密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research,2015,2(6):487–496

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-81033101

PKI 技术的近年研究综述*

林璟锵[1,2], 荆继武[1,2], 张琼露[1,2], 王 展[1,2]

1. 中国科学院数据与通信保护研究教育中心, 北京 100093
2. 中国科学院信息工程研究所, 北京 100093
通讯作者: 林璟锵, E-mail: linjingqiang@iie.ac.cn

摘 要: 公钥基础设施(Public Key Infrastructure, PKI)是典型的密码应用技术. 在 PKI 系统中, 由证书认证机构(Certification Authority, CA)签发数字证书、绑定 PKI 用户的身份信息和公钥. PKI 依赖方(Relying Party)预先存储有自己所信任的根 CA 自签名证书, 用来验证与之通信的 PKI 用户的证书链, 从而可信地获得该用户的公钥、用于各种安全服务. 近 5 年来, 随着 PKI 系统的深入应用, 围绕各种应用场景、出现了新的技术研究成果, 主要包括: SSL/TLS 协议过程中的证书验证和证书管理、PKI 系统的大规模实施部署、以及新的证书撤销方案. 首先, 在 SSL/TLS 协议的相关研究上, 主要包括了客户端证书验证漏洞而导致的中间人攻击和相应解决方案; Certificate Transparency 技术及其改进, 则是考虑了被攻击 CA 签发虚假网站证书的威胁, 公开地审计 CA 的证书签发过程、及时发现虚假证书; 此外, 通过依赖方客户端的 CA 证书管理, 也可以有效降低 CA 被攻击情况下的危害. 其次, PKI 系统的大规模实施部署研究, 主要包括跨国/跨域互操作、ICAO 电子护照、互联网路由安全、互联网 DNS 安全等应用场景. 第三, 近年来的证书撤销相关研究集中在特定需求场景(RFID、电子护照、密钥托管和浏览器隐身模式等)的方案设计和分析. 本文对上述 PKI 技术研究进展进行了详细的分析和总结.

关键词: 公钥基础设施; 数字证书; SSL/TLS
中图法分类号: TP309.7 **文献标识码**: A DOI: 10.13868/j.cnki.jcr.000095

中文引用格式: 林璟锵, 荆继武, 张琼露, 王展. PKI 技术的近年研究综述[J]. 密码学报, 2015, 2(6): 487–496.
英文引用格式: Lin J Q, Jing J W, Zhang Q L, Wang Z. Recent advances in PKI technologies[J]. Journal of Cryptologic Research, 2015, 2(6): 487–496.

Recent Advances in PKI Technologies

LIN Jing-Qiang[1,2], JING Ji-Wu[1,2], ZHANG Qiong-Lu[1,2], WANG Zhan[1,2]

1. Data Assurance and Commnication Security Research Centre, Chinese Academy of Sciences, Bejing100093, China
2. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
Corresponding author: LIN Jing-Qiang, E-mail: linjingqiang@iie.ac.cn

Abstract: The public key infrastructure (PKI) is a typical technology of applied cryptography. In a PKI system, digital certificates are signed by certification authorities (CAs) to bind a PKI user's identity and public key. Then, using the trusted root CA's self-signed certificate, a PKI relying-party verifies the certificate chain of the PKI user,

* 基金项目: 国家重点基础研究发展项目(973 计划)(2013CB338001)
收稿日期: 2015-09-07 定稿日期: 2015-10-03

to obtain the PKI user's public key for various security services. In the recent five years, PKI technologies achieved remarkable progress as PKI systems are widely adopted in various scenarios, including: (1) certificate verification and management in SSL/TLS, (2) PKI systems deployed in large-scale applications, and (3) certification revocation solutions. Firstly, about the PKI research related to SSL/TLS, there exist man-in-the-middle attacks due to the vulnerability of certificate verification and the countermeasures; certificate transparency and its follow-ups publicly audit CAs' certificate signing operations, to detect the fake certificates signed by compromised CAs; CA certificate management in PKI clients, is also useful to mitigate the threats from compromised CAs. Secondly, PKI systems are deployed in the large-scale scenarios of cross-border/domain interoperability, ICAO ePassport, IP routing security and DNS security. Finally, recent certificate revocation solutions focus on the applications with special requirements such as RFID, ePassports, key escrow systems and private-mode browsers. This paper surveys the advances of these PKI technologies.

Key words: PKI; Certificate; SSL/TLS

1 引言

公钥基础设施(Public Key Infrastructure, PKI)是基于公钥密码学算法的安全基础服务设施. 在 PKI 中, 由证书认证中心(Certification Authority, CA) 签发数字证书(简称证书) , 绑定用户的身份信息和公钥. 在通信过程中, 证书依赖方(Relying Party)获得通信对方的证书链, 然后利用自身配置存储的根 CA 自签名证书来逐一地验证证书链中的各张证书, 可信地获得通信对方的公钥, 从而用于机密性、数据完整性、身份鉴别、非否认等各种安全功能.

1978 年, L. Kohnfelder 首次提出证书的概念[1]; 1988 年, 第一版本的 X.509 标准推出, 发展至 2005 年的版本 3 标准; 1995 年, IETF 成立 PKIX 工作组, 将 X.509 标准用于 Internet, 2013 年, IETF PKIX 工作组结束工作任务. 经过多年的技术研究, PKI 技术已经有了长足的进展、广泛的应用, 在全球的信息系统中发挥了重要的安全支撑作用.

