A Survey on X.509 Public-Key Infrastructure, Certificate Revocation, and Their Modern Implementation on Blockchain and Ledger Technologies

Salabat Khan, Fei Luo, Zijian Zhang, *Member, IEEE*, Farhan Ullah, Farhan Amin, Syed Furqan Qadri, Md Belal Bin Heyat, Rukhsana Ruby, Lu Wang, Shamsher Ullah, Meng Li, *Senior Member, IEEE*, Victor C. M. Leung, *Life Fellow, IEEE*, and Kaishun Wu, *Fellow, IEEE*

***Abstract*—Cyber-attacks are becoming more common against Internet users due to the increasing dependency on online communication in their daily lives. X.509 Public-Key Infrastructure (PKIX) is the most widely adopted and used system to secure online communications and digital identities. However, different attack vectors exist against the PKIX system, which attackers exploit to breach the security of the reliant protocols. Recently, various projects (e.g., Let's Encrypt and Google Certificate Transparency) have been started to encrypt online communications, fix PKIX vulnerabilities, and guard Internet users against cyber-attacks. This survey focuses on classical PKIX proposals, certificate revocation proposals, and their implementation on blockchain as well as ledger technologies. First, we discuss the PKIX architecture, the history of the World Wide Web, the certificate issuance process, and possible attacks on the certificate issuance process. Second, a taxonomy of PKIX proposals, revocation proposals, and their modern implementation is provided. Then, a set of evaluation metrics is defined for comparison. Finally, the leading proposals are compared using 15 evaluation metrics and 13 cyber-attacks before presenting the lessons learned and suggesting future PKIX and revocation research.**

***Index Terms*—Blockchain, privacy, revocation, ledger technology, public-key infrastructure (PKI).**

Manuscript received 12 February 2023; revised 11 July 2023 and 8 September 2023; accepted 2 October 2023. Date of publication 13 October 2023; date of current version 22 November 2023. This work was supported in part by China NSFC under Grant U2001207, Grant 61872248, and Grant U2001207; in part by Guangdong NSF under Grant 2017A030312008; in part by the Shenzhen Science and Technology Foundation under Grant ZDSYS20190902092853047 and Grant R2020A045; in part by the Project of DEGP under Grant 2019KCXTD005, Grant 2021ZDZX1068, and Grant 2023KCXTD042; in part by the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019JC01X235 and Grant 2019ZT08X603; and in part by the "Guangdong Provincial Key Laboratory of Integrated Communication, Sensing, and Computation for Ubiquitous Internet of Things." *(Corresponding author: Kaishun Wu.)*

Salabat Khan, Fei Luo, Farhan Ullah, Md Belal Bin Heyat, Rukhsana Ruby, Lu Wang, and Shamsher Ullah are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: salabatwazir@gmail.com; luofei2018@outlook.com; farhan.marwat@gmail.com; belalheyat@gmail.com; ruby@szu.edu.cn; wanglu@szu.edu.cn; shamsher@szu.edu.cn).

Zijian Zhang is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100811, China (e-mail: zhangzijian@bit.edu.cn).

Farhan Amin is with the Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea (e-mail: farhanamin10@hotmail.com).

Syed Furqan Qadri is with the Research Center for Healthcare Data Science, Zhejiang Lab, Hangzhou 311121, China (e-mail: furqangillani79@gmail.com).

Meng Li is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: mengli@hfut.edu.cn).

Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vleung@ieee.org).

Kaishun Wu is with the Data Science and Analytics (DSA) & Internet of Things (IoT) Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China, and also with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: wuks@ust.hk; wu@szu.edu.cn).

Digital Object Identifier 10.1109/COMST.2023.3323640

I. Introduction

RECENTLY, there has been a dramatic increase in Internet usage due to the increasing reliance on smart services and online applications. Cities and homes are becoming smarter, where various Internet of Things (IoT) devices play different roles in making life comfortable. Online applications such as online banking, mailing, and e-commerce have become an integrated and indispensable part of our daily lives due to their significant contributions. A report revealed that around 5 billion people were using the Internet in 2022, with a total increase of around 200 million (+4.1%) over the last year [1]. These Internet users rely on the World Wide Web (WWW) to access information systematically. Recently, Web 3.0 has also attracted the attention of financial institutions, researchers, and academia. With the growing reliance on online applications and intelligent services, cyber-attacks are increasing against users, WWW applications, intelligent services, and devices. The Skybox Security report revealed that 20,175 vulnerabilities were identified in 2021, with an overall increase of almost 24% [2].

