软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(6):2875−2899 [doi: 10.13328/j.cnki.jos.007305] [CSTR: 32375.14.jos.007305]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# PKI 技术研究与进展*

张 宾[1], 张 宇[1,2], 张伟哲[1,2], 乔延臣[1], 刘 翔[1], 刘鹏辉[1]

[1](鹏城实验室, 广东 深圳 518055)

[2](哈尔滨工业大学 网络空间安全学院, 黑龙江 哈尔滨 150001)

通信作者: 张宇, E-mail: yuzhang@hit.edu.cn

**摘 要**: PKI 系统是目前用户进行安全访问基础资源的重要设施, 通过公共的第三方认证来保障用户访问资源的安全. 随着 PKI 技术的逐步部署应用, 各类在部署应用中的安全问题也随之而来, 攻击者可以通过攻击 PKI 系统窃取用户信息和破坏用户访问. 从 PKI 的基本工作原理出发, 全面介绍 PKI 系统在实际部署应用中涉及的各个要素, 包括 PKI 架构、工作流程、证书、证书链、证书撤销、CI 日志服务. 在 PKI 基本工作原理的基础上, 重点从 PKI 系统安全角度全面梳理和总结 PKI 系统工作过程中面临的安全问题, 包括 PKI 系统应用过程中面临的运营风险和技术风险、PKI 系统的测量、风险检测及各类 PKI 系统的风险防范技术, 并对未来 PKI 领域的研究方向进行展望.

**关键词**: 公钥基础设施; 公钥基础设施安全; 证书; 证书透明度

**中图法分类号**: TP309

中文引用格式: 张宾, 张宇, 张伟哲, 乔延臣, 刘翔, 刘鹏辉. PKI技术研究与进展. 软件学报, 2025, 36(6): 2875–2899. http://www.jos.org.cn/1000-9825/7305.htm

英文引用格式: Zhang B, Zhang Y, Zhang WZ, Qiao YC, Liu X, Liu PH. Research and Progress of PKI Technology. Ruan Jian Xue Bao/Journal of Software, 2025, 36(6): 2875–2899 (in Chinese). http://www.jos.org.cn/1000-9825/7305.htm

## Research and Progress of PKI Technology

ZHANG Bin[1], ZHANG Yu[1,2], ZHANG Wei-Zhe[1,2], QIAO Yan-Chen[1], LIU Xiang[1], LIU Peng-Hui[1]

[1](Pengcheng Laboratory, Shenzhen 518055, China)

[2](School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China)

**Abstract**: PKI system is currently an important facility for users to securely access basic resources. It ensures the security of users' access to resources through public third-party authentication. With the gradual deployment and application of PKI technology, various security issues in deployment arise. Attackers can steal user information and disrupt user access by attacking the PKI system. This study starts from the basic working principle of PKI and comprehensively introduces all the elements involved in the practical deployment and application of the PKI system, including PKI architecture, workflow, certificates, certificate chains, certificate revocation, and CI log services. Based on the basic working principles of PKI, this study focuses on comprehensively sorting out and summarizing the security issues that the PKI system faces during its operation from the perspective of PKI system security, including operational and technical risks, measurement and risk detection of PKI system, and various risk prevention technologies for PKI systems. Finally, future research directions in the field of PKI are prospected.

**Key words**: public key infrastructure (PKI); public key infrastructure security; certificate; certificate transparency

公钥基础设施 (public key infrastructure, PKI) 是基于证书的公钥密码系统的安全基础服务设施. PKI 系统由第三方可信公共机构, 通常称为证书认证中心 (certification authority, CA), 为用户签发管理数字证书, 使用一对数学上相关的密钥, 称为私钥和公钥, 对机密信息进行加密和解密, 从而为用户提供强大身份验证, 安全数据和通信, 以

* 基金项目: 国家重点研发计划 (2024YFB31NL00105); 国家自然科学基金青年科学基金 (62102202); 广东省基础与应用基础研究重大项目 (2019B030302002); 鹏城实验室重大攻关项目 (PCL2023A05)
收稿时间: 2024-01-24; 修改时间: 2024-08-15; 采用时间: 2024-10-14; jos 在线出版时间: 2025-02-19
CNKI 网络首发时间: 2025-02-19

满足通信双方的机密性, 数据完整性和不可否认性要求.

Kohnfelder[1]在 1978 年首次提出数字证书的概念. PKI 逐步得到了许多国家企业和政府的广泛重视, 相关技术也得到了快速发展. 1988 年, X.509 标准第 1 版正式推出. 1995 年, IETF 成立公钥基础实施 (public key infrastructure X.509, PKIX) 工作组, 将该标准用于互联网. 2005 年, 推出目前比较成熟的第 3 版 X.509 标准. 经过多年的理论研究和实际部署, PKI 技术逐步过渡到商业化应用阶段, 支撑了全球重要通信信息系统的安全.

国内外都在积极推进 PKI 的部署, 美国是最早提出 PKI 技术并进行积极推广应用的国家, 在 1996 年成立了联邦 PKI 指导委员会, 极大地推进了 PKI 在世界范围内的应用. 加拿大、欧盟等国家也相继建立了自己的 PKI 体系. 这些国家的 PKI 系统由政府授权, 并进行统一的审核和管理. 在亚洲, 韩国的 PKI 系统建设较早, 他们的认证架构分成 3 个等级: 信息通信部-国家 CA 认证中心-下级授权认证机构. 日本的 PKI 系统的发展也比较迅速, 他们的管理架构分为公众和私人两个部分. 这些国家相继通过了 PKI 相关法律, 并在建设上积极推动 PKI 技术的应用.

我国的 PKI 技术应用起步较晚, 在基础理论等方面还依赖于国外的先进技术, 但近 10 多年来我国的 PKI 行业的发展十分迅速. 国内认证中心分为行业性认证中心 (例如 CFCA 中国金融认证中心, CTCA 中国电信认证中心), 区域性认证中心 (例如 SHECA 上海市数字证书认证中心, 天津 CA) 和纯商业性认证中心 (天威诚信, 沃通等) 和企业型认证中心. 这些机构有力地推动了 PKI 系统在我国的部署应用.

我们通过分析目前的证书透明度 (certificate transparency, CT) 日志, 发现美国占全球市场份额的 76% 左右, 远超过其他国家, 德国占 11% 左右, 中国占 2% 左右, 位居第三, 但中国市场份额增长率为全球最高. PKI 在金融、政府和电信行业得到了广泛应用, 特别是在目前世界性技术出口限制与竞争的大环境下, 国产 PKI 技术会发挥越来越重要的作用. 近年来, 支持国密算法的 SM2 证书也已列入国家标准, 国内多家公司也在推动相关应用系统的部署工作.

CA/Brower 论坛是一个由证书颁发机构和互联网浏览器软件供应商组成的自愿工作组. 该组织试图解决证书颁发机构引入的诸多安全风险, 并于 2011 年 11 月颁布了证书发布和管理基线要求 (baseline requirements for the issurance and management of policy-trusted certificates)[2], 并且在近些年持续更新了该基线要求, 然而, 该组织中只有 20% 的组织为商业证书颁发机构, 有 25% 的商业证书颁发机构虽然参与论坛但与论坛组织关系松散. 目前仍有许多组织在实际部署 PKI 时忽视论坛的基线要求, 许多 CA 发布的证书不符合论坛发布的基线要求. 由于 CA/Brower 论坛不是标准化组织, 无法强制要求论坛成员执行基线要求, 无法保证基线要求在实际部署 PKI 时的效用. 因此, 目前一方面需要加强措施以保障在实际部署 PKI 系统时遵从论坛基线要求, 另一方面需要对论坛基线要求进行标准化, 以保障 PKI 系统更好的发展.

虽然 PKI 技术取得了广泛的应用和发展, 但是目前国内外鲜有工作对相关技术进行梳理. 依据目前对国内外调研的文献, 国内仅有 2015 年《密码学报》文献 [3] 对 2015 年前 5 年内的技术发展进行了总结和分析, 国外仅有 2013 年 S&P 国际会议中文献 [4] 对 PKI 技术的发展进行剖析和总结. PKI 技术近些年取得了快速发展, 特别是在 PKI 测量和安全方面, 研究人员做了大量的新的工作, 然而, 目前国内外并没有一篇文献对相应的工作进行系统总结. 究其原因是 PKI 技术点较多, 相关文献研究关注点也比较发散, 梳理总结起来难度较大. 为了更好地帮助从事该领域研究的学者能全面快速了解 PKI 技术的总体进展情况, 本文主要对 PKI 技术发展的重要相关文献进行全面梳理和总结, 以填补这个空白. 由于 PKI 的应用以 Web PKI 最为典型和广泛, 业界和学术界研究和应用最广, 本文 PKI 的调研和分析以主要定位于 Web PKI.

本文从 PKI 的基本工作原理出发, 深入浅出地分析了 PKI 系统的各个环节. 在读者详细了解 PKI 整个工作体系的基础上, 重点对 PKI 系统面临的风险、风险测量检测及风险防范上将现有工作进行分析和总结, 以帮助科研人员和工程人员更加透彻了解 PKI 系统工作原理, 熟悉各类 PKI 风险情况及检测防范措施, 为更好地改进 PKI 系统提供帮助. 本文第 1 节对 PKI 的基本工作原理进行阐述, 包括架构及工作流程、证书、证书链、交叉签名. 第 2 节对证书撤销进行分析. 第 3 节对 CT 日志进行详细剖析. 第 4 节对 PKI 风险包括面临的运营风险和技术风险进行全面分析. 第 5 节对 PKI 系统的测量进行总结. 第 6 节对 PKI 系统的风险检测进行分析. 第 7 节对 PKI 系统的各种风险防范措施进行分类总结. 第 8 节对 PKI 系统的研究工作进行未来展望. 最后进行简要总结.

## 1 PKI 基本工作原理

### 1.1 架构及工作流程

最初, 在 PKI 体系中, 主要有 4 个参与方[5].

(1) CA: 证书颁发机构, 是 PKI 体系的核心设施, 负责颁发, 吊销证书;

(2) 证书注册机构 (registration authority, RA): 核实申请证书的用户身份, 此功能可以合并到 CA 中;

(3) 证书订户: 使用证书来表明自己的合法身份, 通常为网站;

(4) 证书依赖方: 用户通过浏览器访问网站, 在建立 TLS 会话的过程中获取证书, 鉴别网站身份.

为了更加高效安全地运营 PKI 系统, 目前 PKI 系统增加了保障证书签发行为透明公开的证书透明日志服务器和用于存储证书及证书吊销列表 (certificate revocation list, CRL) 的目录服务器. PKI 系统主要架构如图 1 所示.

通常情况下, PKI 系统只需要单向认证, 整个过程如图 2 所示.

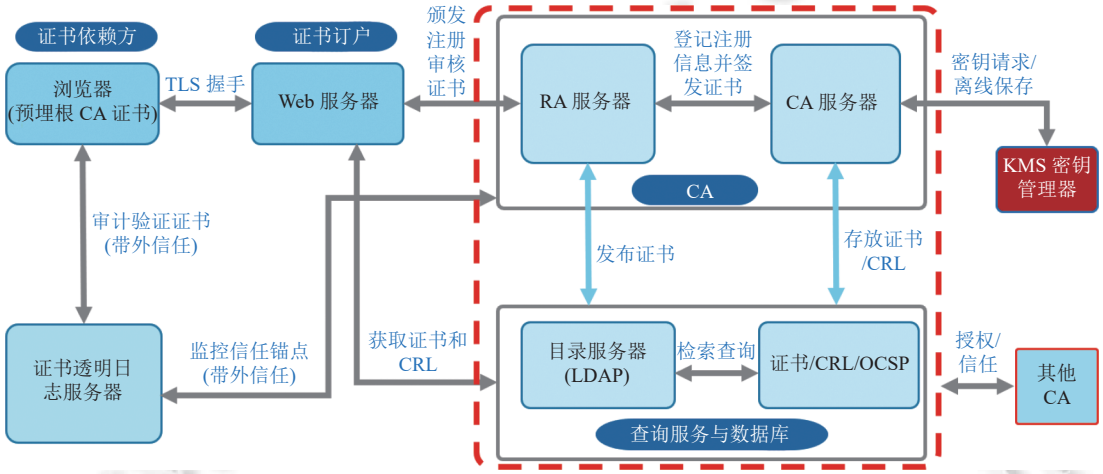


图 1 PKI 系统架构图

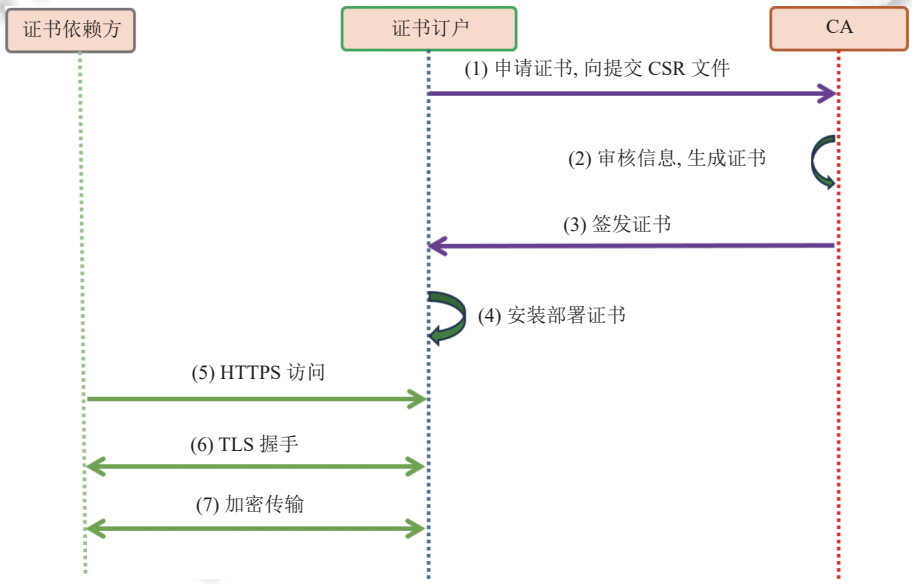


图 2 PKI 系统工作流程图

(1) 申请证书: 证书订户通常指网站向 CA 申请证书, 需要先生成证书签名申请 (certificate signing request, CSR), 主要包含公钥、网站域名 (可以有多个) 以及私钥对申请信息的签名; 当网站用户申请本地证书时, 对证书注册请求消息使用 CA 证书的公钥进行加密和自己的私钥进行数字签名. 如果 CA 要求验证密码, 则证书注册请求消息必须携带密码, 向 CA 提交申请证书的 CSR 文件.

(2) 审核信息: CA 可以通过线上或线下多种手段验证申请者提供信息的真实性, 核实后生成网站证书, 审核信息时, CA 使用自己的私钥和网站的公钥解密数字签名并验证数字指纹[6]. 数字指纹一致时, CA 才会审核实体身份等信息, 审核通过后, 同意申请, 颁发网站证书.