虽然 PKI 技术已经较为成熟、进入大规模应用的阶段, 但是随着在实际应用系统中的大量部署、各种重要应用领域采纳 PKI 技术, 近 5 年来继续出现了大量全新的 PKI 技术研究成果, 克服应用推广部署中的种种难题. 近 5 年来的 PKI 技术研究进展, 主要包括了如下几个方面:

(1) SSL/TLS 协议[2]是目前实施最为广泛的 PKI 技术应用. 研究表明, SSL/TLS 协议中的证书验证安全漏洞大量存在于各种系统中[3–5], 会导致危害严重的中间人攻击; 相关的攻击检测、防护技术成果随之被提出. 2013 年, Google 公司提出的 Certificate Transparency 技术被正式采纳为 IETF RFC 6962[6], 用于提升 SSL/TLS 协议的服务器证书的可信程度; 在 Certificate Transparency 技术基础上, 又有多种改进方案出现[7–10]. 作为证书依赖方, SSL/TLS 协议客户端(通常是浏览器)的根 CA 证书配置直接影响到 SSL/TLS 服务器证书的验证结果; 各种改进的 CA 证书配置管理方案[11–13], 可用于限制恶意 CA 的攻击效果.

(2) 如前文所述, PKI 技术已经进入大规模应用阶段; 随着 PKI 系统在全球电子护照、DNS 系统、HTTPS 服务器等信息服务中的部署, 研究人员总结了多种全新的技术挑战、也提出了相应的解决方案[14–20].

(3) 长期以来, 证书撤销方案都是 PKI 技术研究的重要内容. 由于不存在普遍最优适用的证书撤销方案, 针对不同的应用场景, 各种证书撤销方案各有优势和不足. 所以, 针对新型应用场景的证书撤销方案、基于新型技术的证书撤销方案[19–22], 近年来仍然不断有相关成果完成.

2 数字证书与 SSL/TLS 协议

SSL/TLS 协议是 PKI 技术的最重要应用. 近年来, 随着 SSL/TLS 协议的大面积使用, 例如 Google 在搜索服务中全面启用 SSL/TLS 支持、大量网站默认使用 HTTPS 安全通信、各种云计算服务的网络连接使用 SSL/TLS 协议等, 协议使用范围不断扩展, 直接关系到越来越多的应用系统. SSL/TLS 协议相关的漏洞和攻击频繁出现, 其安全性受到了更加广泛的关注. 服务器证书验证是 SSL/TLS 协议中的必要步骤[2], 直接关系到 SSL/TLS 协议能否正确地发挥安全作用; 近年来有大量 PKI 技术研究进展与此密切相关.

首先, 为了保证证书的正确性, 在 SSL/TLS 的服务器证书验证过程中, 需要正确地执行证书链验证和域名检查. 其次, 分散和限制 CA 在证书签发和发布过程中的权力, 从而避免因 CA 被攻击而导致包含虚假信息但可验证成功的服务器证书(例如, 近年来 TurkTrust、DigiNotar、Comodo 等 CA 公司的安全事件). 具体地, 可在证书签发和发布过程中引入额外的部件、或者是由依赖方来实施限制.

2.1 SSL/TLS协议的数字证书验证和中间人攻击

文献[3]和文献[4]分别研究分析了 SSL/TLS 软件开发包和 Android 程序中的服务器证书验证漏洞, 该漏洞导致中间人攻击问题. 在 SSL/TLS 协议中, 客户端会在线地接收服务器证书, 在正确地验证该证书之后, 使用证书中的公钥与服务器进行密钥协商, 以保护后续的数据通信. 如果客户端不能正确地验证服务器证书, 则攻击者可以伪装成合法的服务器, 使用自己的非法证书与客户端建立 SSL/TLS 连接, 从而窃听或篡改客户端与合法服务器之间的通信数据. 分析结果表明[3], 在大量的 SSL/TLS 软件中, 都未能正确地检查(1)所接收的服务器证书链是否由可信的根 CA 所签发、和/或(2)所接收的服务器证书中的 DNS 域名与合法服务器的 DNS 域名是否一致. 基于 Android SDK 开发的大量 Android 程序, 在建立 SSL/TLS 安全通道时, 同样也没有检查服务器证书是否由可信的根 CA 所签发、和/或不检查证书中的 DNS 域名; 在 2012 年的研究分析表明[4], 从 Google Play Market 下载的 13500 个 Android APP 中, 8%的 APP 存在上述安全问题. 对于混合使用 Native UI 和 WebView UI 的 Android APP, 在使用 Android 系统 WebView 功能来实现 HTTPS 通信时, 也会有类似安全问题[5]: 虽然 WebView 能正确地检测服务器证书错误, 但是仍有大量 Android APP 直接忽略该错误、并继续执行功能.

上述文献分析发现的、证书验证漏洞导致的 SSL/TLS 中间人攻击问题, 引起了更深入的后续研究.

(1) SVM-Hunter 系统[23]完成了静态分析和动态分析相结合的、针对 Android APP 的自动化检测: 先使用静态分析来定位 APP 中的证书验证功能部分, 然后使用动态方式触发 APP 中的证书验证功能, 从而判定是否存在漏洞. SVM-Hunter 系统使用混合方式, 能够快速(静态分析的优势) 并准确(动态分析的优势) 地检测 APP 中的证书验证漏洞.