Moreover, Snowden's report (The 2013 National Security Revelations) revealed that government agencies are involved in breaching the security and privacy of Internet users [3]. Recently, state-level adversaries have tried to intercept Internet user traffic in Kazakhstan [4]. The traffic interceptions of Internet users by different governments and other powerful

adversaries set a dangerous precedent [4]. In response, various initiatives and projects have been started to secure and encrypt Internet traffic using cryptographic algorithms. For example, Google started Certificate Transparency (CT) [5] and encrypted the links between data centers [6], while Apple set encryption to default on its mobile devices [7]. Similarly, the Internet Engineering Task Force (IETF) declared passive monitoring as a cyber-attack [8] and started the Let's Encrypt [9] project to encrypt Web traffic through cryptographic methods. Cryptographic algorithms can be divided into two major classes: symmetric and asymmetric (i.e., Public-Key Cryptography (PKC) algorithms. The first class uses a single key to encrypt/decrypt information, which raises concerns about the secure distribution of the encryption/decryption key in open communication scenarios such as the Internet, which is known as the key management problem of the symmetric algorithm.

To address the key management problem of the symmetric algorithm, Diffie and Helman introduced PKC in 1976, which uses a pair of keys: Public-Key (PK) to encrypt information, and Secret-Key (SK) used to decrypt information. They used the concept of a public registry to maintain the PK to name mappings to solve the problem of key management. However, another identity authentication problem was raised with the introduction of PKC. To solve the identity authentication problem, Public-Key Infrastructure (PKI) has emerged, providing a foundation for PKC. Among the different PKI standards, the X.509 Public-Key Infrastructure (PKIX), known as the Internet PKI/Web PKI, gained widespread adoption among the Internet and Web community. Unfortunately, the PKIX design has severe flaws and vulnerabilities that are exploited by attackers [10] and pointed out by researchers [11], [12]. Several schemes and experimental projects have been initiated to fix the vulnerabilities.

The main aim of this survey article is to present an up-to-date and comprehensive overview of PKIX, certificate revocation proposals, recent proposals, and threats to and from them, along with their impact on users. In addition, this survey provides a succinct comparison among the leading proposals using 15 evaluation metrics and 13 cyber-attacks. Table I lists all the terms and abbreviations used in this article.

A. Structure of This Survey

Fig. 1 illustrates the structure of this survey, and the rest of the survey is organized as follows. Section II reviews related surveys and presents the contributions of this survey. Section III provides an overview of PKIX, PGP, the World Wide Web, and the application of PKIX. Section IV introduces the issuance of the Domain Validated (DV) certificate (DV-certificate), the validation methods, the array of cyber-attacks, and CA failures. Section V presents the taxonomy of PKIX and revocation proposals. Section VI defines the evaluation metrics and performs a detailed comparison of the leading PKIX and revocation proposals. Lessons learned, research gaps, and future perspectives are presented in Section VII before drawing the final conclusion in Section VIII.

TABLE I
ABBREVIATIONS USED THROUGHOUT THIS ARTICLE

Acronyms	Definitions
ACME	Automated Certificate Management Protocol
AS	Autonomous System
BF	Bloom Filter
BGP	Border Gateway Protocol
CA	Certification Authority
CDN	Content Delivery Networks
CMHT	Chronological Merkle Hash Tree
CRLs	Certificate Revocation Lists
CSR	Certificate Signing Request
CT	Certificate Transparency
DDoS	Distributed Denial of Service
DV	Domain Validated
DV-certificate	Domain Validated certificate
DNS	Domain Name System
HTML	Hyber Text Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hyper Text Transfer Secure
IMAP	Internet Message Access Protocol
ISO	International Organization for Standardization
ISPs	Internet Service Providers
LDAP	Lightweight Directory Access Protocol
MitM	Man-in-the-Middle
MHT	Merkle Hash Tree
OCSP	Online Certificate Status Protocol
PK	Public-Key
PKI	Public-Key Infrastructure
PGP	Pretty Good Privacy
POP	Post Office Protocol
RA	Registration Authority
SCI	Secure Channel Injection
SMTP	Simple Mail Transfer Protocol
S/MIME	Secure Multipurpose Internet Mail Extensions
SK	Secret-Key
SSL	Socket Layer Security
SPoF	Single-Point-of-Failure
SPKI	Simple PKI
TLD	Top Level Domains
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOFU	Trust-On-First-Use
UDP	User Datagram Protocol
WoT	Web of Trust
WWW	World Wide Web
XMPP	Extensible Messaging and Presence Protocol

II. RELATED WORK

Despite a large body of work on PKIX and revocation proposals, to the authors' best knowledge, there is so far no complete survey on PKIX and revocation proposals, recent attacks on them, their modern implementation on ledger and blockchain technologies having the same scope and approach as presented in this survey article.