(3) 签发证书: CA 使用 PKI 实体的公钥进行加密和自己的私钥进行数字签名, 将证书发送给网站用户, 也会发送到证书/CRL 存储库, 同时在透明日志系统中记录其签发行为.

(4) 安装部署证书: 网站在 TLS 网关 (如 Nginx) 上配置证书 (如果网站证书往上还有若干中间证书, 则需要把中间证书和网站证书都放在一起), 启用 HTTPS 服务; 当网站申请本地证书时, 会用配置的 HASH 算法对 CA 证书进行运算得到数字指纹, 与提前配置的 CA 服务器的数字指纹进行比较, 如果一致, 则 PKI 实体接受 CA 证书, 安装本地证书, 否则 PKI 实体丢弃 CA 证书.

(5) 证书依赖方使用浏览器通过 HTTPS 访问网站.

(6) TLS 握手: 在 TLS 握手过程中读取到网站证书, 通过对比证书的信息摘要验证证书的合法性, 通过 CRL 或在线证书状态协议 (online certificate status protocol, OCSP) 方式验证对端实体的本地证书的有效性, 并验证证书是否被内置的 CA 信任, 验证完毕后协商一个用于加密通信的对称密钥.

(7) 加密通信: 在后续的浏览器用户到服务器网站的数据访问过程中使用协商的对称密钥进行加密通信.

另外, 如果 CA 机构签发管理的网络节点需要管理其下属的其他网络节点, 那么该节点可以作为一个 CA 中级根节点, 用于发行下属网络节点的证书, 签发工作过程与上述步骤相同, 通过这种层次式签发证书构造信任关系链, 最终形成一个以最高权威 CA 为根的树型证书结构, 为网络中的所有节点构造信任链. 在两个网络节点通信时, 一个网络节点需要通过层层下载 (网站查询并下载) 或利用本地保存的证书对通信的另一端节点进行层层证书认证, 确保对方的身份可信. 查询 CRL/OCSP, 验证证书的有效性, 并从网站证书开始, 循着证书链验证每一个节点证书的合法性.

### 1.2 证　书

PKI 系统中的证书是证明身份和授权公钥之间连接的签名文件, 主要目的是验证网络标识符 (即 IP 地址或 DNS 域名) 是否控制某些加密公钥, 生成证明此链接的证书. X.509 标准是密码学里公钥证书的标准格式, 对于一个经权威可信机构签名的证书, 证书的拥有者就可以用他的证书及对应的私钥创建安全的通信. 证书是一个数据结构[7], 其中包含一个 public key 和一个 name; 权威机构对证书进行签名, 将 public key xxx 关联到了 name xx; 对证书进行签名的 entity 称为 issuer, 证书中的 entity 称为 subject. 图 3 示例了一个典型证书的结构. Validity 区域是证书的有效时间, Issure 区域指示了证书的签发机构, Subject 区域指示了哪个域名被签发, Subject Public key info 区域指示了证书的公钥信息, Extensions 区域指示允许的密钥使用、策略、OCSP 和 CA issuer 的一些信息, Signature 区域是具体对证书的签名.

证书依赖方通常指浏览器, 可以通过 issuer 的公钥验证签名是否合法, 通常证书依赖方在自己的信任库 (trust store) 预先配置了一个它信任的根证书 (trusted root certificates, 也称为 trust anchors) 列表[8], 预配置的方式是从另一个 PKI 来 bootstrap, 可以用一些自动化工具, 通过 SSH 将根证书拷贝到依赖方. 信任仓库中的根证书是自签名的 (self-signed): issuer 和 subject 相同.

证书的信任锚 (trust anchors) 非常关键. 一个自签名的证书, 只有当它进入信任仓库的过程是可信任时, 才应该信任这个根证书. 在 macOS 上, 信任仓库是由 Keychain 管理的. 在一些 Linux 发行版上, 可能只是 /etc 或其他路径下面的一些文件. 主流浏览器默认使用的信任仓库基本上由 3 个组织维护.

(1) 苹果: iOS/macOS 上运行的客户端程序和浏览器使用苹果维护的信任锚仓库.

(2) 微软: 微软为 Windows 系统维护了自身的信任锚仓库, Windows 系统上运行的客户端程序和大部分浏览器通常应用 Windows 系统内嵌的信任锚仓库.

(3) Mozilla: Mozilla 的网络安全服务 (network security services, NSS) 是 Mozilla 维护的信任锚仓库. Mozilla 主要为其 Firefox 浏览器, Thunderbird 电子邮件客户端和其他与 Mozilla 相关的软件产品开发和使用 NSS, 但由于其开放和透明, 也作为其他一些信任仓库的基础, 目前 Linux 和 Android 操作系统的信任锚仓库基本源于 NSS.

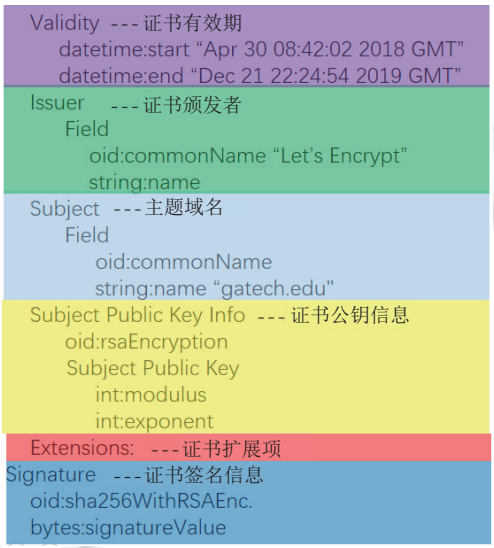


图 3 证书示例图

目前大部分的客户端程序基本使用操作系统的信任锚仓库. Mozilla 的客户端比如 Firefox 浏览器等使用 NSS, 中国的 360 浏览器也维护了自身的信任锚仓库, 通过浏览器版本的更新完成信任锚仓库的更新. 信任仓库中通常包含了超过 100 个由这些程序维护的常见 CA. 如 Let's Encrypt、Symantec、DigiCert、Entrust 等.

### 1.3 证书链

为了保障 PKI 系统的安全性, 根 CA 一般不直接对用户颁布证书, 通常他授权下级 CA 即中级根 (intermediate CA) 对用户颁布证书, 中级根还可以进一步授权下级 CA, 形成一个树型的签发机构[9], 这样根 CA 可以在离线状态仍然保证证书的签发, 根 CA 的私钥也可以离线保存, 以保障整个系统的安全性. 在根 CA 离线的前提下, 为使证书安全可扩展, 根 CA 的私钥只在很少情况下使用, 主要用来签发中级根证书.

中级根用相应的中级根私钥来签发用户证书. 中级根证书通常并不包含在信任仓库中, 所以撤回比较容易, 因此通过中级根, 就实现了证书分发的在线和自动化, 这种用户实体、中级根、根 CA 组成的证书捆绑机制, 形成了一个证书链 (certificate chain). 证书订户由中级根签发实体证书, 中级根由根 CA 签发中间证书, 根 CA 进行自签名形成根证书. 如图 4 所示.

证书链也就是 RFC 5280[5]中描述的证书路径, 指的是以最终实体证书开头, 后跟一个或多个 CA 证书, 且通常最后一个是自签名证书, 具有如下关系: 除了链上的最后一个证书外, 每个证书的颁发者等于其后一个证书的主题, 主题就是使用者; 除了链上的最后一个证书外, 每个证书都是由其后的一个证书签名; 最后一个根证书是信任锚, 根证书是自签名证书, 用户下载根证书就表示信任该根证书签发的所有下级证书. 在操作系统或浏览器中, 已经内置了一部分受信任的根证书.

为了验证证书实体的合法性, 需要获得颁发该证书的 CA 机构公钥, 这个公钥就存在于上一级证书中. 因此, 为了验证证书的合法性, 就需要沿着证书链向上追溯直到根证书为止. 证书链用来检查目标证书链中的第 1 个证书中包含的公钥和其他数据是否属于其主题. 用证书链中的下一个证书的公钥来验证它的签名, 一直检查到证书

链的尾端, 由于最后一个证书是信任锚, 成功达到该证书将证明目标证书可以信任. 在实际验证过程中还要涉及额外的检查, 例如验证证书和中间证书的有效性, 包括验证证书和中间证书的有效日期, 是否被吊销等.

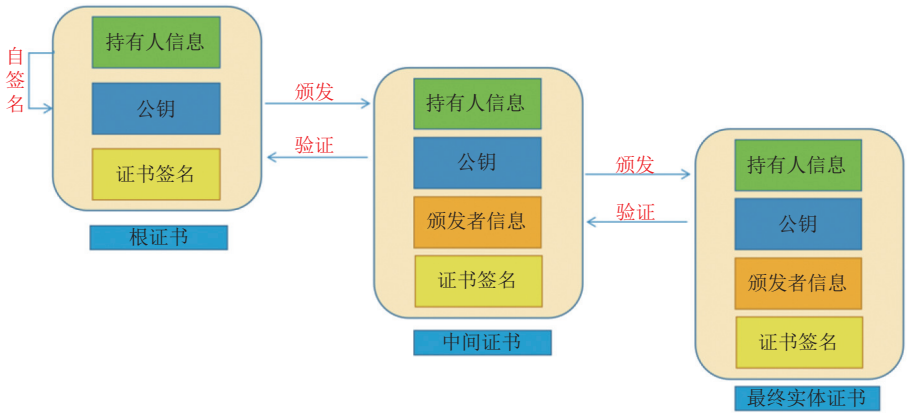


图 4 证书链例图

通过一个简单的例子说明证书链的认证过程.

(1) 当浏览器客户访问 baidu.com 的时候, 百度会返回网站证书 (记为证书 X) 及中间证书 (记为证书 Y) 给浏览器.

(2) 浏览器本地存有 baidu.com 的根证书, 发现证书 X 不由根证书签发, 于是无法用根证书验证证书 X 是否可信.

(3) 浏览器根据证书 X 的 issuer 字段, 找到证书 X 的签发机构, 并找到其对应的中间证书 Y.

(4) 用根证书的公钥验证证书 Y (公钥对证书 Y 做计算得到结果 A, 对比证书 Y 中的签名 B, 一致则认为证书 Y 由根证书签发), 验证通过则认为证书 Y 可信任.

(5) 证书 Y 可信任后, 通过证书 Y 的公钥去验证证书 X (Y 的公钥+证书 X 的内容, 计算得到签名 A, 拿到证书 X 自带的签名 B), A 和 B 一致则认为证书 X 可信.

(6) 于是证书 X 由只信任根证书, 到信任根证书的代表证书 Y, 信任证书 Y 等于信任根证书.

### 1.4 交叉签名

在研究证书链的构建和验证方式时, 需要特别注意的是, 具体的证书可以是不同的证书链的一部分, 链上的所有证书都有效, 这是因为可以为相同的主题和公钥生成多个 CA 证书, 但使用不同的私钥 (来自不同的 CA 或来自同一 CA 的不同的私钥) 进行签名, 也就是说, 交叉签名是存在两个不同的证书, 证书具有相同的主题和公钥, 但具有不同的颁发者和签名. 因此, 尽管单个 X.509 证书只能具有一个颁发者和一个 CA 签名, 但是它可以有效地链接到多个证书, 从而建立完全不同的证书链, 这对于 PKI 与其他应用程序之间的交叉认证至关重要.

假设有两个受信的信任链. 如图 5 所示, 有 ROOT R1 和 ROOT R2 两个信任链, 如果有些客户端只预置 R1 根而没有 R2 根, 而有些客户端只预置 R2 根而没有 R1 根. www.test2.com→…→R1 是一条证书链, www.test2.com →…→R2 也是一条证书链. 交叉签名机制可以使得在客户端只信任 R1 或者 R2 的时候, 也可以同时验证同一个证书.

比如利用交叉签名的机制, 使用 R1 对 R2 进行签名生成一张叫 R1-R2 的证书 (即使用 R1 的私钥对 R2 进行签名), 这样在 ROOT R2 中就包含 R2 CA 和 R1-R2 两个证书, R1-R2 拥有和 R2 CA 一样的 subject+公钥, 与 R2 CA 唯一的区别就是 R2 CA 是自签名证书, 其 issuer 是 R2, 而 R1-R2 是中间证书, 其 issuer 是 R1. 接着, 使用 R1-R2 来签发 www.test2.com, 即 www.test2.com 的 issuer 是 R2, 这张 R1-R2 就是交叉证书.

该交叉证书签发的网站证书通过验证证书链, 既可以让只信任 R1 的客户端进行验证, 又能让只信任 R2 的客

户端进行验证. 即对于图 5 中无论只预置了 R1 还是只预置了 R2 的浏览器, 都可以通过交叉验证访问 www.test2.com, 而 www.test1.com 由于其签发者 R1 未经 R2 的交叉签名, 该网站只能被预置 R1 根的浏览器进行访问, 而无法被预置 R2 根的浏览器访问.

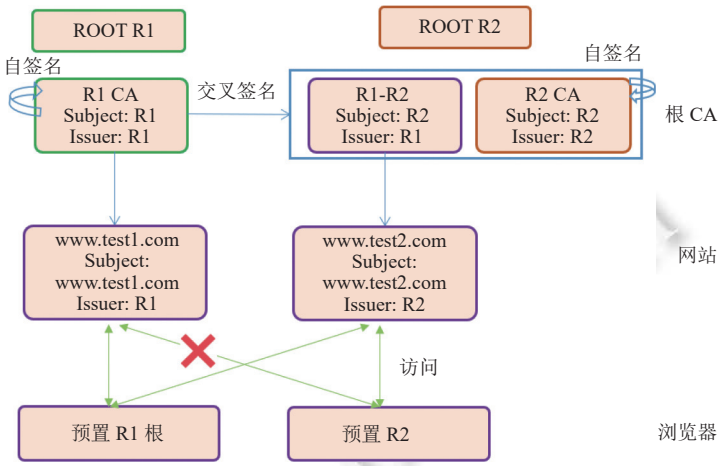


图 5 交叉签名示例

文献 [10] 基于被动 TLS 监视器和公共 CT 日志中包含 7 年多的 2.25 亿个证书和 93 亿个信任路径数据集, 系统地分析了 Web PKI 中交叉签名的现状和发展趋势, 使用交叉签名的优缺点, 交叉签名可以快速引导新的 CA (例如 Let's Encrypt 通过 IdenTrust 的交叉签名获得浏览器的信任), 并通过提供向后兼容性来实现无中断的用户体验. 但是, 交叉签名可能会导致证书吊销后仍保留有效的信任路径, 非浏览器的软件通常盲目信任所有 CA 证书并忽略撤销, 文献进一步为交叉签名提出了新的规则和指导, 在保持交叉签名应用优势的同时减轻附带的风险.