(2) SSLINT 系统[24]采取静态分析机制, 将正确的 SSL/TLS 协议处理流程和被检测程序源代码都分别表示为程序依赖图(Program Dependence Graph), 进而通过程序依赖图的匹配来检测 SSL/TLS 客户端程序中的安全漏洞(主要是服务器证书链验证和 DNS 域名检查).

(3) CertShim 方案[25]是服务器证书验证漏洞的补丁工具. 针对基于 OpenSSL 和 GnuTLS 开发的程序, CertShim 使用 API Hook 技术, 嵌入正确的服务器证书验证和域名检查步骤, 使得能够在不更改原有软件程序的前提下, 快速修复漏洞.

(4) Crossbear 系统[26]通过比较在不同位置的 SSL/TLS 客户端收到的服务器证书, 定位中间人攻击者的位置. 该系统包括了大量的分布式探测节点(作为 SSL/TLS 客户端), 不同节点分别与相同的 Web 服务器建立 SSL/TLS 连接, 然后由中心服务器比较多个节点的 IP 路径和所收到的服务器证书(有的是正确的服务器证书、有的则是中间人攻击者的证书), 就可以推断中间人攻击的插入位置. Crossbear 系统的探测节点越多, 则定位精度越高.

文献[27]从另一角度分析了 SSL/TLS 服务器证书问题: 现有不少合法的 Web 服务器, 在有意或无意地

使用无法被浏览器正常验证的服务器证书, 例如, 使用自签名证书、不能由浏览器预装的根 CA 证书来验证、证书链配置不完整、域名不准确、证书过期等等. 上述的“合法”服务器, 在使用过程中, 与中间人攻击难以区分. 事实上, 由于上述“合法”情况的存在, 使得 SSL/TLS 客户端的证书验证难以采取非常严格的检查措施(例如, 在服务器证书验证出错时, 直接中断 SSL/TLS 会话、而不是警告). 调研表明[27], 以上问题的部分原因是 Web 管理员的配置失误、部分则是因为 Web 管理员不愿意从商业 CA 公司申请证书(主要是价格考虑). 所以, 提升 SSL/TLS 协议通信的安全性, 不仅仅需要客户端软件的改进, 也需要 Web 服务器管理员的配合.

在 SSL/TLS 协议中, 客户端也可以有自己的证书/密钥对, 服务器也会接收客户端的证书、验证客户端是否持有相应的私钥(可选步骤). 在很多应用场景中, SSL/TLS 协议并不执行客户端证书验证步骤, 因为用户身份鉴别一般都是在 SSL/TLS 会话建立之后、再由应用层协议(如 HTTP 协议)来完成. TLS-OBC 方案[28]利用客户端证书验证步骤来绑定上层应用凭证与会话层: 每次与特定的应用服务器建立 SSL/TLS 连接时, 浏览器实时地产生新的自签名证书, 与服务器执行客户端证书验证步骤, 再由服务器将该证书绑定上层应用凭证(如 HTTP Cookie), 从而能够抵抗凭证泄露的漏洞和攻击. 文献[29]对现有 SSL/TLS 协议实现中的客户端证书验证, 进行了全面深入的分析, 包括: 该步骤执行导致的服务器的配置信息泄露、用户隐私泄露和对服务器的 DoS 攻击, 以及客户端签名证据的及时性、证书撤销检查、客户端浏览器支持等等.

此外, 文献[30]提出了在 HTTPS(即 HTTP over SSL/TLS)协议中, 由浏览器自动地分析网页链接来预先获取和验证服务器证书, 以提高协议执行速度、改善用户体验. 文献[31]对 SSL/TLS 和 HTTPS 中服务器证书的信任和安全, 有全面深入的总结.

2.2 基于Certificate Transparency技术的数字证书服务

对于 SSL/TLS 服务器证书, 另一重大安全隐患是被攻击或恶意 CA 所导致的: 如果依赖方信任的 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, 则中间人攻击就很容易成功(即使客户端执行了正确的证书验证和域名检查步骤). TurkTrust CA 公司的虚假信息证书、DigiNotar CA 公司被攻击、Comodo CA 公司安全事件等等, 都显示此类攻击完全可能发生. 针对上述问题, Google 公司于 2013 年完成了 RFC 6962 Certificate Transparency[6], 引入独立运行的 Log Server 部件, 要求 CA 在签发了服务器证书之后、及时地将其发布到公开的 Log Server 上, 相应的服务器证书在 SSL/TLS 通信过程中才会被浏览器接受; Log Server 以 Merkle Tree 的形式来存储服务器证书, 保持 Append-Only 特性、只增不删. 另有 Auditor 部件(可以是 SSL/TLS 客户端或者独立的服务器)检查 Log Server 上存储的记录, 确认 Log Server 的证书存储服务是 Append-Only.

Certificate Transparency 技术的基本思路是: 使 CA 的证书签发服务成为可公开审计的操作, 只有已经被审计的服务器证书(即已经存储在 Log Server)才会被 SSL/TLS 客户端接受. Certificate Transparency 技术也引起了大量的后续研究. 在 Certificate Transparency 方案基础上, 文献[7,10]完成了对证书撤销操作的公开审计, 在 Log Server 上维护可审计的证书撤销状态信息.