Rueppel and Wildhaber [13] discussed the legal and technical challenges faced during the design and operation of a PKI framework. Grant [14] evaluated Convergence [15], Perspectives [16], Domain Name System (DNS)-based Authentication of Named Entities (DANE) [17], [18], [19], Certification Authority Authorization (CAA) [20], Mutually Endorsed Certification Authority Infrastructure (MECAI) [21], PK Pinning (PKP) [22], Sovereign Keys Infrastructure (SKI) [23], and Google Certificate Transparency (CT) [5]. Amin et al. [24] analyzed security issues along with the energy and time cost of PKC algorithms for wireless sensor networks.

Clark and Van Oorschot [25] surveyed and classified HTTPS security problems and presented future research challenges. Furthermore, they compared some existing certificate management schemes. Parsovs [26] analyzed the problems with

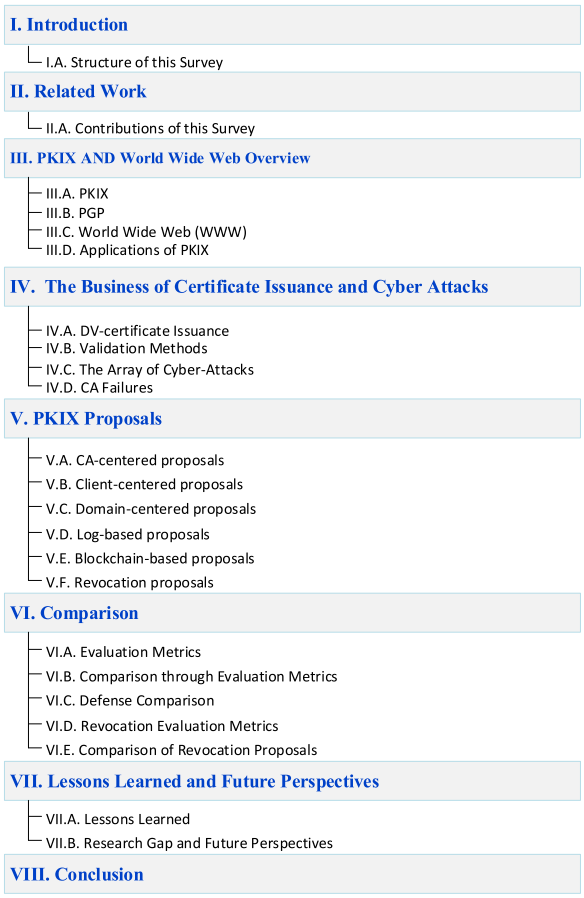


Fig. 1. Structure of this survey.

the TLS certificates of clients in practice using data collected in Estonia. Delignat-Lavaud et al. [27] measured the actual level of adherence of issued certificates to the CA/Browser Forum regulations. Albarqi et al. [28] discussed the PKI component operations and compared PKIX and Pretty Good Privacy (PGP). Huang et al. [29] analyzed Man-in-the-Middle (MitM) attacks on Facebook through forged TLS/SSL certificates. Zhang et al. [30] investigated an OpenSSL bug known as Heartbleed, which had revealed the private keys of web servers.

Holz et al. [31] investigated the security of chat and email frameworks. Michael and Joseph [32] analyzed the adoption of strict transport security (HSTS) [33] and PKP by the Web community and their information leakage. At the same time, the work in [34] examined the client and server side implementation of HSTS and PKP. Vandersloot et al. [35] measured certificate-related data collected from different famous sources, such as Google CT logs, and proposed recommendations on how to perform future measurements. Heiland et al. [36] presented a general overview of PKIX with a discussion on modeling and simulation of the cost of certificate validation. The work in [37] presented in-depth PKC and the execution time, energy, and other resources consumed by the PKC primitives on wireless devices. Yu and Ryan [38] presented a chapter on conventional and log-based PKIX proposals. They also listed evaluation metrics for the evaluation of PKIX proposals. Gustafsson et al. [39] used passive and active methodologies to pinpoint similarities and differences among Google CT [5] logs along with their usage, including certificates, and the association between regularly monitored certificates in Web traffic with transparency log-based certificates.

Amann et al. [40] investigated the new security features of protocols such as Google CT, DNS Security Extension (DNSSEC) [41], and HSTS added to HTTPS to complement its security. Nykvist et al. [42] measured the adoption of Signed-Certificate-Timestamp (SCT) of Google CT among the top one million famous domains. They developed a client side measurement tool to carry out their investigation of SCT adoption. Gasser et al. [43] investigated the compliance of certificates with the baseline requirements in PKIX. Li et al. [44] explored the reliability of monitors in the context of CT. Malick et al. [45] investigated and compared PKC-based key bootstrapping proposals for IoT applications. Li et al. [46] identified that CT's third-party monitors are potentially exposed to MitM attacks. They concluded that this vulnerability of the monitors could jeopardize CT security. Khan et al. [47] explored the security and privacy issues of 5G technologies caused by the introduction of new technologies such as software-defined networking, cloud computing, and virtualization of network functions.