## 2 证书撤销

在 X.509 标准中使用证书撤销列表 (CRL) 机制实现证书的撤销[8]. CRL 是一个已经被撤销 (Revoke) 或挂起 (Hold) 的证书的列表. 在 CRL 中包含一个颁发日期 (thisUpdate) 以及下一个 CRL 的颁发日期 (nextUpdate). 由这两个日期信息用户可以确定当前拥有的 CRL 是否是最新的.

证书轻型目录存取协议 (lightweight directory access protocol, LDAP)[11]的主要功能是维护和发布其内部数据库中由 CA 签发的证书和 CRL, 用户定期从 LDAP 服务器上下载整个 CRL 到客户端, 客户端执行证书是否被撤销的验证; 对于采用在线证书状态协议 (OCSP) 的目录机制, 用户可以直接在 LDAP 服务器上实时查询证书的状态. 但是 CRL 机制由于发布是周期性的, 存在证书状态不能及时更新的问题, 同时用户需要每次都下载 CRL, 存在带宽占用量较大问题 .

文献 [12] 从实际 PKI 系统部署应用的角度详细剖析了 CRL 的缺陷, 当互联网中部署的应用证书节点逐渐增加时, 会使得 CA 经常签发 CRL, 这样导致 CRL 迅速变大, 如果增加 CRL 更新时延, 会导致 CRL 不能及时反应证书真正的撤销状态, 如果及时更新 CRL, 则会由于每次都需要下载完整的 CRL 导致占用大量的网络带宽, 这两者是一对无法调和的矛盾, 任何基于 CRL 的改进措施都无法从根本上解决问题.

针对 CRL 缺点, RFC 2560[13]提出 OCSP. OCSP 旨在通过允许客户端向 CA 查询单个证书的吊销状态来减少 CRL 的开销, OCSP 允许客户端生成对给定证书序列号状态的 HTTP 请求, 证书中含有客户端应该查询的具体的 OCSP 的链接, 收到客户端验证请求后, OCSP 应答器返回一个签名响应, 其中包括证书的当前状态 (良好、已吊销或未知) 以及证书的有效期, 客户端可以缓存响应, 通常以天为单位.

OCSP 解决了 CRL 的许多问题, 大大减轻了 CRL 机制的带宽占用, 但在客户端发起连接时都要进行 OCSP 查询的请求, 也带来了一定的通信开销. 此外, 使用 OCSP 会向 CA 透露有关用户浏览行为的信息, 会带来用户隐私

泄露方面的隐患.

OCSP Stapling[14]是一种用于提高 SSL/TLS 证书验证性能和安全性的机制, 允许服务器缓存 OCSP 响应并将其发送到客户端, 作为 TLS 握手的一部分, 它通过将 CA 的响应缓存在 Web 服务器上, 从而避免了每次客户端发起连接时都要进行 OCSP 查询的开销, 并提高了验证的性能和安全性. 因此, 当与支持 OCSP Stapling 的服务器通信时, 客户端接收服务器的证书和证书有效性的 OCSP 声明. 客户端可以验证 OCSP 语句, 因此可以确保证书不会被吊销. OCSP Stapling 是一个很有前景的改进, 因为它减少了 CA 带宽成本和网页加载时间, 但并不是所有的浏览器都支持它, 有些浏览器只是忽略了响应.

OCSP Stapling 消除了与验证证书吊销状态相关的大部分延迟惩罚, 但是 OCSP Stapling 并没有完全消除客户端的延迟损失, 因为 OCSP Stapling 只包括叶证书的 OCSP 响应 (该协议不允许服务器包括中间证书的缓存 OCSP 响应). 最近提出的对 OCSP Stapling 的扩展[14]通过允许服务器包括中间和叶证书的 OCSP Stapling 响应来解决这一限制, 但尚未被广泛采用.

文献 [15] 改进了 CRL 和 OCSP 两种证书撤销机制, 分别对 CRL 和 OCSP 进行了分析和评估, 并在其结果的基础上提出了一系列的改进方式, 为证书撤销的执行提升了效率, 并针对不同的应用环境, 提出了证书撤销机制的选择方式. 文献 [16] 在 OpenSSL 开源库的基础上扩展了支持国密 SM2 算法的密码卡设备, 设计并实现了一套基于 OCSP 协议的证书状态查询系统.

文献 [17] 使用 74 次完整的 IPv4 HTTPS 扫描, 发现 PKI 所服务的证书中有相当大的一部分 (8%) 被吊销, 而且从客户端的延迟和带宽角度来看, 获取证书吊销信息通常是昂贵的; 文献继续研究了 30 种不同的 Web 浏览器和操作系统组合的吊销检查行为, 发现浏览器通常不检查证书是否被撤销 (特别是移动浏览器, 它们通常从不检查); 文献还研究了谷歌 Chrome 内置的 CRLSet, 发现 CRLSet 仅涵盖所有撤销的 0.35%; 文献 [17] 提出为了减少 CRL 大小, CA 可以简单地维护更多, 更小的 CRL (在极端情况下, 每个证书都可以分配一个唯一的 CRL, 称为 oneCRL, 从而获得 OCSP 的近似值), 但目前很少有 CA 采用这样的方法.

表 1 从撤销系统目标的完整性 (不需要 CA 参与用户就可以查验证书的撤销状态)、查验性 (查验不了证书状态就拒绝接受)、高效性 (是否给用户带来高的延迟和大的带宽消耗)、私密性 (不向证书权威机构暴露用户私密信息)、可审计性 (第三方机构可以验证是否将证书标记为已吊销), 易部署性 (是否支持增量部署) 来比较目前提出的撤销机制.

表 1 撤销机制的对比

| 对比项 | CRL | OCSP | OCSP Stapling | CRLSet | oneCRL |
|---|---|---|---|---|---|
| 完整性 | √ | √ | √ | × | × |
| 查验性 | √ | × | √ | × | × |
| 高效性 | × | × | √ | √ | √ |
| 私密性 | √ | × | √ | √ | √ |
| 可审计性 | √ | √ | √ | × | × |
| 易部署性 | √ | √ | × | √ | √ |

传统的 CRL 需要对浏览器和 CA 进行修改, 浏览器需要一个模块来获取, 存储和搜索 CRL 列表, CA 需要维护 CRL 可公开访问的服务器; OCSP 也需要对浏览器和 CA 进行修改, 浏览器需要一个模块来执行 OCSP 查询, CA 需要维护独立的 OCSP 响应程序以及时提供吊销信息; OCSP Stapling 需要在浏览器和网站端进行修改, WEB 服务器必须定期轮询 OCSP 服务器以了解证书的吊销状态, 并在 TLS 握手期间将 OCSP 响应和证书一起发送到浏览器; CRLSet 和 OneCRL 需要对浏览器进行大量修改, 供应商需要定期从世界各地的主要 CA 中抓取 CRL, 并组成自己的综合列表, 与 CRL 相反, 此列表不包括服务器叶证书, 而只包括中间 CA 证书.

## 3 CT 日志服务

证书透明度 (certificate transparency, CT) 是 Google 提倡的为了提高 PKI 系统信赖度的新技术[18]. CT 要求

CA 公开其颁发的每一个数字证书到证书日志中, 并对其监控和审计, 以确保证书透明度机制正确有效运行, 从而缓解证书错误颁发, 减少安全问题以及隐私泄露的发生. CT 也可以理解为是一个证书预签备案系统, 目前被列为 RFC 国际标准 (RFC 6962[19], RFC 9162[20]), 意在实时监测 CA 系统被黑或者 CA 机构草率签发的不是用户自愿申请的证书, 和让浏览器能及时阻止非法证书的使用, 以保护 HTTPS 加密安全, 其作用是每当 CA 机构签发证书时, 将所有证书的签发行为都记录到审计日志中. 通过这个审计日志服务器, 网站运营者, 域名管理者等人员确认自己的域名对应的证书有无被其他 CA 机构签发, 以此来防止非正常签发的可信证书被他人滥用.

谷歌于 2013 年 6 月推出 RFC 6962 标准后就开始打造证书透明生态, 这个生态由证书透明日志系统, 浏览器, CA 系统和监督系统等 4 个部分组成. 谷歌首先是研发和开源了证书透明日志系统, 要求 CA 在签发每一张证书给用户之前把这张预签证书提交到证书透明日志系统上获得一个日志系统的证书签署时间戳 (signed certificate timestamp, SCT), 如图 6 所示, CA 把这个签名数据 (SCT 列表字段的数据) 写到 SSL 证书中, 写入 SCT 列表数据后的 SSL 证书才能给用户部署使用.

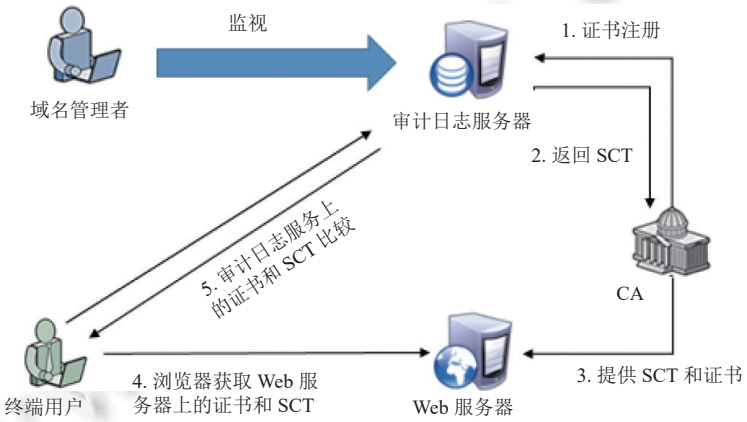


图 6 CT 日志工作原理

TLS 服务器必须将一个或多个日志中的 SCT 与证书一起提交给 TLS 客户端. TLS 客户端必须拒绝没有有效 SCT 作为最终实体证书的证书. TLS 服务器 (如: Web 服务器) 将从审计日记里取得 SCT, 然后与证书一起在 TLS 客户端 (如: 浏览器) 给出提示. 这样, TLS 客户端就会根据获取的证书和 SCT 来确认审计日志中是否有该证书的记录.

由于 CA 签发的 SSL 证书已经在给用户之前就已经在证书透明日志系统备案, 而这个日志系统是全球公开可查询的数据库, 并且是采用区块链一样的默克尔树哈希技术, 使得 CA 的证书签发行为是不可否认的, 因为这个基于默克尔树的数据库只能追加数据而不能修改数据. 这个公示证书签发行为之所以称之为透明就是因为这个数据全球实时公示, 这个签发行为的公示是生态中最重要的一环, 任何人包括网站域名所有者都能在证书正式签发部署使用之前就能实时发现这张证书是否是网站域名所有者同意签发的证书, 这就高效地保证了任何错误签发或者恶意签发的证书都能在第一时间被发现和被制止.

由谷歌发起的证书透明日志系统从 2013 年开始已经记录了 74 亿张证书, 并以每月至少 1 亿多张的速度在不断增加中. 从谷歌浏览器 68 版本开始 (2018 年 5 月) 就已经强制要求全球所有 CA 签发的每一张证书都必须提交到指定的证书透明日志中才会被浏览器信任, 有效地保障了证书的自身安全. 通过分析证书透明日志, 能够看到哪些 CA 在何时以及为哪些域名颁发了证书. CT 诞生后, 基于 CT 的工作主要有 CT 的扩展和改良, CT 的部署和基于 CT 的测量这 3 个方面.

● CT 扩展. 文献 [21] 扩展了证书透明系统使之可以有效地处理证书撤销, 展示了如何使用此扩展构建安全的端到端电子邮件或消息系统, 而不需要信任证书颁发机构, 也不需要依赖复杂的对等密钥签名, 使得端到端加密邮件成为可能, 与未加密邮件相比, 用户不需要理解或关心密钥或证书. 文献 [22] 提出一种基于日志和面向域的综

合架构 PoliCert. 在公共日志中记录主题证书策略和证书, 提供存在和不存在的加密证明. PoliCert 进一步限制了 CA 在证书签发过程中的权力, 并且要求 certificate 和 subject certificate policy 都存储在日志服务器上, 可用于公开审计. 相比已有 PKI 系统, 证书持有者的权力得到了增强, 可以与 CA 协商来设定自己的证书策略, 且不会随意修改.

文献 [23] 提出 ARPKI, 可确保证书相关操作 (如证书颁发, 更新, 吊销和验证) 透明. 同 PoliCert 方案类似, 服务器证书上要求有多个不同 CA 的签名, ARPKI 以较低的开销有效地处理认证过程, 并且不会导致 TLS 的额外延迟, ARPKI 提供了极为强大的安全保障, 其中泄露 $n-1$ 个可信签名和验证实体不足以发起伪造攻击, 此外, 由于它的所有操作都是公开的, 可以阻止不当行为.

文献 [24] 定义了日志记录方案的安全特性, 并正式证明 CT 实现了这些特性. 文献 [25,26] 提出了一种有效的 Gossip 协议来检测几种类型的日志不一致. 文献 [27] 通过零知识证明, 完成了在不暴露用户隐私的情况下对日志进行审计, 并通过承诺对非公共子域进行支持绑定和隐藏属性. 文献 [28] 研究了部署 CT 框架的动机, 并提出了部署状态过滤器来检测域的部署状态以抵御攻击.

PKISN[29]按时间顺序在公共日志中记录所有证书和撤销, 它可以有效撤销 CA 证书, 同时保证撤销之前颁发的实体证书不会变为无效. CONIKS[30]基于 Merkle 前缀树构建透明的密钥目录, 允许用户在保持隐私的同时审计他们的公钥. 软件透明度[31]要求开发人员向公共日志提交包括所有源代码和元数据在内的更新包, 监视器在每次更新时重新构建所有更改的包, 并检查生成的二进制文件是否匹配.

● CT 部署. 文献 [32] 完成了一项关于跨网络采用 CT 的全面研究, 包括合规性, 用户体验和潜在风险. 文献 [33] 描述了 11 个公共日志, 重点是记录的证书及其在 TLS/HTTPS 中的使用. 文献 [34] 研究了 Alexa Top-1M 网站中 CT 的采用以及 SCT 的交付方法. 文献 [35] 完成了一项关于采用各种 TLS/HTTPS 安全增强的大规模研究. 文献 [36] 测量分析了 CT 随时间的演变, 并从安全和隐私的角度探讨了公开证书的域名带来的影响, 测量分析发现 CT 日志中的证书呈指数级增长, 网站对 CT 的支持也在不断增加, 随着 CT 部署的不断增加, 由于 CT 日志中的所有证书都可见, 存在信息泄露的问题, 文献进一步讨论了公共日志中证书导致的域名信息泄露造成的新的安全问题. 文献 [37] 探讨了第三方监视器的 TLS/HTTPS 配置, 并与常见网站进行了比较.

