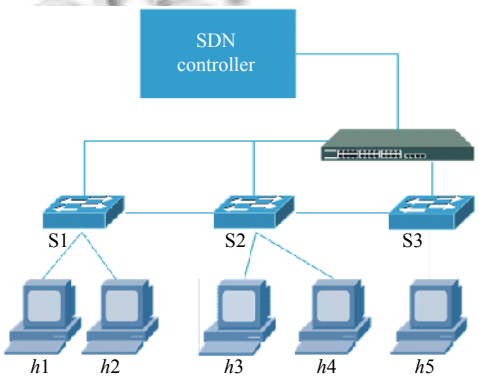

图 2 原型环境

测试场景: 向原型实验环境中逐渐加入 50 台主机, 且每 10 台为一组, 划分为 5 组.

将从申请者发起接入申请到申请者获得认证结果所用的时延记作Tt. 统计每一组加入 SDN 时使用的平均时延Tt, 实验结果如图 3 所示.

测试结果及分析: 参与共识的主机数与完成接入申请所用时延成正比. 从 1 台到 50 台接入 SDN, Tb的增幅不大, 增幅维持在 1 s 之内.

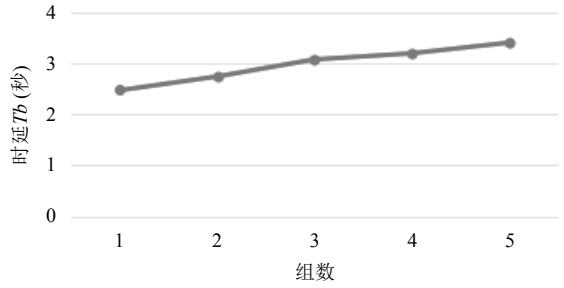

图 3 时延Tb

以组为单位, 分别统计参与共识的主机的 CPU 利用率、内存的使用率, 实验结果如图 4、图 5 所示.

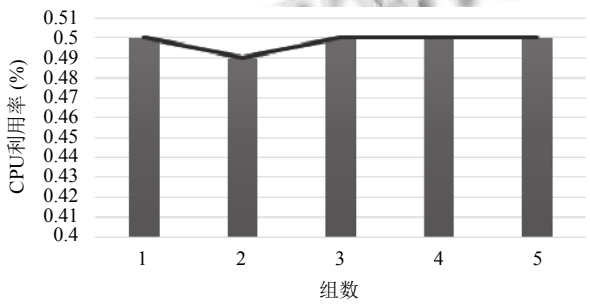

图 4 CPU 利用率

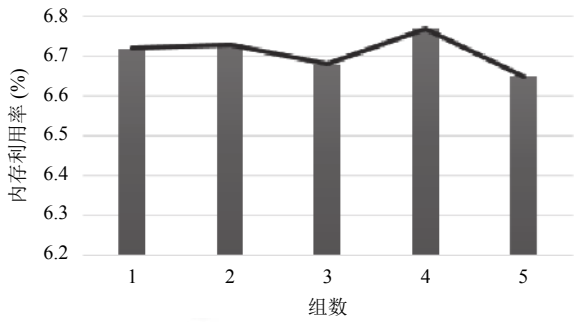

图 5 内存使用率

测试结果及分析: 参与共识的主机 CPU、内存消耗不大, 且每台主机的 CPU 利用率维持在 0.5%、内存使用率维持在 0.67%.

将身份认证共识分成两个部分: 对接入者身份达成共识、通知申请者认证结果和其他节点存储认证结果. 前者使用的时延记作Tc, 后者使用的时延为Tt. 以组为单位, 分别统计每组内平均时延Tc与平均时延Tt占身份认证共识时延的百分比, 实验结果如图 6 所示:

测量结果及分析: 对接入者身份达成共识所用时延为身份认证阶段主要时延. 通过分析, 得知影响共识的主要因素包括: 参与共识的主机的个数; 主机之间通信的质量; 加、解密算法的复杂度.

通过以上实验可知, 在 SDN 网络中, 去中心化的接入身份认证所用时延会随着接入 SDN 网络中主机数的增多而有所提升, 但增幅稳定, 在接受范围之内. 主机因参与共识消耗的 CPU 及内存较低. 通过分析身份认证共识, 可以从影响共识的因素入手, 优化共识过程.

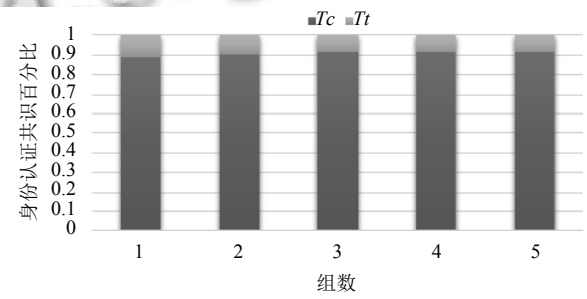

图 6 Tc与Tt占身份认证共识的百分比

5 总结

本文研究去中心化的网络接入身份认证问题, 提出的身份认证方案不泄露除自身公钥以外的任何额外信息, 既满足移动主机接入网络的灵活需求, 又充分考虑网络中主机的不确定性. 下一步将继续优化共识机制的效率, 以便在更大规模网络中推广应用.