The works in [50], [51], [52] examined DANE [17], [18], [19] deployment and its impact on the security of web applications. Brunner et al. [53] discussed and compared the implementation of blockchain-based PKI and provided recommendations for future implementation. Chuat et al. [54] discussed the delegation of trust, revocation, and proxy certificates used in Web communication. They also presented parameters for comparing the surveyed schemes. Aldahwan and Alghazzawi [55] conducted a systematic review of the literature by designing research questions and a search string. They highlighted the problems of PKI and how these problems can be fixed using the blockchain. The work in [56] shed light on the past 30 years of DNS deployment and on its future. Albogam et al. [57] discussed the conventional and blockchain-based PKIX proposals along with the comparison between them. Bansal and Sethumadhavan [62] enlightened the security issues of DNS and the schemes presented by researchers to overcome these problems. They also covered blockchain-based DNS proposals that could solve DNS problems. Meiklejohn et al. [58] analyzed proposals that are proposed for privacy-preserving auditing of domain certificate inclusion in Google CT logs. Maldonado-Ruiz et al. [59] focused on decentralized identity management and the trend to implement PKI in blockchain technology. They concluded that CA is still the main trust anchor in the new PKI implementations. de Carnavalet and van Oorschot [60] studied 30 proposals and discussed the challenges posed by TLS traffic interceptions in middleboxes such as Content Delivery Networks (CDN). Safaei Pour et al. [61] reviewed the Internet measurement techniques and provided a taxonomy of them across two dimensions. One dimension is related to the components of the Internet ecosystem, while the second is related to internal proper functioning against the negative influence of external third parties on the Internet.

TABLE II
A COMPARISON OF RELATED SURVEY PAPERS CONTRIBUTIONS

Survey Paper	Year	Domain	PKIX coverage	Revocation	PKIX Challenges	Revocation Challenges	Detailed Comparison	Defense Comparison
Rueppel et al. [13]	1995	Technical and legal issues of public PKI design.	✓	✗	✗	✗	✗	✗
Amin et al. [24]	2008	PKC algorithms resource consumption on resource contrianed devices..	✗	✗	✗	✓	✗	✗
Grant [14]	2012	Comparison of PKIX proposals.	✓	✗	✗	✓	✗	✗
Amann et al. [48]	2013	Malicious certificates detection.	✓	✗	✗	✗	✗	✗
Akhawe et al. [49]	2013	Browser warning caused by certificates.	✓	✗	✗	✗	✗	✗
Clark et al. [25]	2013	Security issues of HTTP.	✓	✓	✗	✓	✗	✗
Parsovs et al. [26]	2013	Clients' TLS certificates issues.	✓	✓	✗	✗	✗	✗
Delignat-Lavaud et al. [27]	2014	certificates compliance to the CA/Browser Forum regulations.	✓	✗	✗	✓	✗	✗
Albarqi et al. [28]	2014	Background of PKI and discussion on PKIX and PGP.	✓	✓	✗	✗	✗	✗
Holz et al. [31]	2015	Chat and email security investigation.	✓	✗	✗	✗	✗	✗
Zhu et al. [50]	2015	DNS deployment and its security impact.	✓	✓	✓	✓	✗	✗
Aishwarya et al. [51]	2015	DNS deployment and its security impact.	✓	✓	✓	✓	✗	✗
Michael [32]	2015	HSTS and PKP adoption analysis and information leakage.	✓	✗	✗	✗	✗	✗
Kyung-Ah [37]	2015	PKC primitives and their cost on wireless devices.	✓	✗	✗	✗	✗	✗
De et al. [34]	2016	HSTS and PKP client and server side adoption.	✓	✗	✗	✗	✗	✗
Vandersloot et al. [35]	2016	certificate analysis and measurement.	✓	✓	✓	✓	✗	✗
Heiland et al. [36]	2016	PKIX overview and certificate validation cost.	✓	✗	✗	✓	✗	✗
Yu et al. [38]	2017	Classical and log-based PKIX proposals.	✓	✓	✓	✓	✓	✗
Gustafsson et al. [39]	2017	certificates logged in Google CT anlysis.	✓	✗	✗	✗	✗	✗
Amann et al. [40]	2017	Security analysis of deployed PKIX proposals.	✓	✗	✓	✗	✗	✗
Nykvist et al. [42]	2018	Google SCT adoption among famous websites.	✓	✓	✗	✗	✗	✗
Gasser et al. [43]	2018	Certificate compliance to PKIX requirements.	✓	✓	✓	✗	✗	✗
Li et al. [46]	2019	Google CT monitor vulnerabilities.	✓	✗	✗	✗	✗	✗
Khan et al. [47]	2019	Security and privacy issues of 5G technologies.	✓	✗	✗	✗	✗	✗
Malick et al. [45]	2019	Key bootstrapping for the application of IoT.	✗	✗	✗	✗	✗	✗
Lee et al. [52]	2020	DANE security.	✓	✓	✓	✗	✗	✗
Brunner et al. [53]	2020	Blockchain-based PKI comparison.	✓	✓	✓	✗	✗	✗
Chuat et al. [54]	2020	Trust delegation and proxy certificates.	✓	✓	✓	✓	✓	✗
Aldahwan and Alghazzawi [55]	2020	PKIX problems.	✓	✓	✓	✓	✗	✗
Work [56]	2021	DNS deployment past and future perspective.	✓	✗	✗	✗	✗	✗
Albogam et al. [57]	2021	Traditonal and modern PKI implementation comparison.	✓	✓	✗	✗	✗	✗
Meiklejohn et al. [58]	2022	Privacy-preserving auditing of CT logs	✓	✗	✗	✗	✗	✗
Maldonado-Ruiz et al. [59]	2022	Blockchain-based PKI implementation.	✓	✓	✓	✓	✗	✗
Carnavalet et al. [60]	2023	TLS traffic interception.	✗	✗	✗	✗	✗	✗
Pour et al. [61]	2023	Internet measurement techniques.	✓	✓	✗	✗	✗	✗
Our work	2023	Traditional and modern PKIX, revocation proposals, and their detailed comparison.	✓	✓	✓	✓	✓	✓