PoliCert 方案[8]又进一步限制了 CA 在 SSL/TLS 服务器证书签发过程中的权力: (1) 要求服务器证书上有多个不同 CA 的签名, 单个 CA 不能完成服务器证书的签发操作; (2) 由证书持有者(也就是 Web 服务器)以类似于证书的形式(称为 Subject Certificate Policy), 公开发布自己的证书策略, 其中包括可给该 DNS 域名签发证书的 CA 列表、证书中应该包含的 CA 签名数量等信息. 同时, 要求服务器证书和 Subject Certificate Policy 都存储在 Log Server 上, 可公开审计. 相比已有 PKI 系统, 在 PoliCert 方案中, 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. 相比 Accountable Key Infrastructure 方案[32], PoliCert 方案使用独立的 Subject Certificate Policy 来发布证书持有者的证书策略、而不是直接包含在服务器证书之中, 就使得 Web 服务器可以同时有多张有效的服务器证书、更新更为灵活.

ARPKI 方案[9]从另一角度扩展了 Certificate Transparency 技术: (1) 类似于 PoliCert 方案, 服务器证书上

要求有多个不同 CA 的签名; (2) 由多个 CA 协作来多次执行 Auditor 部件的功能, 确认证书已经发布到 Log Server 上. 相比原有提出的、由 CA 的多个内部部件完成门限签名的证书服务方案[33–35], PoliCert 方案、ARPKI 方案和 Accountable Key Infrastructure 方案中的证书同样也需要有多次密码计算才能完成, 不同的是上述方案都是以公开可见的方式来执行多次签名, 而且门限参数是由证书持有者自主控制的、每张证书可以不相等.

2.3 PKI依赖方的CA证书管理和信任管理

Certificae Transparency 及其扩展增强方案, 都是在证书签发阶段, 由 Log Server、证书持有者介入来分散部分权力. 下面, 本节介绍了由证书依赖方(对于 HTTPS 协议, 也就是浏览器)实施的、对于 CA 权力的限制措施, 包括减少根 CA 自签名证书数量[36]、降低对根 CA 的信任程度[11]、限制 CA 的服务器证书服务范围[12,13]、在证书验证过程中辅以口令验证[37]等等.

在操作系统/浏览器中, 通常预置了大量的根 CA 证书, 只要服务器证书链能够由其中任意某一根 CA 证书验证, 则 SSL/TLS 连接就能够正常建立. 文献[36]分析了现有 11 种主流操作系统/浏览器所预置存储的根 CA 自签名证书, 发现在所有 431 张根 CA 证书中、有约 34%的根 CA 并不直接或间接地签发 HTTPS 服务器. 过多的、不必要的根 CA 证书给攻击者留下可乘之机: 攻击者只要能够成功入侵其中安全强度最低的某一个 CA, 就能够随意地签发所有域名的服务器证书, 顺利地建立 SSL/TLS 连接、实施中间人攻击.

Certlock 机制[13]和 CAge 方案[12]分别提出: 在客户端浏览器上, 设定各服务器证书只能由满足特定条件的 CA 来签发; 或者, 设定各 CA 只能给满足特定条件的 Web 服务器签发证书. Certlock 机制[13]通过检查签发证书的 CA 所属国家, 来限制恶意 CA 的攻击影响: 浏览器在第一次访问 Web 服务器时(假定此时没有攻击行为), 记录签发该服务器证书的 CA 所属国家; 要求在之后的所有 SSL/TLS 协商中, 该 Web 服务器的证书不能由来自其它国家的 CA 签发, 否则就给出警告. Certlock 机制假定 Web 服务器不会或极少向不同国家的 CA 申请证书; 然而, 通过大规模的实际网络数据分析(涉及 170 亿条 SSL 会话), 2013 年的后续研究表明[38]: 即使在正常合法的情况下, 服务器证书也是频繁变化的, 其上层根 CA 的所属国家也会变化. 某种程度上说, 正常合法的服务器证书变化, 在表象上与恶意的中间人攻击很类似、难以区分; 给攻击检测带来了很大挑战. 根据现有 CA 所签发服务器证书的顶级域名情况(例如.com、.org、.net、.de、.uk 等)来分析, 各 CA 所服务的顶级域名比较稳定; CAge 方案[12]提出, 在客户端浏览器上给各 CA 设定所允许签发服务器证书的顶级域名范围; 一旦发现有 CA 签发了设定范围之外的服务器证书, 则警告提示用户. 事实上, 除了.com、.org、.net 等少数域名外, 其它的顶级域名(如.de、.uk)通常都表示了国籍; 也就是, Certlock 机制和 CAge 方案分别考虑了 CA 和 Web 服务器所属国家, 来限制 CA 可以给任意 Web 服务器签发证书的巨大权力.

CA-TMS 方案[11]提出了对 CA 的信任程度计算和信誉评价系统, 不同的 PKI 依赖方对于各 CA 有不同的信誉评价, 各依赖方可以利用其它依赖方的评价、综合地计算自己对特定 CA 的信任值; 只有满足阈值的 CA 才会被信任、所签发的服务器证书才会被接受.

DVCert 协议[37]则考虑直接依赖用户自身、而不是第三方来获取正确的服务器证书. 在建立 SSL/TLS 连接之前, 用户利用其与服务器之间事先已经共享的口令、运行 DVCert 协议[37]鉴别服务器、然后接收服务器发送来的证书列表; 该证书列表用于在随后的 SSL/TLS 协议中与服务器证书对比, 确认通信对方是可信的服务器、而不是恶意的中间人攻击者. 相比 TLS-SRP[39]、PAKE-HTTP[40]等结合口令和 SSL/TLS 的协议机制, DVCert 协议不需要对已有网络协议做改动. DVCert 协议排除了恶意 CA 的攻击影响, 但是场景受限, 只能用于用户与服务器之间已有共享秘密的情况.