● CT 测量. CT 日志中的记录有助于了解 TLS/HTTPS 生态系统, 文献 [38] 通过主动扫描量化各种证书来源的覆盖范围, 将日志中的证书与来自被动测量, 主动扫描和证书搜索引擎的数据集成在一起, 以呈现证书的完整视图. 利用公共日志中的数据, 文献 [39] 分析了 Let's Encrypt 在不同组织, 主机和域中采用的证书服务. 文献 [40] 调查了违反证书颁发标准的行为. 文献 [41] 利用来自公共日志和被动测量的 TLS 服务器证书, 分析了伪造证书的属性.

## 4 PKI 系统风险

本文所述的 PKI 系统风险不考虑对 TLS 协议及其涉及的加密算法的攻击, 主要针对 PKI 系统本身, 包括认证过程本身, 谁被允许成为证书颁发机构 (锚定信任), 如何授权 (信任的传递性), 如何吊销证书 (信任的维护), 以及用户如何与证书信息交互 (指示和解释信任).

### 4.1 运营风险

运营风险主要从我国 PKI 系统的现状来分析可能存在的运营方面的风险.

(1) 几乎所有最重要系统都部署美国 CA 签发的 SSL 证书

如图 7 所示, 我国前十大银行的网银证书、淘宝网、京东商城、支付宝和财付通都是部署的美国 CA 颁发的 SSL 证书.

目前, 我国各网银网站、各大电商网站、第三方支付网站、网上证券基金网站等的可信网站身份几乎都是由美国 CA 来做身份验证, 每年花上亿元. 美国 CA 在审查完我国银行、证券、电商、支付等单位的真实身份后签发一张服务器证书给各个网站, 用于证明网站的真实身份和用于网站与客户端之间的数据加密传输, 保证网站的可信身份和数据加密安全, 也就是说, 我国的网络空间的最重要的一端, 服务器端的安全与可信是由美国 CA 来保障的, 其巨大安全隐患和风险, 主要表现在以下 3 个方面.


图 7　国内主要 PKI 应用机构证书

1) 根 CA 停止证书签发风险: 当证书注册依赖于其他国家根 CA, 一旦注册在其他国家的根 CA 对新的域名证书停止签发服务或网络中断, 如发生海底光缆意外断裂, 则使用国外 CA 颁发的证书的系统就无法正常使用, 因为证书正常使用时需要到美国的签发服务器上查验证书是否有效, 将造成相应域名网站的加密中断, 导致本国新的域名网站和用户之间无法获得加密应用服务.

2) 根 CA 吊销已签发的证书风险: 一旦注册在其他国家的根 CA 的证书被相应的根 CA 吊销, 证书将失效, 对应的域名网站的加密服务将消失, 域名网站和用户之间加密服务将被中断, 会引起相应系统的瘫痪, 轻则影响用户的使用, 重则影响到国家金融系统稳定, 甚至是影响到国家安全.

3) 证书被篡改风险: 依赖方信任的 CA 被攻击或执行恶意操作, 签发包含虚假信息但可验证成功的服务器证书, 则容易造成中间人攻击, 致使对应网站被劫持到非法网站. 目前中国用户访问网银系统的访问信息, 如什么时候访问, 从哪里访问, IP 地址和使用什么浏览器, 每日有多少次访问中国的网银系统, 多少次访问支付宝和淘宝网, 这些重要的机密信息都完全掌握在国外 CA 的手中, 是非常危险的.

(2) 很多重要系统都没有部署 SSL 证书

《谷歌透明度报告》[42]披露的最新数据表明, 近些年加密流量的访问有明显提升. 如图 8 显示了不同操作系统平台上 Chrome 浏览器用户 HTTPS 访问的频次, 可以看到从 2016 年开始到 2024 年加密访问流量有明显的增加, 但是目前仍然有相当部分的流量是非加密流量, 特别是 Linux 操作系统和 Windows 操作系统中的用户. 我国很多重要系统包括一些重要的政务部门和电子商务网站没有部署 SSL 证书, 许多邮件系统都没有部署 SSL 证书和使用客户端证书来加密邮件. 也就是说: 这些重要系统中的机密信息都是明文传输的, 可以毫不费力地被偷走, 无需费力去解密. 为了确保电子政务网站和电子商务网站的用户机密信息安全, 必须部署 SSL 证书来保证机密信息安全.

同时, 移动 APP 中的很多连接都没有使用 SSL, 其中包括个别银行的移动网银 APP, 各种社交 APP 等. 而这些裸奔的 APP 大量在各种免费 WiFi 上运行. 由此可见, 各种移动应用包括移动支付的安全问题也是非常严峻的, 移动 APP 必须使用 HTTPS 与服务器通信, 才能保证移动用户的机密信息安全.

(3) 许多重要的系统部署了浏览器不信任的自签证书

自签名证书是自己颁发给自己的证书, 即证书中的颁发者和主体名相同, 这类证书是由网站或服务器的所有者自己创建并签名的, 没有经过第三方认证机构的验证, 证书持有者可以自行决定其内容和真实性, 这增加了伪造的风险, 由于自签证书的免费性和易用性, 常被用于部署在测试平台上. 根据文献 [43,44] 的测量结果显示, 很多系统部署了自签名证书, 部署了自签证书当然比不部署证书的系统安全, 但是如果用户都习惯了即使浏览器显示不

信任的安全警告也继续浏览的话, 这就帮了欺诈网站和假冒网站的忙, 因为假冒网站往往由于拿不到全球信任的证书而采用自签证书, 浏览器会有安全警告, 但由于用户已经养成了继续浏览的习惯而遭遇中间人攻击, 导致个人账户被盗. 值得注意的是: 各个高校的 VPN 系统和各种需要登录的系统都在使用自签证书, 正在不断“培养”大批学生养成这种不安全的上网习惯!

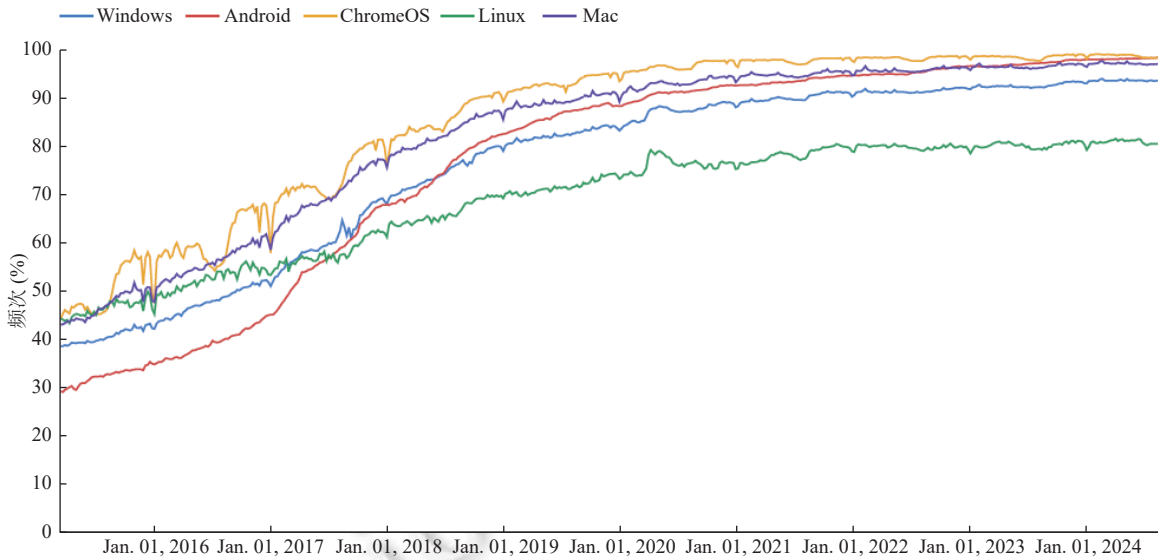


图 8 不同平台上 Chrome 用户 HTTPS 访问的频次

### 4.2 技术风险

(1) 客户端在进行安全通信前需要验证服务器证书, 如果客户端不进行验证或者不能正确验证, 攻击者可以伪装成合法的服务器[45], 使用自己的非法证书与客户端建立 SSL/TLS 连接, 或者在 CRL 更新之前使用已吊销的证书并中断 OCSP 查询, 从而窃听或篡改客户端与合法服务器之间的通信数据. 尽管当前的浏览器平台验证收到的站点证书是否与主机名匹配, 但一些非浏览器软件存在严重的验证不足, 文献 [46] 研究发现, 13 500 个安卓应用程序中有 1 074 个没有验证主机名. 在一些私有网络上, 特别是在公司环境中, 组织的根证书可以配置为员工的机器. 然后, 组织可以代理 (即 MITM) HTTPS 连接, 具有专门为该任务设计的中间盒来执行内容检查, 代理承担证书验证的全部责任, 文献 [47] 研究发现, 许多实现存在验证缺陷.

(2) 被攻击或恶意 CA 执行恶意操作, 签发包含虚假信息, 如果用户信任的第三方公共 CA 本身从一些政治或利益因素执行恶意操作, 签发的证书虽然可以被用户验证通过, 但由于证书存在恶意的虚假信息, 容易对用户造成攻击. 比如, TurkTrust CA 公司的虚假信息证书, Comodo CA 公司安全事件[48], DigiNotar CA 公司被攻击[49]等, 都显示此类攻击完全可能发生. 2015 年 3 月, 谷歌发现中国互联网络信息中心 (CNNIC) 向一家埃及公司颁发了一份不受约束的中间证书, 该公司使用该证书拦截访问谷歌的网络用户的通信 (即 TLS MITM 攻击).

(3) 使用证书的客户端存放过多的信任的根证书, 为了和不同 CA 机构颁发证书的系统建立安全连接, 一般在用户浏览器和使用的操作系统中, 都会预置大量的根证书, 这些预置的默认信任的根证书对应的 CA 系统不一定都是安全健壮的. 攻击者只要能侵入其中安全强度最低的一个 CA 系统, 他就能随心所欲地对所有域名的服务器证书进行签发, 从而建立有效的 TLS 连接, 进行 MITM 攻击, 也就是说, 客户端使用的证书系统的脆弱程度决定于脆弱性最弱的 CA, 这是目前 PKI 系统在实际应用中面临的困境.

(4) 单一的 CA 提供商通常是不健壮的, 很容易出现单点失效问题. 不管出于何种原因, 比如运营操作出错或安全问题, 网络业务很可能变得无法使用. 除此之外, 为了提高 TLS 的可靠性, TLS 证书使用者应该依赖多个 CA 来获取证书. 例如, 为了减少业务中断风险, 证书的生命周期管理解决方案应该支持在系统发生故障时可以从一个

CA 切换到另一个 CA. 使用多个 CA 可以确保依赖第三方对 CA 更改的灵活性, 为了确保设备的长期兼容性, 应该使用多个 CA 来确保支持的依赖方生态系统能够灵活应对这些变化. 为了实现这一点, 通常的解决方案是, 业务需要一个可以互换使用的 CA 生态系统. 这方面证书自动化部署在很大程度上能有所帮助, 提供了一种与 CA 交互以获得这些证书的标准化方式. 只有当多个 CA 实现该标准化协议, 并且这些 CA 都能够得到扩展, 以满足依赖特定的需求时, 多 CA 系统才能真正发挥作用.

(5) CA 在进行域名验证时, 依赖于准确的域名信息, 对 CA 接收准确 DNS 记录的能力的任何干扰, 例如, DNS 缓存中毒都可能导致证书颁发不正确. 2015 年, CA WoSign 因一系列操作问题受到公众审查[50], 其中一个问题是发布错误, 该错误允许子域 (例如, evil.github.com) 的所有者接收父域 (github.com) 的证书, 这与 WoSign 的其他违规行为一起, 促使根存储运营商开始讨论删除对 WoSign 证书的信任; 2016 年 7 月, 一项新发现显示, 一家以色列 CA StartCom 能够颁发由 WoSign 签署的证书. 对该事件的深入调查最终显示, 2015 年 11 月 1 日 WoSign 获得了 StartCom 100% 所有权. 进一步的证据表明, StartCom 的 CA 证书很可能早在 2015 年 12 月就与 WoSign 操作集成了, 当时 WoSign 证书似乎即将从根存储中删除. WoSign 对 StartCom 的秘密收购强调了运营 CA 控制透明度对安全网络的重要性. 对 WoSign 证书的不信任仍然允许 WoSign 通过其 StartCom CA 证书悄悄颁发受信任的证书.

(6) 目前很多恶意软件使用 SSL/TLS 加密技术对安全网站进行拦截或者钓鱼攻击, 比如检测 HTTPS 流量的客户端软件和网络中间盒通过充当透明代理, 它们终止并解密客户端发起的 TLS 会话, 分析内部 HTTP 明文, 然后发起到目标网站的新 TLS 连接完成拦截攻击. 再比如一些网站通过使用 PKI 系统仿造目标网站欺骗用户[51], 比如网站 amazon.com-offers.com 通过使用证书, 一些用户看到浏览器显示锁图标等 HTTPS 访问标志, 可能会相信他们与亚马逊有安全连接. 根据 WatchGuard 在 2022 年发布的一份互联网安全报告[52], 大多数恶意软件隐藏在安全网站使用的 SSL/TLS 加密中, 而且有逐步上升趋势, 目前 93% 的恶意软件隐藏在加密之后, 如果不检查这些流量, 就会错过对大多数恶意软件的检测.

## 5 PKI 系统测量

PKI 系统测量主要指对互联网中 PKI 系统的整体服务情况进行测量, 主要包括对服务器网站部署的证书的情况, CA 机构的情况, TLS 的部署情况等反映 PKI 系统总体状况的证书生态进行测量. 由于对 CT 日志和证书吊销的测量在前文已有专门阐述, 这部分的 PKI 系统测量忽略了这两部分测量的内容.

Holz 等人在 2011 年发表了一项研究[43], 该研究基于对 Alexa Top 100 万域名的定期扫描, 并通过对慕尼黑科学研究院网络上 TLS 流量的被动监测, 在 2009 年 10 月–2011 年 3 月期间, 该小组平均每次扫描收集了 212 000 份证书, 共收集了 554 292 份唯一证书, 重点关注叶证书 (网站服务器证书) 的动态情况和证书在 IP 地址之间的分布, 并试图对所服务证书的总体质量进行粗略分类.

2013 年, Durumeric 等人[53]通过反复扫描 IPv4 地址空间来分析 PKI 的状态. 他们简要考虑了 Alexa Top Million 中有多少个能使用服务器名称指示 (server name indication, SNI) 访问. SNI 是 TLS 的普遍支持的扩展, 它充当选择器, 允许客户端指定特定的主机标头, 目标服务器使用此信息来正确地将流量路由到预期的服务, SNI 使得在单个服务器上管理多个网站证书服务. 他们测量发现当时并没有广泛要求使用 SNI, 他们调查了根授权机构, 中间授权机构和 Web 服务器使用的叶证书之间的信任关系, 最终确定并分类了 1 800 多个能够颁发证书以保证任何网站身份的实体, 并发现了可能危及生态系统安全的做法.