Table II compares the related survey articles and their contributions. In Table II, the year shows the time of publication of the survey article, while the domain shows the area covered by the survey article. Table II also examines related articles in terms of PKIX and coverage of the revocation process. It can be observed from the table that most surveys cover PKIX and the revocation process. However, the PKIX and the revocation process challenges are covered in a few articles. In terms of comparison through evaluation metrics, only Yu and Ryan [38] and Chuat et al. [54] conducted a detailed comparison. However, both works partially compare all schemes. For example, Yu and Ryan [38] compare only classical and log-based PKIX proposals, while Chuat et al. [54] investigate revocation schemes, delegation schemes, and certificate features. Regarding the defense comparison, none of the existing surveys has provided a detailed comparison between PKIX proposals and revocation proposals.

From Table II and the comprehensive discussion on related survey articles, it is clear that neither of the survey articles covers the recent attacks on the certificate issuance process and a detailed comparison of defense of leading proposals against these attacks. In addition, only a few articles either partially covered revocation proposals or provided a detailed comparison. The survey works carried out in [38], [54] investigate classical PKIX proposals, log-based PKIX proposals, and some revocation proposals without discussing blockchain-based PKI proposals. No survey articles in the literature, however, have presented a detailed survey on traditional PKIX proposals, traditional revocation proposals, and their modern implementation, including comparison through evaluation metrics and recent attacks among leading proposals. To our knowledge, this effort represents the first survey covering PKIX proposals, revocation proposals, and their advanced versions with detailed comparisons.

A. Contributions of This Survey

The main contributions of this survey are as follows.

- We survey the state-of-the-art PKIX proposals by starting with a comprehensive discussion on the PKIX architecture, Web history, PGP architecture, the business of the certificate issuance process, and possible recent cyber-attacks.

- A taxonomy of PKIX proposals is provided based on their nature of defense and implementation technology, and a taxonomy of revocation proposals is presented based on data structure and revocation information dissemination methods. Additionally, the strengths and weaknesses of each scheme are discussed.
- A set of evaluation metrics is defined to evaluate, examine, and compare PKIX and revocation proposals. A detailed and unified comparison is made among the leading PKIX and revocation proposals through the evaluation metrics defined in Section VI-A. Furthermore, the defense of leading PKIX and revocation proposals against the recent attacks discussed in Section IV-C is investigated.
- Finally, the lessons learned and future challenges of PKIX and the revocation process are presented.

III. PKIX and World Wide Web Overview

This section reviews the widely deployed key management infrastructure known as the PKI of the Internet based on the recommendations of the ITU-T X.509 certificate. The history of the WWW is briefly discussed before delving into the PGP key management infrastructure presented by Phil Zimmermann.

A. PKIX

In 1988, ITU-T approved specifications on how to link an identity to a PK, authenticate the identity, and developed a standard called X.509 [63] in association with the X.500 standard [64]. Since the introduction of PKI, the deployment of PKI on the Internet has been dominated by the ITU-T X.509 standard [65], known as PKIX. Major protocols such as TLS/SSL and protocols that use TLS/SSL for secure communication, such as HTTPS, POP, SMTP, IMAP, XMPP, and LDAP, inherently rely on PKIX to establish secure connections [65]. PKIX defines a standard format for a certificate, known as the X.509 certificate signed by Certification Authorities (CA), and procedures for registration, initialization, certification, cross-certification, key pair generation, update, certificate expiry, and certificate revocation. Fig. 2 shows a traditional PKIX architecture that comprises the following five main components.