3 大规模 PKI 系统的实施部署

经过了二十多年的发展, PKI 技术已经开始进入了大规模实施部署. 在单一 PKI 域中, 可以容易地完成

证书认证路径构建和证书验证. 在大规模 PKI 系统实施中, 需要考虑采取交叉认证等方式实现跨域互联; 其次, 还需要分析跨域依赖方的证书验证. 在证书验证过程中, PKI 依赖方通常需要访问资料库获得证书链上的中级 CA 证书/交叉证书/CRL 等文件、还需要访问 OCSP 等撤销信息服务. 对于非跨域情况, 依赖方一般都已正确配置了资料库和撤销信息服务的访问地址; 但是对于跨域互操作的 PKI 依赖方, 并不知道其它 PKI 域的资源访问地址, 给证书认证路径构建和证书验证带来障碍[41]. 文献[41]提出 PRQP 协议: 建设公共的 PRQP 服务器, 各 CA 都可以在 PRQP 服务器上注册自己的各种服务资源访问地址(包括资料库、CRL、OCSP、代理验证服务等); 然后跨域的 PKI 依赖方就可以在 PRQP 服务器上自动地查询各种访问地址.

文献[14]分析了欧盟 Euro6IX 泛欧网络建设和跨域身份联合等服务中的 PKI 系统需求, 按照欧洲的 PKI 系统真实情况, 构建完整了的 PKI 互操作模拟环境, 涉及层次 CA 系统、CA-CA 交叉认证、桥 CA、证书扩展、证书验证服务等; 利用该模拟环境, 在不同的请求模式下(串行和并发), 实验比较了不同的撤销服务(CR/ARL 和 OCSP)和证书认证路径长度对跨域 PKI 依赖方验证证书的性能影响. 模拟实验结果对 CA 系统互联模式、证书扩展、证书认证路径查找方法、撤销服务等给出了建议.

文献[15]分析了现有 ICAO(International Civil Aviation Organization)ePassport 电子护照标准中的 PKI 系统设计, 讨论了如何实现各国 ePassport 电子护照 PKI 系统的互联, 然后在此基础上进一步扩展建设为全球范围的、互操作的身份标识系统. 具体措施包括: 各国 Country Signing Certification Authority(CSCA)之间的交叉认证、将已有 ePassport 电子护照转为用户的证书/密钥对、电子护照的目录服务、电子护照的撤销服务等等.

为了规范 CA 的证书签发服务, CA/Browser Forum 在 2010 年推出了 Extended Validation SSL(EV SSL)规范[42], 被世界上主要 CA 公司采纳. 然而, 最近的一系列 CA 公司安全事件表明, SSL/TLS 服务器证书生态环境的整体安全性, 更取决于全体 CA 公司的整体证书服务水平. 为此, CA/Browser Forum 于 2013 年推出了 Baseline requirements for the issurance and management of policy-trusted certificates(基线要求)[43]. 2014 年, 文献[16]收集了互联网上公开使用的 1,480,028 张 SSL/TLS 服务器证书, 全面的数据分析表明: 现有的服务器证书远没有符合上述的基线要求, 包括在命名标识、算法、证书扩展、证书撤销服务等方面都仍有很大差距; 全面提升 SSL/TLS 服务器证书生态环境的安全程度, 仍然任重而道远.

2011 年, IETF 开始由 SIDR 工作组来进行 Resource PKI(RPKI)的标准化工作[44]. RPKI 使用 X.509 PKI 来保护 BGP 路由协议通信安全; IANA(Internet Assigned Numbers Authority)执行根 CA 功能, 与 RIR(Regional Internet Registry)、LIR(Local Internet Registry)形成 3 层 CA 结构, 为用户签发证书. 在 RPKI 证书中, 以证书扩展的方式表示了 IP 地址资源的分配. 目前 RPKI 的部署仍未全面开展, 文献[17]给出了 RPKI 的实验性实施情况.

在 DNSSEC 标准中, 也采取了类似 PKI 服务的方式来传递 DNS 服务器的公钥(虽然并不是使用标准的 X.509 证书格式), 用该公钥来验证数字签名保护的 DNS 域名数据. 按照现有 DNSSEC 标准[45,46], 其 PKI 系统是严格的树形结构, 与 X.509 PKI 系统类似: 上层 Zone 为下层 Zone 的公钥执行数字签名、绑定公钥数据; 公钥数据和数字签名以 DNSKEY RR 格式和 RRSIG 格式分别存储. 在本文中, 方便起见, 我们也将其统称为证书. 与 X.509 PKI 的显著不同在于, 在 DNSSEC 中, 每一个 Zone 都可以执行 CA 功能、为其下一层 Zone 签发证书, 因为 DNS 域名可以任意多级; 在这一点上, 与 PGP 有相似之处. 针对 DNSSEC 服务的独有特性, 文献[18]从多个方面讨论了现有 DNSSEC 标准中的证书服务: (1) 可伸缩性, 新的 DNSSEC 标准克服了原有标准在证书存储位置设计缺陷而导致密钥更新困难问题, 使得即使有超大规模的子 Zone、也能以较低代价更新密钥, 以及相应的 ZSK 和 KSK 的分离设计; (2) 灵活性, 原有 DNS 服务器的冗余配置能容忍错误和失效, 但是在 DNSSEC 实施之后, 相应地复制多份密钥对反而会增加密钥泄露的风险, 需要权衡二者的利弊; (3) 渐进部署, 在 DNSSEC 的部署过程中, 如何将逐步将新建立的根 CA 公钥安全地配置到所有 DNS 解析器中; (4) DNS 缓存影响, DNS 缓存所导致的、在密钥更新和撤销情况下的密钥和签名不匹

配; (5) 撤销机制, 现有 DNSSEC 标准尚未有设计完善的撤销服务、存在安全隐患.