参考文献

1 Matias J, Garay J, Mendiola A, *et al*. FlowNAC: Flow-based network access control. Third European Workshop on Software Defined Networks. London, UK: IEEE. 2014. 79–84. [doi: 10.1109/EWSDN.2014.39]

2 王秀磊, 张国敏, 胡超, 等. SDFAC: 软件定义的流接入控制机制. 通信学报, 2015, 36(S1): 188–196.

3 Nayak A, Reimers A, Feamster N, *et al*. Resonance: Dynamic access control for enterprise networks. Proceedings of the 1st ACM Workshop on Research on Enterprise Networking. 2009. 11–18. [doi: 10.1145/1592681.1592684]

4 Lamport L, Shostak R, Pease M. The Byzantine generals problem. ACM Transactions on Programming Languages and Systems, 1982, 4(3): 382–401. [doi: 10.1145/357172.357176]

5 Eduroam Web site: https://www.eduroam.org.

6 公绪晓, 付中南, 吕洁. 基于 eduroam 和 SDN 的无线漫游认证授权技术研究. 华东师范大学学报 (自然科学版), 2015, (S1): 157–162.

7 樊蕊. 跨域身份认证系统的研究与实现[硕士学位论文]. 西安: 西安电子科技大学, 2007.

8 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.

9 Swan M. Blockchain thinking: the brain as a decentralized autonomous corporation. IEEE Technology and Society Magazine, 2015, 34(4): 41–52. [doi: 10.1109/MTS.2015.2494358]

10 Wilson D, Ateniese G. From pretty good to great: Enhancing PGP using bitcoin and the blockchain. Network and System Security. In: Qiu M, Xu S, Yung M, Zhang H, eds. Network and System Security. NSS 2015. Lecture Notes in Computer Science, vol 9408. Springer, Cham. 2015. [doi: 10.1007/978-3-319-25645-0_25]

11 Kraft D. Difficulty control for blockchain-based consensus systems. Peer-to-Peer Networking and Applications, 2016, 9(2): 397–413. [doi: 10.1007/s12083-015-0347-x]

12 李琳, 岳建华. 基于零知识证明的匿名身份认证机制. 计算机科学, 2013, 40(12): 197–199. [doi: 10.3969/j.issn.1002-137X.2013.12.041]

13 李殿伟, 王正义, 赵俊阁. 椭圆曲线密码体制安全性分析. 计算机技术与发展, 2012, 22(4): 227–230.

14 Base58check web site. https://en.bitcoin.it/wiki/Base58Check_encoding.

15 韩家炜, 坎伯. 数据挖掘: 概念与技术 (第 3 版). 机械工业出版社, 2012.

16 Marshall B, Alon R, Manuel S, *et al*. Proofs of Useful Work. http://eprint.iacr.org/2017/203.pdf. [2017-2-27] .

17 韩璇, 刘亚敏. 区块链技术中的共识机制研究. 信息网络安全, 2017, (9): 147–152. [doi: 10.3969/j.issn.1671-1122.2017.09.034]

18 Daniel Larimer. 授权股权证明机制白皮书. http://www.8btc.com/dpos_bitfarm. [2014-04] .

19 卢风顺, 宋君强, 银福康, 等. CPU/GPU 协同并行计算研究综述. 计算机科学, 2011, 38(3): 5–9. [doi: 10.3969/j.issn.1002-137X.2011.03.002]

20 Yu D, He S, Huang Y, *et al*. A fast parallel matrix inversion algorithm based on heterogeneous multicore architectures. 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). Orlando, FL, USA. 2015. [doi: 10.1109/GlobalSIP.2015.7418328]

21 范捷, 易乐天, 舒继武. 拜占庭系统技术研究综述. 软件学报, 2013, (6): 1346–1360.

22 王秀群. 可实用的拜占庭容错系统理论研究[博士学位论文]. 杭州: 浙江大学, 2007.

23 Haber S, Stornetta WS. How to time-stamp a digital document. Journal of Cryptology, 1991, 3(2): 99–111.

24 张朝昆, 崔勇, 唐翯祎, 等. 软件定义网络 (SDN) 研究进展. 软件学报, 2015, 26(1): 62–81.

25 左青云, 陈鸣, 赵广松, 等. 基于 OpenFlow 的 SDN 技术研究. 软件学报, 2013, (5): 1078–1097. [doi: 10.3724/SP.J.1001.2013.04390]

26 江国龙, 付斌章, 陈明宇, 等. SDN 控制器的调研和量化分析. 计算机科学与探索, 2014, 8(6): 653–664.

27 王蒙蒙, 刘建伟, 陈杰, 等. 软件定义网络: 安全模型、机制及研究进展. 软件学报, 2016, 27(4): 969–992. [doi: 10.133280.cnki.jos.005020]

28 许名广, 刘亚萍, 邓文平. 网络控制器 OpenDaylight 的研究与分析. 计算机科学, 2015, 42(S1): 249–252.

29 唐宏, 刘汉江, 陈前锋, 等. OpenDaylight 应用指南. 电信科学, 2017, 33(S1): 267.

30 杨帆, 晏思宇, 黄韬. OVS 的编程扩展技术. 电信科学, 2017, 33(5): 21–28.

31 王文涛, 王奇枫, 郭峰, 等. 基于 Open vSwitch 的 SDN 网络平台构建方法. 中南民族大学学报 (自然科学版), 2014, 33(4): 99–104. [doi: 10.3969/j.issn.1672-4321.2014.04.023]