- Certification Authority (CA): It is responsible for granting and revoking X.509 certificates. It can delegate certificate issuance power to sub-ordinate CAs (sub-CAs) and administrative functions such as identity verification to Registration Authorities (RA).
- Server: An entity to whom a certificate is issued by a CA and is identified on the subject part of the X.509 certificate. The server (e.g., www.a.com) uses the acquired certificate(s) for authentication and encryption/decryption.
- Registration Authority (RA)[1]: An additional entity that usually performs identity verification and entity registration. It can assume various types of administrative functions of the corresponding CA.

[1]RA is an optional entity in PKIX. For simplicity, RA is omitted from the remainder of the discussion on PKIX.

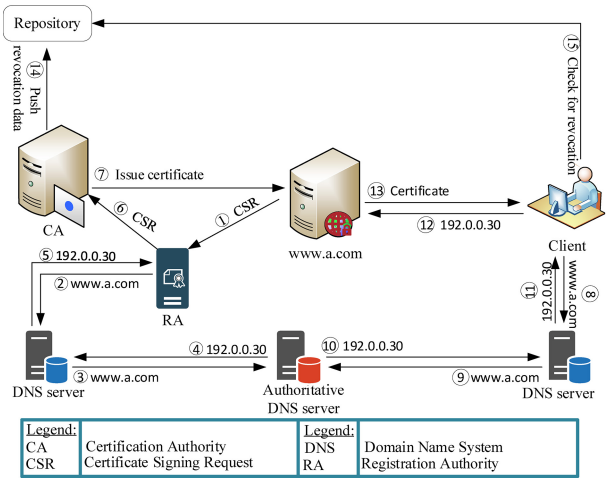


Fig. 2. A high-level overview of traditional PKIX architecture.

- Client: An entity that uses the services offered by Web servers and authenticates them through certificate verification.
- Repository: An entity holding and offering Certificate Revocation Lists (CRLs) to validate the revocation status of X.509 certificates.

A server (end-entity) that intends to use a certificate for secure communication first sends a Certificate Signing Request (CSR) after registering directly or indirectly through RA to CA. A certificate is issued against the CSR that contains basic identity information (e.g., common name, IP address, and domain name) after successfully verifying the identity and the information provided. Client browsers are provided with the necessary material, such as root CA PKs during the initialization process, that enables clients to verify a CA-signed domain certificate. During the establishment of a secure connection, a client receives CA-signed certificates from the server to which the client wants to connect. The client validates the received certificate by verifying that a trusted CA signs the certificate, that it has not expired, and that it has not been revoked by checking the revocation status on the CRLs. In PKIX design, CAs are entirely trusted servers that issue and revoke digital certificates for clients. PKIX has become a de facto standard for securing Web communication, but its design has several drawbacks. For example, CA is a Single-Point-of-Failure (SPoF), and a single compromised CA can undermine trust globally [48]. The failure of the famous CA revealed the fragile security and trust placed in CAs in practice. Consequently, Google started the CT project to fix CA issues by introducing the CT log, monitor, and auditor entities to the PKIX. Various solutions to mitigate the problems of PKIX, including Google CT, are discussed in Section V in detail.

B. PGP

Phil Zimmermann designed PGP [66] in the late 1991. PGP differs entirely in trust and key management from PKIX. PGP uses a decentralized Web of Trust (WoT) as a trust model

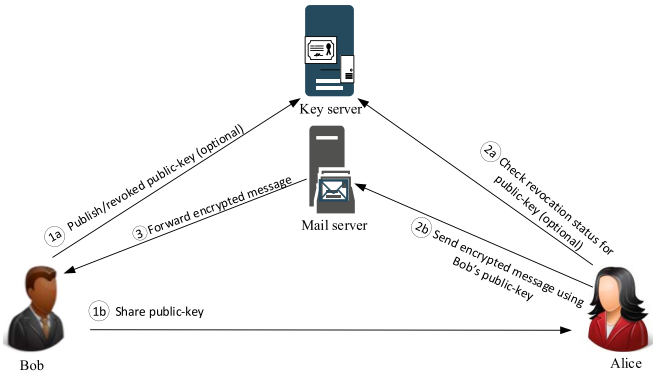


Fig. 3. A generic PGP architecture.

instead of the centralized CA trust model. In WoT, users are responsible for key generation, distribution, attestation, and revocation. In PGP, clients designate PKs of other clients via semi-trusted paths with different degrees of trust, representing how trustworthy the attestation of the certificate owner is when the certificate owner attests the certificates of other clients. More specifically, how reliable the client's introduction of other clients is in the network. In PGP, each client maintains a directory of its private key in encrypted form and a directory of PKs, including its own and other peers. Directories are called rings, and rings grant double trust ratings. The first rating implies how trusted the user's PK is, and the second rating implies how trusted the PK of a client is to add new clients. When enough trusted peers sign a peer PK, that user (peer) is considered a trusted user. PGP assigns four levels to each client regarding the introduction of other clients.