此外, 文献[19]和文献[20]讨论了大规模 PKI 系统应用中的证书撤销服务, 请参考下一节阐述.

4 数字证书撤销技术及其它

在 PKI 技术研究中, 证书撤销一直都是受关注的重点. 主要原因是, 证书撤销服务通常都涉及 PKI 依赖方的在线操作, 为了权衡在撤销服务中各方的通信、计算、存储、延迟等因素, 就有了各种不同的证书撤销方案, 常见的撤销方案有 CRL[47]、OCSP[48]、CRT[49]、NOVOMODO[51]等. RevCast 系统[21]利用 FM 广播系统的单向广播信道来及时地传播证书撤销列表 CRL. 相比已有的、使用 TCP/IP 网络的 CRL 发布方式, 基于 FM 广播信道的方案具有及时、低成本、保护隐私等特点.

在现有各种证书撤销服务方案中, 都要求 PKI 依赖方能检查撤销状态信息的有效性. 在撤销状态信息(如 CRL 文件、OCSP 响应消息等)的检查过程中, 时效性检查是重要步骤; 否则, 攻击者就可以重放以前的、尚未被撤销时的撤销状态信息(例如, 重放旧的 CRL 文件). 然而, 在某些场景中, 时效性检查难以执行. 针对没有内置时钟的 RFID 标签, 文献[22]给出了用于高端 RFID 标签的、检查读卡器撤销状态的方案: 要求 RFID 标签配置简单的数字显示接口, 使得 RFID 在接收读卡器的证书和撤销状态信息、自动验证其中的数字签名后(RFID 标签存储有可信的根 CA 证书), 显示证书和撤销状态信息的失效时间, 供持有人确认后再开始通信.

现有的 ePassport 电子护照系统标准中, 并未明确证书撤销服务方案. 文献[19]比较了 CRL、NTP+OCSP、SCVP、Hoepman、BioPACE V2、OSEP 等多种证书撤销方案, 从安全性、易用性、代价、可伸缩性、可靠性、可行性等方面完成了深入全面的比较. 分析结果认为, NTP+OCSP 和 BioPACE V2 具有优势. WebDAV 协议[20]将证书和撤销状态信息分别定义为不同的 Web 资源, 使用 REST(Representational State Transfer)架构来存储; PKI 依赖方使用 HTTP 协议访问证书和撤销状态信息, 使得能够完成即时的证书撤销服务. WebDAV 协议成功地用于 PERMIS 授权管理基础实施[51].

利用 IBE 算法天然的密钥备份和托管特性, RIKE 方案[52]将 IBE 算法的 PKG 部件功能引入 PKI 系统, PKI 依赖方将数字证书的二进制比特串作为 IBE 算法的 ID 输入、可以从一张证书同时获得 2 对密钥对: 其中 1 对是按照传统方式绑定在证书中、密钥不托管, 用于数字签名, 另一对是 IBE 算法导出的密钥对、私钥被自动托管, 用于数据加密. 使用 RIKE 方案, 一张证书自动绑定 2 对密钥对, 且正好分别满足了数字签名和数据加密对于密钥托管的矛盾需求. 与此同时, 又可以使用 PKI 系统中丰富的证书撤销服务来解决 IBE 密钥对的撤销: 在 RIKE 方案中, 证书撤销就认为 2 对密钥对都不再有效.

文献[30]分析了现有浏览器的隐身模式(Private Browsing Mode)实现中、由于证书撤销状态检查而导致的隐私泄露: 浏览器在与服务器建立 SSL/TLS 会话时, 使用 OCSP 来验证服务器证书的撤销状态; 然而, 当浏览器结束使用、删除各种数据时, 并没有删除所缓存的 OCSP 响应消息, 留下了访问痕迹、泄露隐私.

此外, 近年来的 PKI 技术研究成果还有如下. 基于 PKI 技术的代码签名, 已经在 Android 平台上广泛使用、用于保护 APP 代码完整性; 在 APP 更新时, Android 系统要求新版本 APP 与原有版本带有相同的开发者证书, 否则就不能更新. 上述机制保证了只有源自于相同开发者的更新版本才能够访问原有 APP 的数据. 但是当开发者更换证书(例如, 因证书过期、密钥更新等), 就无法顺利地完成 APP 更新. Baton 系统[53]改进了 Android 系统的代码签名验证机制, 使得: 利用原有证书可验证的 Token、向新的开发者证书授权, 就能在 APP 更新时、同时更新使用新的开发者证书. 文献[54]列出了新近发现的、若干与 PKI 证书服务相关的安全漏洞, 包括证书命名解析漏洞、PKCS #10 请求消息的 SQL 插入攻击、EV SSL 会话劫持等等.