Chung 等人[44]在 2016 年进行了类似的研究, 但重点关注无效证书的来源和用途. 他们展示了为什么 Web 的 SSL 生态系统中充斥着如此多的无效证书, 它们来自哪里, 以及它们如何影响安全性. 他们使用超过 8 000 万个数据集, 确定大多数无效的证书来自少数类型的终端用户设备, 并且与有效的证书相比具有显著的不同性质.

文献 [38] 研究发现, 扫描 IPv4 地址空间中的证书只捕获了整个证书生态系统的一小部分, 而证书透明度 (CT) 包含了占主导地位且不断增加证书份额. 他们发现由于 SNI 部署, IPv4 扫描不再提供 TLS 服务器配置的代表性视图, IPv4 扫描遗漏了超过 2/3 的有效证书, 一种更全面但更容易获得的方法是结合 CT 和 Censys 数据, 构建可信 HTTPS 证书的全面视图.

Kotzias 等人[54]对 TLS 部署进行了 5 年的纵向分析, 他们专注于 TLS 部署和行业实践中由于协议漏洞的披露而引起的变化. 文献 [55] 测量了 TLS 1.3 的部署速度, 发现标准化后仅 15 个月, 约 20% 的连接都使用了 TLS 1.3, 在流行域上的部署率为 30%, 在 com/net.org 顶级域 (TLD) 中的部署率约为 10%, 同时展示了 Cloudflare 本身就带来了相当大的部署数量, 并描述了 Facebook 和谷歌等参与者如何利用他们对客户端和服务器端点的控制来试验该协议, 并最终进行大规模部署.

## 6 PKI 系统风险检测

PKI 系统风险检测指对 PKI 系统在互联网应用过程中出现风险情况的检测, 其中包括对 CA 签发的证书是否符合 CA/Browser 论坛的基线要求的检测, 对 CA 签发的恶意证书的检测, 对客户端和 TLS 库存在的证书验证漏洞的检测, 对 HTTPS 拦截即中间人攻击的检测, 密钥泄露漏洞及对隐藏根 CA 的检测等.

一系列 PKI 系统的安全事件表明, PKI 整个生态环境的安全性, 主要取决于 CA 公司的整体证书服务水平. 为此, CA/Browser 论坛在 2011 年推出基线要求[2]并于 2013 年[56] 及后续进行扩展, 以对 CA 公司证书的服务进行进一步规范和约束, 尽管出台了这样的基线约束, 但目前的证书服务仍然远跟不上 CA/Browser 论坛基线要求限定的服务水平.

文献 [57] 通过扫描 IP 地址空间和对前 100 万个 Alexa 网站进行爬取, 总共收集了互联网上公开使用的 148 万多张 CA 证书, 收集的这些证书的时间轴贯穿了基线要求发表的前后几年. 数据表明, 在基线要求生效日期前一年, 只有 0.39% 的已颁发证书严格遵守所有基线和扩展指南, 在接下来的一年里, 符合标准的上升到 0.73%, 大多数大型商业 CA 基本能遵守基线要求, 而违规行为往往随着授权的频率和深度以及特定 CA 表现出的发行政策的多样性而增加, 另一方面, 大量小型公司和政府运营的 CA 合规性差. 因此, 全面提升 CA 证书生态环境的安全程度, 仍然任重而道远.

文献 [58] 提出了一种从可信 CA 中检测恶意证书的方法. 该方法是从大量的证书集合中开发的, 通过使用深度神经网络 (DNN) 构建机器学习模型来自动分类. Georgiev 等人[45]证明, 一些广泛使用的应用程序和开发库, 如亚马逊的 EC2 Java 库、SDK、osCommerce 和 Java Web 服务等, 都存在证书验证漏洞, 导致了通用的 MITM 攻击, 这些漏洞主要因为使用了设计不当的 API, 如 JSSE 和 OpenSSL.

Brubaker 等人[59]设计了一种自动化方法, 用于测试几个著名 TLS 实现的证书验证模块. 他们首先使用 ZMap 在互联网上扫描 443 端口打开的服务器, 并收集所有可用的证书, 然后, 他们对证书参数进行了排列, 编制了一个包含 800 万个 Frankencerts 的列表. Brubaker 等人使用 Frankencerts 和差分测试, 在这些常用的 TLS 实现 (例如, OpenSSL, GnuTLS 和 NSS) 中发现了 200 多个差异.

与 RFC 5280 相比, Chau 等人[60]使用符号执行方法来测试 9 个 TLS 库的证书验证过程. 他们发现了 48 起违规事件, TLS 库忽略了几个 X.509 证书参数, 如 pathLenConstraint, keyUsage, extKeyUsage 和 notBefore 有效期. 为了总结 HTTPS 拦截带来的安全威胁, 文献 [61,62] 研究了数十种插入自建根证书并执行 TLS 拦截的客户端应用程序 (例如, 防病毒软件和代理), 他们通过代码分析评估了证书生成和验证的整个过程, 并发现了实现缺陷.

Durumeric 等人[63]从网络服务器的角度进一步分析了 HTTPS 拦截, 并得出结论, 中间盒的存在会严重降低 HTTPS 连接的安全性. 他们对 HTTPS 拦截的普遍性和影响进行了全面的研究, 展示了 Web 服务器可以通过识别 HTTP 用户代理和 TLS 客户端行为之间的不匹配来检测拦截. 他们开发构建一组启发式方法来检测拦截, 并在 3 家大型网络提供商部署了这些启发式方法, 他们通过调查流行的中间盒和客户端安全软件, 发现几乎所有软件都降低了连接安全性, 许多软件引入了严重的漏洞.

Zhang 等人[64]在 2014 年进行了一项应对一个普遍存在的漏洞 Heartbleed 的研究. 该漏洞导致无法检测的密钥泄露, 他们开发了新的启发式方法, 以确定最受欢迎的 100 万个网站在证书管理方面对此漏洞的反应, 以及对使用它们的客户端的安全性有何影响. 他们发现绝大多数易受攻击的证书都没有重新颁发, 在响应 Heartbleed 重新颁发证书的域中, 60% 不会撤销其易受攻击的证书, 如果这些证书最终没有被撤销, 其中 20% 的证书将在两年或更长时间内有效. 这一发现的后果令人担忧, 在未来很长一段时间内, 浏览器仍可能容易受到恶意第三方的攻击,

这些第三方可以使用窃取的密钥伪装成一个受感染的网站.

为了描述 HTTPS 拦截的普遍性和原因, 2014 年, Huang 等人[65]主动向 Facebook Web 服务器发送 SSL 握手并提取证书. 他们发现拦截率为 0.2%, 他们对最终用户在连接 Facebook 时遇到的公钥证书进行了大规模分析, 在 300 万个 Facebook 的连接中, 发现了 6000 多个可疑证书的使用. 随后, 在 2016 年, O'Neill 等人[66]进行了类似的实验, 通过主动连接受控服务器检测到 TLS 拦截, 并根据 Issuer 字段的信息发现了恶意软件和垃圾邮件的 HTTPS 劫持.

文献 [67] 提供了 Web PKI 生态系统中隐藏根 CA 的第 1 个客户端全国视图, 通过与一家领先的浏览器供应商合作, 在 5 个月内分析了志愿者用户在网络访问中的证书链及其验证状态, 总共捕获了超过 117 万个隐藏的根证书, 确定了大约 5000 个持有隐藏根证书的组织, 包括模拟大型可信证书的伪根 CA, 文献进一步揭示了隐藏根 CA 和证书的实现存在很大缺陷, 弱密钥和签名算法等问题普遍存在, 并呼吁社区审查本地根存储的完整性.

文献 [68] 为了帮助 Web PKI 参与者了解控制每个 CA 证书的组织, 开发了一个对 CA 操作行为进行建模和集群的系统 Fides, 以便在共享操作控制下检测 CA 证书. Fides 发现的集群建立了一个新的 CA 所有权数据库, 该数据库为 241 个 CA 证书更正了 CA 运营商, 并将覆盖范围扩大到 651 个新 CA 证书, 从而实现了 CA 证书控制的更完整画面. 该应用程序设计用于域名所有者检测错误颁发的证书, 并用于公共审计员揭露不符合基线要求的证书.

文献 [69] 分析了 CA 颁发证书的机制本身容易受到中间人攻击各种风险情况, 通过自治系统 (AS) 利用边界网关协议 (BGP) 中的漏洞劫持流向受害者域的流量, 演示了对手可以用来获得伪造证书的攻击的各种场景, 首次在现实世界中演示 BGP 攻击对 PKI 系统的影响. 为了评估 PKI 的漏洞, 他们收集了 180 万个证书的数据集, 发现对手能够获得绝大多数域的伪造证书, 并提出和评估了两种保护 PKI 的对策: 1) CA 从多个有利位置验证域, 使其更难发起成功的攻击; 2) CA 的 BGP 监控系统, 用于检测可疑的 BGP 路由并延迟证书颁发, 使网络运营商有时间对 BGP 攻击做出反应.

## 7 PKI 系统风险防范

为了解决 PKI 中的安全问题和减少对 CA 的信任, 目前已经提出了各种建议. 如图 9 所示, 现有的方法可以分为以用户为中心, 以 CA 为中心或以域名为中心, 下文分别简要阐述.

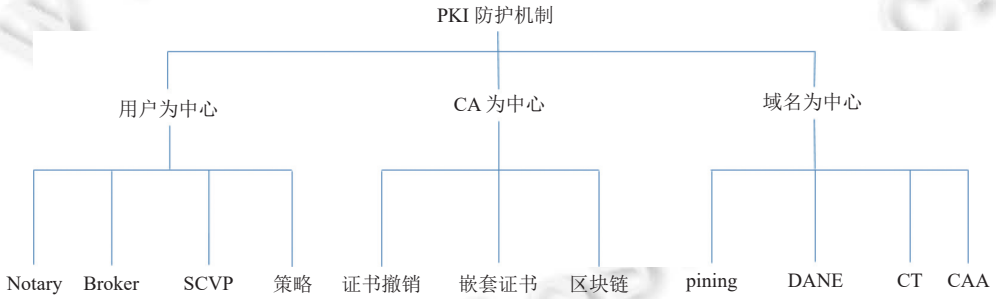


图 9 PKI 系统风险防御策略分类

### 7.1 用户为中心的防范机制

以用户为中心的防范机制主要是指在证书依赖方进行安全防护, 通常指改造浏览器或 APP 使其支持一些本地的库及保障策略等, 在证书应用的客户端保障证书的应用安全, 分为公证人、信任代理、基于服务器的证书验证协议和本地策略这 4 种保障机制.

(1) 公证人 (Notary)

从不同的网络角度观察服务器证书的想法已被用于改进 Web PKI 信任模型的几项提案中, 这一想法是在 Perspective[70]中引入的, 其中 Perspective 将公证人定义为部署在网络上不同位置的公开可用的半信任主机. 主要思想是, 在客户端获得服务器证书之后, 它可以将接收到的证书与从公证人的网络角度获得的服务器证书进行比

较. 证书之间的差异可能表示证书替换, 意味着可能出现攻击. Convergence[71]通过使用反弹公证人来防止隐私问题, 允许使用其他方法 (DANE, CA 等) 进行身份验证, 并解决公证滞后问题, 改进了 Perspective 方案.

Doublecheck[72]建议使用 TOR 网络而不是公证人. DetecTor[73]是一个类似的解决方案, 将 TOR 思想的使用扩展到任何协议. ICSI 证书公证人[74]和 EFF SSL 观测站项目[75]收集 SSL 证书并发布有关它们的统计信息. ICSI 证书公证人还提供了一个公共 DNS 接口来查询其数据库. 这些项目通过积极调查网站来收集证书. 文献 [76] 提出服务器公证来检测 PKI 模型中可信 CA 或子 CA 的潜在滥用, 该方案与当前模型互操作, 并允许逐步部署, 与早期方案的区别在于, 它为 TLS 服务器分配了一个更活跃的角色, 以检测针对其域的证书替换攻击.

(2) 信任代理 (trust broker, TB)

四角信任模型[77]引入了信任代理, 根据 CA 的信息为依赖方提供有关可信度的定量信息和有关公钥证书预期用途的定性信息, 信任代理独立于 CA, 并有效地分散了对单个 CA 的信任. 文献 [78] 提出通过添加 TB 实体来正式扩展原始的 X.509 信任模型, 这个新模型现在被纳入 X.509 标准的第 8 版. 文献 [79] 提出量化证书质量 (QoCER), 以允许证书依赖方对证书做出决定. 文献 [80] 提出一种实现信任代理的方法, 通过评估 CA 颁发证书的质量来向证书依赖方提供有关 CA 的信任信息, 该实现演示了证书依赖方如何在全球网络环境中对证书持有者做出明智的决定, 而不需要大量的处理资源.

(3) 基于服务器的证书验证协议 (server-based certificate validation protocol, SCVP)

RFC 5055[81]提出了一个基于服务器的证书验证协议允许客户端将认证路径构建和认证路径验证委托给服务器, 根据包含一个或多个信任锚的验证策略执行路径构造或验证 (确保路径中的所有证书都没有被吊销). 它允许简化客户端实现和使用一组预定义的验证策略, 但基于 SCVP 的方法需要额外的连接来查询存储库, 这会增加建立 HTTPS 连接时的延迟.

(4) 策略

文献 [82] 提出客户端可以定义证书本地策略 (例如: 加密要求, 基于观察到的历史的证书的一致性等), 达到对证书更加灵活的控制使用. 它使客户能够做出适合客户的信任决策, 例如, 客户端可能决定不信任带有 RSA 密钥的证书小于 2 048 比特, 或者不允许外国 CA 认证企业的内部电子邮件系统等. 文献 [82] 设计了一个策略引擎并进行原型实现, 该引擎为客户端提供了改进的能力, 使其能够根据从服务器收到的证书序列做出信任决策.

### 7.2 以 CA 为中心的防范机制

以 CA 为中心的防范机制主要指在 CA 设计层面对证书应用风险进行防范, 比如设计更加高效安全的 CA 架构, 发布更加高效的证书以加快验证过程, 对签发的证书进行更加高效安全的吊销管理等. 分为证书吊销机制、嵌套证书、区块链证书机制这 3 种保障机制.

(1) 证书撤销机制

证书撤销机制[5]及扩展[83]在第 2 节已经详细阐述, 基于证书撤销目前机制的缺陷, 用户仍有可能使用过期证书进行通信的风险, 一些工作[15]从当前的 CA/B 模型修改了 CRL 分发的体系结构, 不从用户客户端获取 CRL (和 OCSP 响应), 浏览器供应商定期从 CRL 分发点获取 CRL, 并向浏览器发送更新的 CRL 以供存储, 所有主要浏览器都通过软件更新手动吊销高风险证书.