- Fully trusted: Entirely trusted to add new peers.
- Partially trusted: Partially trusted to add new peers.
- Untrusted: Not trusted to add new peers.
- Unknown: Not rated yet to add new peers.

As shown in Fig. 3, the management of PK rings is reliant on peers. Any user may want to evict its PK in a ring in case of a key compromise. The user has to inform all users by sending them a signed revocation message. The decentralized nature of PGP makes it immune to SPoF; however, it also imposes two difficulties: joining the network by new clients and revoking keys. New clients wishing to join the PGP network must visit existing clients for identity verification. As mentioned above, key revocation is also cumbersome in PGP-based networks. PGP is still used by many applications to secure emails and encrypt files, although it has several problems. A detailed survey on PGP is left for future work.

C. WWW

Tim Berners-Lee introduced the WWW in 1989 to offer on-demand information and data sharing among scientists, researchers, and experts in universities and educational institutes around the globe while working at the CERN lab. His idea was to create a decentralized information sharing protocol and platform that enables information sharing anywhere on Earth. The first realization of WWW from 1990 to 2004 is called Web 1.0. In Web 1.0, end-users could only read information shared on the web.

Later, Web 2.0 began with the emergence of social media platforms, where end-users can write on the Web instead of merely being readers. During this era (2004-present), end-users were enabled by Web companies to share end-users-generated content on the Web and engage them in peer-to-peer interactions. This model further birthed the advertisement-driven revenue generation model. While end-users can create content, they cannot own it and benefit from monetization.

To enable end-users ownership of content, the concept of Web 3.0[2] (2014-Future) was presented by Ethereum cofounder Gavin Wood[3] to address the absolute trust in a handful of companies and enterprises to act in the public's best interests shortly after Ethereum was launched in 2014. The core design is to eliminate blind trust and build the infrastructure over blockchain, where end-users can read, write, and own assets. Fig. 4 shows the evolution of the Web from version 1.0 to future version 3.0.

D. Application of PKIX

As mentioned earlier, PKIX is the most widely used PKI on the Internet. Various applications, including Virtual Private Network (VPN) software, software development tools, mobile applications, cloud services and applications, e-commerce applications, and many more, rely on PKIX for secure communication. Some applications have developed their PKI by modifying the PKIX model to meet the requirements of the applications. The different set of requirements for different applications is the main driver behind the development of different PKI flavors. For example, Vehicular PKI (VPKI) has been developed to secure vehicular communication as PKIX fails to meet vehicular communication requirements. Standardization bodies in the U.S. and Europe have mandated the use of VPKI for secure vehicular communications [67]. VPKI was developed for privacy-preserving vehicular communication by adopting the PKIX model. The vehicular scenario fundamentally differs in terms of privacy, authentication, revocation, and delay requirements. Obscuring a vehicle's real identity and location is a top priority in VPKI, while PKIX was not framed with privacy as the primary goal and objective. In terms of authentication, VPKI relies primarily on broadcast-based real-time authentication, where each vehicle is required to validate safety messages broadcast by devices within the communication range within a predefined time limit (e.g., 100 milliseconds $\sim$ 300 milliseconds); otherwise, they expire. In contrast, PKIX relies primarily on the client-server authentication model, where the client is required to validate messages from a known server without a strict time limit. The revocation mechanism used in PKIX, such as CRL, fails when applied to vehicular communication due to the dynamic network topology and real-time validation requirements [67].

[2]Although Web 3.0 aims to integrate different technologies such as AI. Here, we cover Web 3.0 only from the design perspective presented by Gavin Wood.