5 总结

PKI 技术作为能够实现身份鉴别、机密性、完整性、非否认等核心安全服务的基础设施，在信息系统安全中发挥着重要作用. 作为典型的密码应用技术，近 5 年来，PKI 系统在走向大面积应用推广的过程中，其技术研究出现了新的深度. 这些技术研究成果得益于 PKI 技术在智能移动平台、SSL/TLS、大规模网络系统等领域的深入应用. 我们有理由相信，作为普适性的安全基础设施，随着信息技术发展，PKI 技术扩展到新的应用领域，将来必定会有更丰硕的研究成果. PKI 证书服务的安全性、大规模 PKI 应用技术、新型领域的 PKI 技术(包括物联网、虚拟化环境、无线通信领域和下一代高速网络等场景)等工作将会在未来有更进一步的深入探讨和研究.

References

[1] Kohnfelder L. Towards a practical public-key cryptosystem[D]. MIT Bachelor Thesis, 1978.
[2] Dierks T, Rescorla E. RFC 5246: The transport layer security (TLS) protocol version 1.2[S]. 2008.
[3] Georgiev M, Iyengar S, Jana S, et al. The most dangerous code in the world: Validating SSL certificates in non-browser software[C]. In: ACM Conference on Computer and Communications Security—CCS 2012. ACM, 2012: 38–49.
[4] Fahl S, Harbach M, Muders T, et al. Why Eve and Mallory love Android: An analysis of Android SSL (in)security[C]. In: ACM Conference on Computer and Communications Security—CCS 2012. ACM, 2012: 50–61.
[5] Zuo C, Wu J, Guo S. Automatically detecting SSL error-handling vulnerabilities in hybrid mobile Web Apps[C]. In: ACM Symposium on Information, Computer and Communications Security—AsiaCCS 2015. ACM, 2015: 591–596.
[6] Laurie B, Langley A, Kasper E. RFC 6962: Certificate transparency[S]. 2013.
[7] Ryan M. Enhanced certificate transparency and end-to-end encrypted mail[C]. In: ISOC Network and Distributed System Security Symposium—NDSS 2014.
[8] Szalachowski P, Matsumoto S, Perrig A. PoliCert: Secure and flexible TLS certificate management[C]. In: ACM Conference on Computer and Communications Security—CCS 2014. ACM, 2014: 406–417.
[9] Basin D, Cremers C, Kim H-J, et al. ARPKI: Attack resilient public-key infrastructure[C]. In: ACM Conference on Computer and Communications Security—CCS 2014. ACM, 2014: 382–393.
[10] Laurie B, Kasper E. Revocation transparency[EB/OL]. 2012.
[11] Braun J, Volk F, Classen J, et al. CA trust management for the Web PKI[J]. Journal of Computer Security, 2014, 22(6): 913–959.
[12] Kasten J, Wustrow E, Halderman J. CAge: Taming certificate authorities by inferring restricted scopes[C]. In: International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2013: 329–337.
[13] Soghoian C, Stamm S. Certified lies: Detecting and defeating government interception attacks against SSL[C]. In: International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2011: 250–259.
[14] Millán G L, Pérez M G, Pérez G M, et al. PKI-based trust management in inter-domain scenarios[J]. Computers & Security, 2010, 29(2): 278–290.
[15] Lekkas D, Gritzalis D. e-Passports as a means towards a globally interoperable public key infrastructure[J]. Journal of Computer Security, 2010, 18(3): 379–396.
[16] Delignat-Lavaud A, Abadi M, Birrell A, et al. Web PKI: Closing the gap between guidelines and practices[C]. In: ISOC Network and Distributed System Security Symposium—NDSS 2014.
[17] Wahlisch M, Holler F, Schmidt T, et al. Updates from the Internet backbone: An RPKI/RTR router implementation, measurements, and analysis[C]. In: ISOC Network and Distributed System Security Symposium—NDSS 2013.
[18] Yang H, Osterweil E, Massey D, et al. Deploying cryptography in Internet-scale systems: A case study on DNSSEC[J]. IEEE Transactions on Dependable and Secure Computing, 2010, 8(5): 656–669.
[19] Buchmann N, Baier H. Towards a more secure and scalable verifying PKI of eMRTD[J]. Journal of Computer Security, 2014, 22(6): 1025–1049.
[20] Chadwick D, Antony S, Bjerk R. Instant certificate revocation and publication using WebDAV[J]. Journal of Computer Security, 2010, 18(3): 475–496.
[21] Schulman A, Levin D, Spring N. RevCast: Fast, private certificate revocation over FM radio[C]. In: ACM Conference on Computer and Communications Security—CCS 2014. ACM, 2014: 799–810.
[22] Nithyanand R, Tsudik G, Uzun E. User-aided reader revocation in PKI-based RFID systems[J]. Journal of Computer Security, 2011, 19(6): 1147–1172.
[23] Sounthiraraj D, Sahs J, Greenwood G, et al. SMV-Hunter: Large-scale, automated detection of SSL/TLS man-in-the-middle