OCSP Stapling[84]是 SSL/TLS 扩展, 允许服务器缓存 OCSP 响应并将其发送到客户端, 在证书 OCSP Stapling 下, 证书持有者定期获得签名和带时间戳的状态报告, 并在握手时将其包含在证书中. 为了进一步降低证书撤销机制的风险, 文献 [85] 提议用短期证书代替长期证书以减少证书撤销.

(2) 嵌套证书 (nested certificates)

文献 [86] 提出使用可有效验证的嵌套证书路径来代替经典证书路径, 嵌套认证是一种用于高效证书路径验证的新方法, 基本思想是为其他证书颁发特殊证书 (称为嵌套证书), 嵌套证书可以与 PKI 中的经典证书一起使用, 这种 PKI 被称为基于嵌套证书的 PKI, 嵌套证书路径中第 1 个证书以加密方式验证, 其他证书仅通过快速哈希计算验证, 文献在嵌套认证开销和证书路径验证的时间改进之间进行了权衡, PKI 到嵌套证书 PKI 的转换保留了

PKI 中现有的层次结构和信任关系, 嵌套证书可以很容易地被采用到 X.509 标准证书结构中, 嵌套证书的 PKI 和嵌套证书的验证效率和撤销优势使其适用于无线应用的分层 PKI.

(3) 区块链证书机制

DPKI[87]为基于去中心化数据存储的 PKI 系统提供了一个初步的理论框架, 证书的注册、更新、续订或吊销应通过安全的分散式对等系统进行. Wang 等人[88]提出了一种基于区块链的方案来构建附加日志以实现证书透明, 该方案由于如下设计在存储成本方面效率低下. 1) TLS 证书在其生命周期内定期添加到区块链中; 2) 为每个吊销的证书向区块链中添加 CRL; 3) 定期向区块链添加发布密钥对. 为了降低研究文献 [86] 中的存储成本, CertLedger[89]通过状态对象管理证书和证书操作, 状态对象是由数据和管理它的不可变智能合约代码组成的数字文档, 状态更改由事务触发, 并通过状态对象进行跟踪, 因此, 在验证过程中可以跟踪证书的最新状态, 并且由于删除了区块链中的 CRL 而降低了存储成本.

CertChain[90]进一步提出了双计数布隆过滤器, 以实现证书吊销检查的高效空间利用和高效查询, 改进传统 PKI 的吊销管理, 他们提出了一种新的基于区块链的吊销管理和状态验证系统, 其中使用公共区块链共享和传播吊销信息. 还设计了一个名为 CertOper 的新数据结构, 用于快速跟踪和公共审计. Yan 等人[91]提出了一种在基于区块链的 PKI 系统中记录证书的存储优化方案, 当一个区块中的所有证书都无效时, 可以从区块链中删除该区块的主体, 保留区块标题以确保完整性.

文献 [92] 提出了一种安全且可扩展的基于区块链的 PKI 解决方案 BPKI, BPKI 引入了称为审计师的新实体来监督 CA 的证书注册操作, 以消除域名抢占攻击. 此外, BPKI 使用可验证伪随机函数和双区块链结构设计了一种新的委托共识, 以解决可扩展性问题, BPKI 在可扩展性方面优于现有的基于区块链的 PKI 解决方案.

文献 [93] 目标是使 CA 完全去中心化, 利用区块链技术提出了一个名为 ProofChain 的去中心化 PKI 框架, 该框架在去中心化的 CA 组之间提供完全信任. 文献 [93] 的解决方案提供了所有传统的 X.509 PKI 操作 (即注册、更新、续订和撤销), 使其与现有的 PKI 标准兼容, 并针对流行的安全标准 (即 CIA 三元组模型) 和 PKI 对抗性攻击对 ProofChain 进行了评估, 并在以太坊网络的专用测试平台上实现 ProofChain, 跨越各种现实世界的 PKI 场景.

### 7.3 以域名为中心的防范机制

以域名为中心的防范机制主要指在证书绑定的域名层面对证书应用风险进行防范, 比如将证书与特定域名进行锚定, 通过 DNS 的 CAA (certification authority authorization) 记录指定某个或某些域名只能由特定 CA 签发, 域名所有者在 DNSSEC 条目上指定特定证书, 对域名的证书签发进行备案等. 分为证书锚定、DANE、CT 日志和 CAA 记录这 4 种保障机制.

(1) 证书锚定 (certificate pinning)

证书锚定技术[94,95]允许将网站与特定证书或公钥集相关联, 这些信息可以作为浏览器的一部分包括在内, 也可以在首次访问网站时通过公证查询获得, 锚定方法尝试在客户端检测证书替换. HPKP[96]根据用户的浏览历史创建 Pin. TACK[97]使用 TOFU 方法的服务器 Pin. 谷歌[98]在 Chrome 中为各种域名部署了预装的 Pin. 这些方法可以成功地检测证书更改, 发现可能的 MITM 攻击, 然而, 这种方法是不可扩展的, 第 1 次访问时锚定很容易受到对手的攻击, 这些对手能够在第 1 次访问中向最终用户提供欺诈证书, 此外, 与其他白名单方法一样, 完整性是一个挑战, 特别是考虑到网站的长尾分布.

(2) DANE

基于 DNSSEC 的提议称为基于 DNS 的命名实体身份验证 (DNS-based authentication of named entities, DANE)[99], 使域名所有者能够在 DNSSEC 条目上指定特定证书, 例如用于向其颁发域证书的可接受 CA 列表, 特定可接受证书或特定信任锚验证证书. DANE 允许将证书信息包含在受 DNSSEC 保护的 DNS 服务器中的域名记录中, 除了依赖可信 CA 外, 该技术还取决于域名注册商, DNS 服务器所有者的可信度以及 DNSSEC 的广泛采用, 它使用 DNSSEC 将 TLS 密钥绑定到 DNS 条目[100]. 然而, DANE 如果发挥效用需要绝大多数 DNS 服务器都配置为使用 DNSSEC, 此外, DANE 中的吊销存在问题, 因为在公钥更新的情况下, 包括全球缓存在内的所有 DNS 记录

都应该更新, DANE 的安全在很大程度上依赖于 DNS 运营商的安全.

(3) CT 日志

CT 日志的具体工作原理在第 3 节已经详细阐述, 这里只对安全相关的部分分析. CT 日志提供了已颁发证书的可公开查询日志, 它依赖于最终用户通过浏览器的参与, 网站所有者自愿提交他们正在使用的证书, 以及参与的证书颁发机构在颁发证书时提交证书, 它利用 Merkle 哈希树来有效证明日志的完整性. CT 本身是一个数据源, 可以保证提交给它的任何证书都保持可见和不变, 但将错误颁发证书的检测留给监控器和可信的第三方.

由于 CT 本身并不是为了解决证书撤销问题而设计的, 因此文献 [101] 提出了一个名为撤销透明度的补充系统. 此外, 证书颁发和吊销透明度 (CIRT)[102]提出了 CT 的有效吊销, 但它要求客户端在密钥丢失后创建新身份. Melara 等人[30]于 2015 年提出了 CONIKS, CONIKS 是一个基于透明度日志提案的关键管理系统, 与 CT 类似, 它使用防篡改和可公开审核的目录, 除了 CT, 最终用户密钥也存储在其中, 因此, CONIKS 遇到了诸如密钥撤销等问题.

2018 年, Chrome[103]和 Apple[104]开始要求未来所有可信证书都包含 CT, 为其他浏览器严格执行 CT 铺平了道路. 一些工作已经使用 CT 作为发现钓鱼 DNS 名称的来源[105], 文献 [36] 分析了证书透明度随时间的演变, 并从安全和隐私的角度探讨了公开证书 DNS 名称的含义, 发现 CT 日志中的证书呈指数级增长, 网站对 CT 的支持也在不断增加, 约 33% 的已建立连接支持 CT, 随着 CT 部署的不断增加, 由于 CT 日志中的所有证书都可见, 也存在信息泄露的问题. 文献引入了一个 CT 蜜罐, 并表明 CT 日志中的数据在证书颁发几分钟后就被用来识别扫描活动的目标, 进一步提出并评估了一种从 CT 记录的证书中提取的大量域中学习和验证新子域的方法.

文献 [106] 首次对 CT 监测进行了系统的研究, 分析了 88 个公共日志和 5 个活跃的第三方监控器服务中的数据, 这些数据涉及 6000 个选定的 Alexa Top-1M 网站的 3 亿张证书, 发现尽管 CT 允许普通域所有者充当监控器, 但由于公共日志中证书的数量迅速增加 (例如, 平均 500 万条记录需要监控的最小日志集每天 28.29 GB), 他们自己执行可靠处理是不切实际的. 文献 [106] 的研究揭示了: (1) 没有一个第三方监视器保证返回域的完整证书集; (2) 对于某些域, 第三方监视器返回的证书的并集也可能是不完整的. 因此, 启用 CT 的浏览器所接受的证书对于所声称的域名所有者来说并不是绝对可见的, 即使 CT 与功能良好的日志一起使用也是如此, 公共日志中存在隐形欺诈证书的风险, 这让人们对 CT 在实践中的可靠性产生了怀疑.

(4) CAA 记录

证书颁发机构授权 (CAA)[107]是一项降低 SSL 证书错误颁发的控制措施, 由互联网工程任务组 (IETF) 批准列为 IETF RFC 6844 规范[108]. 2017 年 3 月, CA/Browser 论坛投票通过 187 号提案, 要求 CA 机构从 2017 年 9 月 8 日起执行 CAA 强制性检查, CAA 记录可以控制单域名 SSL 证书的发行, 也可以控制通配符证书. 当域名存在 CAA 记录时, 则只允许在记录中列出的 CA 颁发针对该域名 (或子域名) 的证书.

在域名解析配置中, 可以为整个域 (如 example.com) 或者特定的子域 (如 subzone.example.com) 设置 CAA 策略. 当为整个域设置 CAA 资源记录时, 该 CAA 策略将同时应用于该域名下的任一子域, 除非被已设置的子域策略覆盖. CAA 资源记录允许 DNS 域的合法所有者指定一个或多个 CA 来为该域颁发证书, 因此受损或恶意的 CA 无法为任何域颁发证书.

## 8 未来工作展望

本文系统总结了目前 PKI 技术的发展现状, 重点从安全角度分析了 PKI 系统面临的风险及相应的检测和防范措施的进展情况, 尽管 PKI 检测和防御技术在不断发展, 但是 PKI 安全事件和它们对社会和经济的影响仍然巨大, PKI 体系特别是在安全方面仍有许多问题需要进一步深入研究和分析, 通过对现有方法的总结分析, 未来的研究工作可以关注以下几个方面.

1) PKI 架构的研究

目前的 PKI 系统采用的集中式的管理架构, PKI 的安全性在很大程度上依赖于这些第三方 CA 的可靠性, 而第三方 CA 是 PKI 的单一故障点, 已经发生了几起流行的 CA 违规事件, 其中 CA 的集中操作模式由于流氓证书的传播而引起了许多有针对性的攻击, 目前一种比较重要的研究趋势是使 CA 池完全去中心化, 特别是基于区块

链技术来部署 CA 在近些年引起了学者的关注.

在未来 PKI 架构的研究中, 需要重点关注构建与已建立的 PKI 标准合作的去中心化解决方案, 在去中心化的 CA 组之间提供完全信任, 以实现有效的集成, 同时, 需要提供兼容所有传统的 X.509 标准规定的 PKI 操作, 使其与现有的 PKI 标准兼容, 此外, 还需要考虑攻击者对单个或多个 CA 进行攻击的情况, 使新的架构在对部分 CA 节点被攻陷的情况下仍然能提供有效的证书服务.

2) 证书撤销机制的研究

用户在使用证书前需要验证证书是否有效, 部分证书由于密钥泄露、主体变更、主体与 CA 间的关系变化等因素可能导致证书的撤销. 证书撤销机制是 PKI 系统中保障用户安全通信非常重要的一环, 证书撤销机制设计的好与坏直接影响到 PKI 的规模扩展. 目前关于证书撤销机制的研究大部分侧重于对现有撤销机制的改进上, 比如基于 CRL 的 CRLSet 和 oneCRL 等, 基于 OCSP 的 OCSP Stapling 等, 主要侧重于对证书撤销的性能 (负载, 延迟) 上进行改进.

证书撤销机制在未来仍然是一个值得探索的研究方向. 除了进一步改进完善现有的撤销机制外, 可以从证书撤销参考模型本身进行改进, 同时, 可以针对不同的网络应用场景研究动态选择的证书撤销机制, 以适用不同情境的需求. 另外, 对一些新的证书技术比如嵌套证书的研究可以有效地改进证书的撤销效率, 也可以研究发布生命周期较短的证书来减少甚至消除证书撤销. 此外, 由于撤销延迟, 与之交易的对方尚不能从 CRL 上验证出该证书已经被撤销, 就会认为该证书仍然有效而继续使用. 由于撤销延迟带来的安全风险及防范也是未来值得关注的方向.

3) CT 日志的研究

CT 日志除了在 CT 扩展、CT 部署、CT 测量方面值得继续研究外, CT 日志的部署带来了新的威胁. 首先, 大部分证书被记录到很少的日志中, 这造成了一个脆弱的生态系统; 其次, CT 记录的证书的域名揭示了可能被视为机密或私有的信息, 由于 CT 日志中的所有证书都可见, 因此也存在信息泄露的问题; 第三, 泄露的域名可被恶意用户用于互联网扫描. 因此, 应对 CT 日志带来的风险也是一个值得研究的重要方向.

此外, 对 CT 日志的监视也是一个非常值得研究的方向. 监视器获取所有证书并监视可疑证书, 目前并没有一个第三方监控器保证在域名的公共日志中返回完整的证书集, 如果监控器未能返回特定域名的完整证书集, 则可能无法检测到潜在的欺诈证书, 从而降低 CT 的可靠性. 由于证书的规模和处理它们的复杂性, 加上 CT 本质上是一个分散的系统, 建议设计对策以提高 CT 监测器的可靠性.

4) PKI 系统测量的研究

近些年对 PKI 系统的测量一直是研究热点, 包括证书的部署情况、证书的动态变化情况、证书总体质量情况、交叉签名及带来的风险情况、隐藏的根 CA 的情况等. 这些测量工作暴露了目前 PKI 系统存在的问题, 在很大程度上推进了 PKI 相关技术的发展. 未来对 PKI 系统特别是对系统风险的测量仍然是一个值得研究的热点方向, 特别是针对根 CA, 中间 CA 和叶子证书间信任关系的测量分析, 对客户端, 特别是移动客户端对证书验证机制的测量都需要重点关注, 这是涉及证书验证的关键链条. 证书是否有效一方面依赖不同 CA 间的信任关系, 更加重要的是客户端需要对证书的有效性进行验证.