vulnerabilities in Android Apps[C]. In: ISOC Network and Distributed System Security Symposium—NDSS, 2014.
[24] He B, Rastogi V, Cao Y, et al. Vetting SSL usage in applications with SSLINT[C]. In: IEEE Symposium on Security and Privacy (S&P), 2015. Springer Berlin Heidelberg, 2015: 519–534.
[25] Bates A, Pletcher J, Nichols T, et al. Securing SSL certificate verification through dynamic linking[C]. In: ACM Conference on Computer and Communications Security—CCS 2014. ACM, 2014: 394–405.
[26] Holz R, Riedmaier T, Kammenhuber N, et al. X.509 forensics: Detecting and localising the SSL/TLS men-in-the-middle[C]. In: European Symposium on Research in Computer Security—ESORICS 2012. Springer Berlin Heidelberg, 2012: 217–234.
[27] Fahl S, Acar Y, Perl H, et al. Why Eve and Mallory (also) love webmasters[C]. In: ACM Symposium on Information, Computer and Communications Security—AsiaCCS 2014. ACM, 2014: 507–512.
[28] Dietz M, Czeskis A, Balfanz D, et al. Origin-bound certificates: A fresh approach to strong client authentication for the Web[C]. In: USENIX Security Symposium, 2012: 317–331.
[29] Parsovs A. Practical issues with TLS client aertificate authentication[C]. In: ISOC Network and Distributed System Security Symposium—NDSS 2014.
[30] Stark E, Huang L S, Israni D, et al. The case for prefetching and prevalidating TLS server certificates[C]. In: ISOC Network and Distributed System Security Symposium—NDSS 2012.
[31] Clark J, van Oorschot P. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements [C]. In: IEEE Symposium on Security and Privacy (S&P). IEEE, 2013: 511–525.
[32] Perrig A, Jackson C, Gligor V. Accountable key infrastructure (AKI): A proposal for a public-key validation infrastructure[C]. In: International Conference on World Wide Web, 2013: 679–690.
[33] Zhou L, Schneider F, Renesse R. COCA: A secure on-line certification authority[J]. ACM Transactions on Computer Systems, 2002, 20(4): 329–368.
[34] Wu T, Malkin M, Boneh D. Building instrusion tolerant applications[C]. In: USENIX Security Symposium, 1999: 79–91.
[35] Jing J, Liu P, Feng D, et al. ARECA: A highly attack resilient certification authority[C]. In: ACM Workshop on Survivable and Self-Regenerative Systems. ACM, 2003: 53–63.
[36] Perl H, Fahl S, Smith M. You won't be deeding these any more: On removing unused certificates from trust stores[C]. In: International Conference on Financial Cryptography and Data Security, 2014. Springer Berlin Heidelberg, 2014: 307–315.
[37] Dacosta I, Ahamad M, Traynor P. Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties[C]. In: European Symposium on Research in Computer Security—ESORICS 2012. Springer Berlin Heidelberg, 2012: 199–216.
[38] Amann B, Sommer R, Vallentin M, et al. No attack necessary: The surprising dynamics of SSL trust relationships[C]. In: Annual Computer Security Applications Conference—ACSAC 2013. ACM, 2013: 179–188.
[39] Taylor D, Wu T, Mavrogiannopoulos N, et al. RFC 5054: Using the secure remote password (SRP) protocol for TLS authentication[S], 2007.
[40] Oiwa Y, Takagi H, Watanabe H, Suzuki H. PAKE-based mutual HTTP authentication for preventing phishing attacks[C]. In: International Conference on World Wide Web, 2009: 1143–1144.
[41] Massimiliano P, Smith S. Finding the PKI needles in the Internet haystack[J]. Journal of Computer Security, 2010, 18(3): 397–420.
[42] CA/Browser Forum. Guidelines for the issuance and management of extended validation certificates, v1.4[S]. 2012.
[43] CA/Browser Forum. Baseline requirements for the issuance and management of policy-trusted certificates, v1.1.5 [S]. 2013.
[44] Huston G, Bush R. Securing BGP and SIDR[EB/OL]. IETF Journal, 2011, 7(1).
[45] Eastlake D. RFC 2535: DNS security extensions[S]. 1999.
[46] Arends R, Austein R, Larson M, et al. RFC 4034: Resource records for the DNS security extensions[S]. 2005.
[47] Cooper D, Santesson S, Farrell S, et al. RFC 5280: Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile[S]. 2008.
[48] Myers M, Ankney R, Malpani A, et al. RFC 2560: X.509 Internet public key infrastructure online certificate status protocol—OCSP[S]. 1999.
[49] Kocher P C. On certificate revocation and validation[C]. In: Financial Cryptography. Springer Berlin Heidelberg, 1998: 172–177.
[50] Micali S. NOVOMODO: Scalable certificate validation and simplified PKI management[C]. In: Annual PKI Workshop. 2002: 15–25.
[51] Chadwick D, Otenko A, Ball E. Role-based access control with X.509 attribute certificates[J]. IEEE Internet Computing, 2003, 7(2): 62–69.
[52] Zhang N, Lin J, Jing J, et al. RIKE: using revocable identities to support key escrow in PKIs[C]. In: Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2012: 48–65.
[53] Barrera D, McCarney D. Baton: Certificate agility for Android's decentralized signing infrastructure[C]. In: ACM Conference on Security and Privacy in Wireless and Mobile Networks. ACM, 2014: 1–12.

[54] Kaminsky D, Patterson M, Sassaman L. PKI layer cake: New collision attacks against the global X.509 infrastructure[C]. In: International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2010: 289–303.

作者信息

林璟锵(1978–), 福建石狮人, 博士, 副研究员. 主要研究领域为云计算安全、PKI 和密钥管理.
E-mail: linjingqiang@iie.ac.cn

荆继武(1964–), 湖南洪江人, 博士, 研究员. 主要研究领域为网络与系统安全.
E-mail: jingjiwu@iie.ac.cn

张琼露(1987–), 河南郑州人, 硕士, 研究实习员. 主要研究领域为可信计算、云计算.
E-mail: zhangqionglu@iie.ac.cn

王展(1986–), 吉林白城人, 博士, 助理研究员. 主要研究领域为云存储安全、物联网安全、移动终端数据保护.
E-mail: wangzhan@iie.ac.cn