此外, 对证书的信任锚进行测量也是未来需要重点关注的方向. 安全 TLS 服务器身份验证依赖于可靠的信任锚点, 一个受损的信任锚可以欺骗几乎所有的网络实体, 因此, 需要仔细评估根存储中的每个信任锚, 目前已有工作开始关注这些方面的研究[109], 未来需要围绕 CA 性能和根提供者性能开展工作. 此外, 根程序和 CA 策略/行为的透明度工作将有助于向基于数据的根存储信任过渡.

5) PKI 系统风险检测技术的研究

目前, 证书订户获取 CA 签发的证书后, 没有证书验证环节 (包括对签发的证书本身的安全验证和部署环境的安全验证) 而直接进行部署, 可能导致部署的证书在后续应用过程中存在安全问题, 此外, 证书部署后在应用过程中也可能存在吊销, 过期等安全问题, 而证书订户缺乏在对证书应用过程中的监测. 因此, 除了研究浏览器等客户端对证书的验证外, 需要重点关注服务器对部署的证书的风险检测及自身部署环境是否安全的检测.

证书本身的风险检测需要检测证书的有效性, 包括是否篡改、是否吊销和过期, 证书可能存在的潜在风险, 包

括是否符合 CA/Browser 论坛基线要求、是否进行 CT 备案、证书是否由 CAA 指定的机构颁发、证书链是否正确、证书信任锚是否被通用浏览器根 CA 库信任. 证书部署环境的安全验证需要检测服务器支持的 TLS 版本是否安全、加密套件是否安全、服务器是否开启 HTTP 严格传输安全协议 (HTTP strict transport security, HSTS) 以强制客户端进行 HTTPS 访问, 及 HSTS 参数安全状况等. 由于目前相关研究只有一些论文涵盖所述的部分关注方向并缺乏相关标准化工作, 后续需要关注证书在应用过程中的总体的安全性研究, 特别是重点关注这个方向的国内外标准化建设.

6) PKI 系统风险防范技术的研究

目前业界和学术界针对 PKI 的系统风险提出了一系列风险防范方案. 这些方案一定程度上缓解了目前 PKI 系统面临的风险, 但是目前存在的技术和运营风险仍然在现实中难以避免, 特别是国家间爆发冲突事件时会使得 PKI 系统风险增加.

目前我国重要系统大多部署国外 CA 签发的证书, 且很多重要系统尚未部署证书, 使系统与用户间的安全通信存在重大隐患, 系统中证书资源所有权与管理权不匹配导致所有权存在被单边停签和单边证书吊销风险. 国内外相关工作难以有效解决和应对此类风险, 未来需要研究构建既能兼容现有的认证体系, 同时能在证书出现风险事件时及时感知, 平滑替代现有 CA 认证, 应对证书被恶意撤销造成的影响的 PKI 安全体系.

## 9 总 结

本文从 PKI 的基本工作原理出发, 全面介绍了 PKI 系统在实际部署应用中涉及的各个要素, 包括 PKI 架构、工作流程、证书、证书链、证书撤销、CI 日志服务. 在 PKI 基本工作原理的基础上, 重点从 PKI 系统安全角度全面梳理和总结了 PKI 系统工作过程中面临的安全问题, 包括 PKI 系统应用过程中面临的运营风险和技术风险、PKI 系统的测量、风险检测及各类 PKI 系统的风险防范技术, 由此对未来此领域的研究方向做了一个展望, 希望抛砖引玉, 为从事 PKI 系统领域研究的学者提供参考.

**References:**


[1] Kohnfelder L. Towards a practical public-key cryptosystem [BS. Thesis]. Cambridge: Massachusetts Institute of Technology, 1978.

[2] CA/Browser Forum. Baseline requirements for the issuance and management of publicly-trusted certificates, v.1.1. 2011. https://www.cabforum.org/Baseline_Requirements_V1_1.pdf

[3] Lin JQ, Jing JW, Zhang QL, Wang Z. Recent advances in PKI technologies. Journal of Cryptologic Research, 2015, 2(6): 487–496 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000095]

[4] Clark J, van Oorschot PC. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In: Proc. of the 2013 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2013. 511–525. [doi: 10.1109/SP.2013.41]

[5] Cooper D, Santesson S, Farrell S, Boeyen S, Housley R, Polk T. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. 2008. https://www.rfc-editor.org/rfc/rfc5280.html

[6] Dierks T, Rescorla E. The transport layer security (TLS) protocol version 1.2. 2008. https://www.rfc-editor.org/rfc/rfc5246.html

[7] CA/Browser Forum. Baseline requirements for the issuance and management of publicly-trusted TLS server certificates version 2.0.6. 2024. https://cabforum.org/working-groups/server/baseline-requirements/documents/CA-Browser-Forum-TLS-BR-2.0.6.pdf

[8] Housley R, Ford W, Polk W, Solo D. Internet X.509 public key infrastructure certificate and CRL profile. 1999. https://www.rfc-editor.org/rfc/rfc2459.html

[9] Malone M. Everything you should know about certificates and PKI but are too afraid to ask. 2024. https://smallstep.com/blog/everything-pki/

[10] Hiller J, Amann J, Hohlfeld O. The boon and bane of cross-signing: Shedding light on a common practice in public key infrastructures. In: Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security. New York: ACM, 2020. 1289–1306. [doi: 10.1145/3372297.3423345]

[11] Wahl M, Kille S, Howes T. Lightweight-directory-access-protocol-V3. 1997. https://www.rfc-editor.org/rfc/rfc2251.html

[12] Ellison C, Schneier B. Ten risks of PKI: What you're not being told about public key infrastructure. Computer Security Journal, 2000, 16(1): 1–7. [doi: 10.1201/9780203498156.ch23]

[13] Myers M, Ankney R, Malpani A, Galperin S, Adams C. X.509 Internet public key infrastructure online certificate status protocol-OCSP. 1999. https://www.rfc-editor.org/rfc/rfc2560.html

[14] Pettersen Y. The transport layer security (TLS) multiple certificate status request extension. 2013. https://www.rfc-editor.org/rfc/rfc6961.html [doi: 10.17487/RFC6961]

[15] He B. Improvement and research on mechanism of certificate revocation based on PKI [MS. Thesis]. Shanghai: Shanghai Jiao Tong University, 2015 (in Chinese with English abstract).

[16] Wang J. Design and implementation of certificate status query system based on SM2 [MS. Thesis]. Xi'an: Xidian University, 2015 (in Chinese with English abstract).

[17] Liu YB, Tome W, Zhang L, Choffnes D, Levin D, Maggs B, Mislove A, Schulman A, Wilson C. An end-to-end measurement of certificate revocation in the Web's PKI. In: Proc. of the 2015 Internet Measurement Conf. Tokyo: ACM, 2015. 183–196. [doi: 10.1145/2815675.2815685]

[18] Google group. How CT fits into the wider Web PKI ecosystem. 2013. https://certificate.transparency.dev/howctworks/

[19] Laurie B, Langley A, Kasper E. Certificate transparency. 2013. https://www.rfc-editor.org/rfc/rfc6962.html

[20] Laurie B, Messeri E, Stradling R. Certificate transparency version 2.0. 2021. https://www.rfc-editor.org/rfc/rfc9162.html

[21] Ryan MD. Enhanced certificate transparency and end-to-end encrypted mail. In: Proc. of the 21st Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2014. 1–14.

[22] Szalachowski P, Matsumoto S, Perrig A. PoliCert: Secure and flexible TLS certificate management. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. Scottsdale: ACM, 2014. 406–417. [doi: 10.1145/2660267.2660355]

[23] Basin D, Cremers C, Kim THJ, Perrig A, Sasse R, Szalachowski P. ARPKI: Attack resilient public-key infrastructure. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. Scottsdale: ACM, 2014. 382–393. [doi: 10.1145/2660267.2660298]

[24] Dowling B, Günther F, Herath U, Stebila D. Secure logging schemes and certificate transparency. In: Proc. of the 21st European Symp. on Research in Computer Security on Computer Security. Heraklion: Springer, 2016. 140–158. [doi: 10.1007/978-3-319-45741-3_8]

[25] Chuat L, Szalachowski P, Perrig A, Laurie B, Messeri E. Efficient gossip protocols for verifying the consistency of certificate logs. In: Proc. of the 2015 IEEE Conf. Communications and Network Security. Florence: IEEE, 2015. 415–423. [doi: 10.1109/CNS.2015.7346853]

[26] Nordberg L, Gillmor D, Ritter T. Gossiping in CT. draft-ietf-trans-gossip-05. 2018. https://datatracker.ietf.org/doc/draft-ietf-trans-gossip/05/

[27] Eskandarian S, Messeri E, Bonneau J, Boneh D. Certificate transparency with privacy. Proc. on Privacy Enhancing Technologies, 2017, 2017(4): 329–344. [doi: 10.1515/popets-2017-0052]

[28] Matsumoto S, Szalachowski P, Perrig A. Deployment challenges in log-based PKI enhancements. In: Proc. of the 8th European Workshop on System Security. Bordeaux: ACM, 2015. 1. [doi: 10.1145/2751323.2751324]

[29] Szalachowski P, Chuat L, Perrig A. PKI safety net (PKISN): Addressing the too-big-to-be-revoked problem of the TLS ecosystem. In: Proc. of the 2016 IEEE European Symp. on Security and Privacy. Saarbruecken: IEEE, 2016. 407–422. [doi: 10.1109/EuroSP.2016.38]

[30] Melara MS, Blankstein A, Bonneau J, Felten EW, Freedman MJ. CONIKS: Bringing key transparency to end users. In: Proc. of the 24th USENIX Conf. on Security Symp. Washington: USENIX Association, 2015. 383–398.

[31] Hof B, Carle G. Software distribution transparency and auditability. arXiv:1711.07278, 2017.

[32] Stark E, Sleevi R, Muminovic R, O'Brien D, Messeri E, Felt AP, McMillion B, Tabriz P. Does certificate transparency break the Web? Measuring adoption and error rate. In: Proc. of the 2019 IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2019. 211–226. [doi: 10.1109/SP.2019.00027]

[33] Gustafsson J, Overier G, Arlitt M, Carlsson N. A first look at the CT landscape: Certificate transparency logs in practice. In: Proc. of the 18th Int'l Conf. on Passive and Active Measurement. Sydney: Springer, 2017. 87–99. [doi: 10.1007/978-3-319-54328-4_7]

[34] Nykvist C, Sjöström L, Gustafsson J, Carlsson N. Server-side adoption of certificate transparency. In: Proc. of the 19th Int'l Conf. on Passive and Active Measurement. Berlin: Springer, 2018. 186–199. [doi: 10.1007/978-3-319-76481-8_14]

[35] Amann J, Gasser O, Scheitle Q, Brent L, Carle G, Holz R. Mission accomplished?: HTTPS security after DigiNotar. In: Proc. of the 2017 Internet Measurement Conf. London: ACM, 2017. 325–340. [doi: 10.1145/3131365.3131401]

[36] Scheitle Q, Gasser O, Nolte T, Amann J, Brent L, Carle G, Holz R, Schmidt TC, Wählisch M. The rise of certificate transparency and its implications on the Internet ecosystem. In: Proc. of the 2018 Internet Measurement Conf. Boston: ACM, 2018. 343–349. [doi: 10.1145/3278532.3278562]

[37] Li BY, Chu DW, Lin JQ, Cai QW, Wang CL, Meng LJ. The weakest link of certificate transparency: Exploring the TLS/HTTPS

configurations of third-party monitors. In: Proc. of the 18th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications/the 13th IEEE Int'l Conf. on Big Data Science and Engineering. Rotorua: IEEE, 2019. 216–223. [doi: 10.1109/TrustCom/BigDataSE.2019.00037]

[38] VanderSloot B, Amann J, Bernhard M, Durumeric Z, Bailey M, Halderman JA. Towards a complete view of the certificate ecosystem. In: Proc. of the 2016 Internet Measurement Conf. Santa Monica: ACM, 2016. 543–549. [doi: 10.1145/2987443.2987462]

[39] Aertsen M, Korczyński M, Moura GCM, Tajalizadehkhoob S, van den Berg J. No domain left behind: Is Let's Encrypt democratizing encryption? In: Proc. of the 2017 Applied Networking Research Workshop. Prague: ACM, 2017. 48–54. [doi: 10.1145/3106328.3106338]

[40] Gasser O, Hof B, Helm M, Korczynski M, Holz R, Carle G. In log we trust: Revealing poor security practices with certificate transparency logs and Internet measurements. In: Proc. of the 19th Int'l Conf. on Passive and Active Measurement. Berlin: Springer, 2018. 173–185. [doi: 10.1007/978-3-319-76481-8_13]

[41] Cui MX, Cao ZG, Xiong G. How is the forged certificates in the wild: Practice on large-scale SSL usage measurement and analysis. In: Proc. of the 18th Int'l Conf. on Computational Science. Wuxi: Springer, 2018. 654–667. [doi: 10.1007/978-3-319-93713-7_62]

[42] HTTPS encryption on the Web—Google transparency report. 2024. https://transparencyreport.google.com/?hl=en

[43] Holz R, Braun L, Kammenhuber N, Carle G. The SSL landscape: A thorough analysis of the X.509 PKI using active and passive measurements. In: Proc. of the 2011 ACM SIGCOMM Conf. on Internet Measurement Conf. Berlin: ACM, 2011. 427–444. [doi: 10.1145/2068816.2068856]

[44] Chung T, Liu YB, Choffnes D, Levin D, Maggs BM, Mislove A, Wilson C. Measuring and applying invalid SSL certificates: The silent majority. In: Proc. of the 2016 Internet Measurement Conf. Santa Monica: ACM, 2016. 527–541. [doi: 10.1145/2987443.2987454]

[45] Georgiev M, Iyengar S, Jana S, Anubhai R, Boneh D, Shmatikov V. The most dangerous code in the world: Validating SSL certificates in non-browser software. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. Raleigh North: ACM, 2012. 38–49. [doi: 10.1145/2382196.2382204]

[46] Fahl S, Harbach M, Muders T, Baumgärtner L, Freisleben B, Smith M. Why eve and mallory love Android: An analysis of Android SSL (in) security. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. Raleigh North: ACM, 2012. 50–61. [doi: 10.1145/2382196.2382205]

[47] Jarmoc J. SSL/TLS interception proxies and transitive trust. 2012. https://media.blackhat.com/bh-eu-12/Jarmoc/bh-eu-12-Jarmoc-SSL_TLS_Interception-WP.pdf

[48] Appelbaum J. Detecting certificate authority compromises and Web browser collusion. 2011. http://blog.torproject.org/detecting-certificate-authority-compromises-and-web-browser-collusion

[49] Hoogstraaten H. Black tulip report of the investigation into the DigiNotar certificate authority breach. 2012. https://www.researchgate.net/publication/269333601_Black_Tulip_Report_of_the_investigation_into_the_DigiNotar_Certificate_Authority_breach [doi: 10.13140/2.1.2456.7364]

[50] Mozilla Wiki. CA: WoSign issues. 2015. https://wiki.mozilla.org/CA/WoSign_Issues

[51] Roberts R, Goldschlag Y, Walter R, Chung T, Mislove A, Levin D. You are who you appear to be: A longitudinal study of domain impersonation in TLS certificates. In: Proc. of the 2019 ACM SIGSAC Conf. Computer and Communications Security. London: ACM, 2019. 2489–2504. [doi: 10.1145/3319535.3363188]

[52] Watchguard Internet Security Report: Q4 2022. 2022. https://www.watchguard.com/wgrd-resource-center/security-report-q4-2022

[53] Durumeric Z, Kasten J, Bailey M, Halderman JA. Analysis of the HTTPS certificate ecosystem. In: Proc. of the 2013 Conf. on Internet Measurement Conf. Barcelona: ACM, 2013. 291–304. [doi: 10.1145/2504730.2504755]

[54] Kotzias P, Razaghpanah A, Amann J, Paterson KG, Vallina-Rodriguez N, Caballero J. Coming of age: A longitudinal study of TLS deployment. In: Proc. of the 2018 Internet Measurement Conf. Boston: ACM, 2018. 415–428. [doi: 10.1145/3278532.3278568]

[55] Holz R, Hiller J, Amann J, Razaghpanah A, Jost T, Vallina-Rodriguez N, Hohlfeld O. Tracking the deployment of TLS 1.3 on the Web: A story of experimentation and centralization. ACM SIGCOMM Computer Communication Review, 2020, 50(3): 3–15. [doi: 10.1145/3411740.3411742]

[56] CA/Browser Forum. Baseline requirements for the issuance and management of policy-trusted certificates, v1.1.5. 2013. https://cabforum.org/uploads/Baseline_Requirements_V1_1_5.pdf

[57] Delignat-Lavaud A, Abadi M, Birrell A, Mironov I, Wobber T, Xie YL. Web PKI: Closing the gap between guidelines and practices. In: Proc. of the 21st Annual Network and Distributed System Security Symp. San Diego: Internet Society, 2014.

[58] Dong Z, Kane K, Camp LJ. Detection of rogue certificates from trusted certificate authorities using deep neural networks. ACM Trans. on Privacy and Security (TOPS), 2016, 19(2): 5. [doi: 10.1145/2975591]

[59] Brubaker C, Jana S, Ray B, Khurshid S, Shmatikov V. Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 114–129. [doi: 10.1109/SP.2014.15]

[60] Chau SY, Chowdhury O, Hoque E, Ge HY, Kate A, Nita-Rotaru C, Li NH. SymCerts: Practical symbolic execution for exposing noncompliance in X.509 certificate validation implementations. In: Proc. of the 2017 IEEE Symp. on Security and Privacy. San Jose: IEEE, 2017. 503–520. [doi: 10.1109/SP.2017.40]

[61] de Carné de Carnavalet X, Mannan M. Killed by proxy: Analyzing client-end TLS interception software. In: Proc. of the 2016 Network and Distributed System Security Symp. San Diego: Internet Society, 2016.

[62] Waked L, Mannan M, Youssef A. To intercept or not to intercept: Analyzing TLS interception in network appliances. In: Proc. of the 2018 on Asia Conf. on Computer and Communications Security. Incheon: ACM, 2018. 399–412. [doi: 10.1145/3196494.3196528]

[63] Durumeric Z, Ma Z, Springall D, Barnes R, Sullivan N, Bursztein E, Bailey MD, Halderman JA, Paxson V. The security impact of HTTPS interception. In: Proc. of the 24th Annual Network and Distributed System Security Symp. San Diego: Internet Society, 2017.

[64] Zhang L, Choffnes D, Dumitraş T, Levin D, Mislove A, Schulman A, Wison C. Analysis of SSL certificate reissues and revocations in the wake of heartbleed. In: Proc. of the 2014 Conf. on Internet Measurement Conf. Vancouver: ACM, 2014. 489–502. [doi: 10.1145/2663716.2663758]

[65] Huang LS, Rice A, Ellingsen E, Jackson C. Analyzing forged SSL certificates in the wild. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 83–97. [doi: 10.1109/SP.2014.13]

[66] O'Neill M, Ruoti S, Seamons K, Zappala D. TLS proxies: Friend or foe? In: Proc. of the 2016 Internet Measurement Conf. Santa Monica: ACM, 2016. 551–557. [doi: 10.1145/2987443.2987488]

[67] Zhang YM, Liu BJ, Lu CY, Li Z, Duan HX, Li JC, Zhang ZF. Rusted anchors: A national client-side view of hidden root CAs in the Web PKI ecosystem. In: Proc. of the 2021 ACM SIGSAC Conf. on Computer and Communications Security. ACM, 2021. 1373–1387. [doi: 10.1145/3460120.3484768]

[68] Ma Z, Mason J, Antonakakis M, Durumeric Z, Bailey MD. What's in a name? Exploring CA certificate control. In: Proc. of the 30th USENIX Security Symp. USENIX Association, 2021. 4383–4400.

[69] Birge-Lee H, Sun YX, Edmundson A, Rexford J, Mittal P. Bamboozling certificate authorities with BGP. In: Proc. of the 27th USENIX Conf. Security Symp. Baltimore: USENIX Association, 2018. 833–849.

[70] Wendlandt D, Andersen DG, Perrig A. Perspectives: Improving SSH-style host authentication with multi-path probing. In: Proc. of the 2008 USENIX Annual Technical Conf. Boston: USENIX Association, 2008. 321–334.

[71] Marlinspike M. Convergence. 2012. http://convergence.io

[72] Alicherry M, Keromytis AD. DoubleCheck: Multi-path verification against man-in-the-middle attacks. In: Proc. of the 2009 IEEE Symp. on Computers and Communications. Sousse: IEEE, 2009. 557–563. [doi: 10.1109/ISCC.2009.5202224]

[73] DetecTor. 2009. http://www.detector.io

[74] The ICSI certificate notary. 2015. https://icsi.berkeley.edu/icsi/node/5065

[75] EFF. The EFF SSL observatory. 2015. https://www.eff.org/Observatory

[76] Emre Y, Ali AS. Server notaries: A complementary approach to the Web PKI trust model. IET Information Security. 2018, 12: 455–461. [doi: 10.1049/iet-ifs.2016.0611]

[77] Wazan AS, Laborde R, Barrère F, Benzekri A. The X.509 trust model needs a technical and legal expert. In: Proc. of the 2012 IEEE Int'l Conf. on Communications. Ottawa: IEEE, 2012. 6895–6900. [doi: 10.1109/ICC.2012.6364860]

[78] ITU X.509. Information technology-open systems interconnection-the directory: Public-key and attribute certificate frameworks. 2019. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=13031

[79] Samer WA, Romain L, Francois B, AbdelMalek B. A formal model of trust for calculating the quality of X.509 certificate. Security and Communication Networks, 2011, 4(6): 651–665. [doi: 10.1002/sec.198]

[80] Wazan AS, Laborde R, Chadwick DW, Barrere F, Benzekri A, Kaiiali M, Habbal A. Trust management for public key infrastructures: Implementing the X.509 trust broker. Security and Communication Networks, 2017, 2017(1): 6907146. [doi: 10.1155/2017/6907146]

[81] Freeman T, Housley R, Malpani A, Cooper D, Polk W. Server-based certificate validation protocol (SCVP). 2007. https://www.rfc-editor.org/rfc/rfc5055.html

[82] Abadi M, Birrell A, Mironov I, Wobber T, Xie YL. Global authentication in an untrustworthy world. In: Proc. of the 14th USENIX Conf. on Hot Topics in Operating Systems. Santa Ana Pueblo: USENIX Association, 2013. 19.

[83] Yee P. Updates to the Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. 2013. https://www.rfc-editor.org/rfc/rfc6818.html

[84] Eastlake III D. Transport layer security (TLS) extensions: Extension definitions. 2011. https://www.rfc-editor.org/rfc/rfc6066.html

[85] Rivest RL. Can we eliminate certificate revocation lists? In: Proc. of the 2nd Int'l Conf. on Financial Cryptography. Anguilla: Springer, 1998. 178–183. [doi: 10.1007/BFb0055482]

[86] Levi A, Caglayan MU, Koc CK. Use of nested certificates for efficient, dynamic, and trust preserving public key infrastructure. ACM Trans. on Information and System Security (TISSEC), 2004, 7(1): 21–59. [doi: 10.1145/984334.984336]

[87] Allen C, Brock A, Buterin V. Decentralized public key infrastructure. A white paper from rebooting the Web of trust. 2015. https://raw.githubusercontent.com/WebOfTrustInfo/rwot1-sf/master/final-documents/dpki.pdf

[88] Wang Z, Lin JQ, Cai QW, Wang QX, Zha DR, Jing JW. Blockchain-based certificate transparency and revocation transparency. IEEE Trans. on Dependable and Secure Computing, 2022, 19(1): 681–697. [doi: 10.1109/TDSC.2020.2983022]

[89] Kubilay MY, Kiraz MS, Mantar HA. CertLedger: A new PKI model with certificate transparency based on blockchain. Computers & Security, 2019, 85: 333–352. [doi: 10.1016/j.cose.2019.05.013]

[90] Chen J, Yao SX, Yuan Q, He K, Ji SL, Du RY. CertChain: Public and efficient certificate audit based on blockchain for TLS connections. In: Proc. of the 2018 IEEE Conf. on Computer Communications. Honolulu: IEEE, 2018. 2060–2068. [doi: 10.1109/INFOCOM.2018.8486344]

[91] Yan JZ, Yang B, Su L, He S. Storage optimization for certificates in blockchain based PKI system. In: Proc. of the 3rd CCF China Blockchain Conf. on Blockchain Technology and Application. Jinan: Springer, 2021. 116–125. [doi: 10.1007/978-981-33-6478-3_8]

[92] Zhai ZH, Shen SB, Mao YQ. BPKI: A secure and scalable blockchain-based public key infrastructure system for Web services. Journal of Information Security and Applications, 2022, 68: 103226. [doi: 10.1016/j.jisa.2022.103226]

[93] Saleem T, Janjua MU, Hassan M, Ahmad T, Tariq F, Hafeez K, Salal MA, Bilal MD. ProofChain: An X.509-compatible blockchain-based PKI framework with decentralized trust. Computer Networks, 2022, 213: 109069. [doi: 10.1016/j.comnet.2022.109069]

[94] Public key pinning. 2011. http://www.imperialviolet.org/2011/05/04/pinning.html

[95] Evans C, Palmer C. Public key pinning extension for HTTP. draft-ietf-websec-key-pinning-01. 2011. https://datatracker.ietf.org/doc/draft-ietf-websec-key-pinning-01

[96] Evans C, Palmer C, Sleevi R. Public key pinning extension for HTTP. 2015. https://www.rfc-editor.org/rfc/rfc7469.html

[97] Kranch M, Bonneau J. Upgrading HTTPS in mid-air: An empirical study of strict transport security and key pinning. In: Proc. of the 22nd Annual Network and Distributed System Security Symp. San Diego: Internet Society, 2015.

[98] Marlinspike M. Trust assertions for certificate keys. draft-perrin-tls-tack-02.txt. 2013. https://datatracker.ietf.org/doc/draft-perrin-tls-tack/02/

[99] Hoffman P, Schlyter J. The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA. 2012. https://www.rfc-editor.org/rfc/rfc6698.html

[100] Dukhovni V, Hardaker W. The DNS-based authentication of named entities (DANE) protocol: Updates and operational guidance. 2015. https://www.rfc-editor.org/rfc/rfc7671.html

[101] Laurie B, Kasper E. Revocation transparency. 2012. https://www.links.org/files/ RevocationTransparency.pdf

[102] Ryan MD. Enhanced certificate transparency and end-to-end encrypted mail. 2013. https://markryan.eu/research/papers/pdf/14-ndss-cert.pdf

[103] Sleevi R. Certificate transparency in Chrome—Change to enforcement date. 2017. https://groups.google.com/a/chromium.org/g/ct-policy/c/sz_3W_xKBNY/m/6jq2ghJXBAAJ

[104] Apple. Apple's certificate transparency policy. 2024. https://support.apple.com/en-us/HT205280

[105] Drury V, Meyer U. Certified phishing: Taking a look at public key certificates of phishing websites. In: Proc. of the 15th USENIX Conf. on Usable Privacy and Security. Santa Clara: USENIX Association, 2019. 211–223.

[106] Li BY, Lin JQ, Li FJ, Wang QX, Li Q, Jing JW, Wang CL. Certificate transparency in the wild: Exploring the reliability of monitors. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 2505–2520. [doi: 10.1145/3319535.3345653]

[107] Hallam-Baker P, Stradling R, Hoffman-Andrews J. DNS certification authority authorization (CAA) resource record. 2019. https://www.rfc-editor.org/rfc/rfc8659.html

[108] Hallam-Baker P, Stradling R. DNS certification authority authorization (CAA) resource record. 2013. https://www.rfc-editor.org/rfc/rfc6844.html

[109] Ma Z, Austgen J, Mason J, Durumeric Z, Bailey M. Tracing your roots: Exploring the TLS trust anchor ecosystem. In: Proc. of the 21st ACM Internet Measurement Conf. ACM, 2021. 179–194. [doi: 10.1145/3487552.3487813]

**附中文参考文献**:

[3] 林璟锵, 荆继武, 张琼露, 王展. PKI 技术的近年研究综述. 密码学报, 2015, 2(6): 487–496. [doi: 10.13868/j.cnki.jcr.000095]

[15] 何斌. PKI 中证书撤销机制的改进与研究 [硕士学位论文]. 上海: 上海交通大学, 2015.

[16] 王杰. 基于商密 SM2 算法的证书状态查询机制设计与实现 [硕士学位论文]. 西安: 西安电子科技大学, 2015.

**张宾**(1976－), 男, 博士, 高级工程师, 博士生导师, 主要研究领域为网络安全, 网络管理, 数据分析.

**乔延臣**(1988－), 男, 博士, 副研究员, 博士生导师, CCF 专业会员, 主要研究领域为网络空间安全, 互联网体系结构.

**张宇**(1979－), 男, 博士, 教授, 主要研究领域为互联网基础设施安全, 网络拓扑测量, 未来网络体系.

**刘翔**(1995－), 男, 博士, 助理研究员, 主要研究领域为密码学, 网络安全.

**张伟哲**(1976－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为信息安全, 系统结构.

**刘鹏辉**(1982－), 男, 高级工程师, 主要研究领域为现代密码学, 网络与信息安全, 侧信道分析, 系统安全.