[3]https://ethereum.org/en/web3/

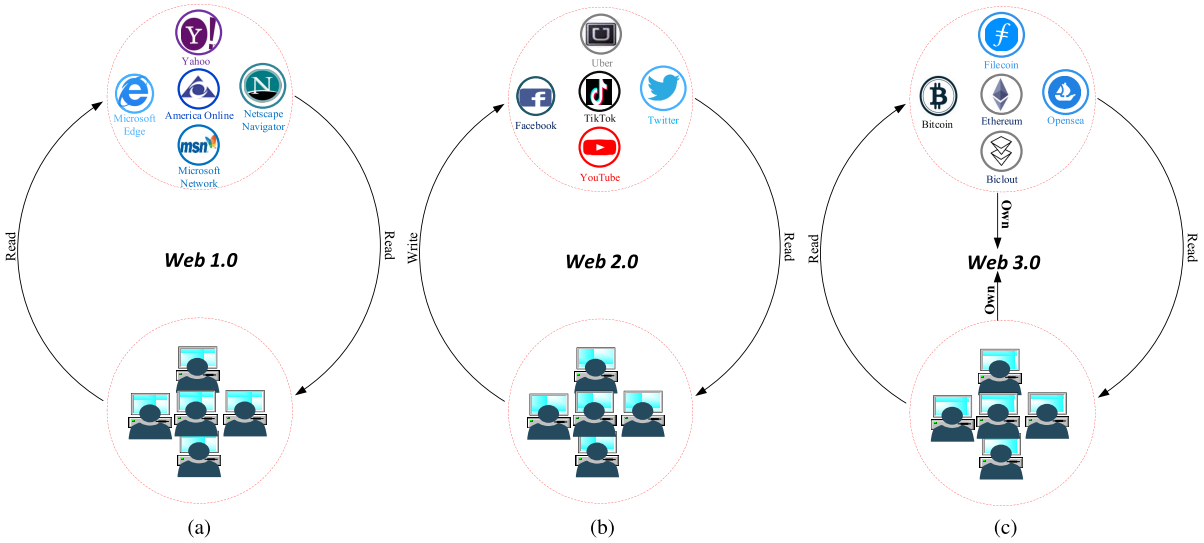


Fig. 4. An overview of web evolution.

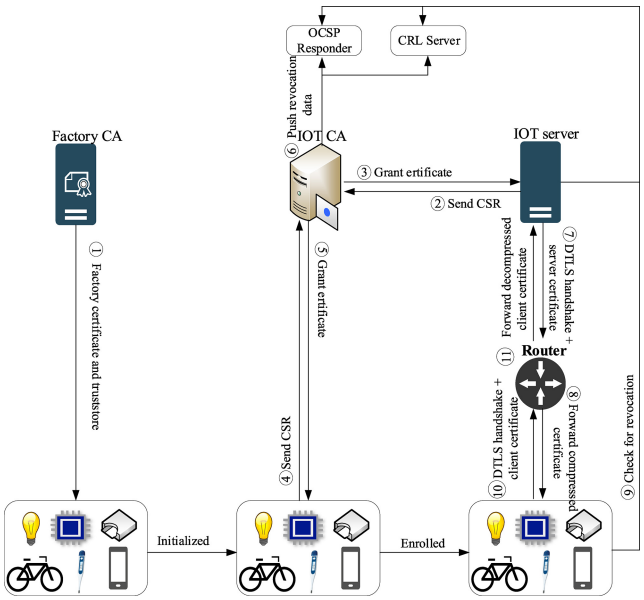


Fig. 5. An overview of PKIX application in IoT scenario source [68].

Similarly, IoT communications and applications have a different set of requirements due to the resource-constrained nature of IoT devices. Since PKI requires heavy computational operations for encryption/decryption process, which could drain the available energy of resource-constrained edge devices. To achieve a tradeoff between energy efficiency and security, various mechanisms such as lighter cryptographic algorithms (Elliptic Curve Cryptography), offloading computation to powerful machines, and on-demand and layered security can be utilized. Therefore, Höglund et al. [68] designed a lightweight flavor of PKI based on the PKIX model for IoT applications by replacing heavy operations with lightweight ones. Fig. 5 shows a PKI framework for secure IoT communications, where communications happen using Datagram TLS (DTLS). The IoT PKI [68] still relies on CRL and OCSP for certificate revocation. The work in [69] proposed lightweight CRL, and the work in [70] presented lightweight OCSP for the revocation of the certificate of IoT devices, while the work in [71] designed a lightweight certificate. A detailed survey on PKI and revocation for each application scenario, such as IoT and integrated network, is left for future work, while interested readers are referred to the work in [67] for a detailed survey on VPKI proposals and the differences between PKIX and VPKI requirements.

IV. The Business of Certificate Issuance and Cyber-Attacks

In this section, the DV-certificate issuance process and domain control validation methods are first discussed. Then, a typical cyber-attack against conventional PKIX is illustrated before discussing the possible attacks against PKIX and the DV-certificate issuance process.

A. DV-Certificate Issuance

The Automated Certificate Management Protocol (ACME) [72] domain validation and certificate issuance process is discussed, which is used by Let's Encrypt CA and other CAs to automate domain validation and certificate issuance. The ACME protocol relies on identity validation mechanisms (see Section IV-B) to validate control over a domain. Fig. 6 shows the domain control verification and the certificate issuance process using the HTTP-based challenge response verification. The ACME protocol comprises three major steps: 1) a domain owner sends a CSR to an ACME-based CA, and then the CA sends a set of challenges for ownership verification; 2) the owner completes the set of challenges by replying with responses, and 3) the CA grants DV-certificate to the domain owner upon successful completion of challenges. Other control verification methods include email-based verification, DNS-based verification, or offline methods that are not reliant on the Internet (e.g., letters of authorization). Although the ACME protocol makes the

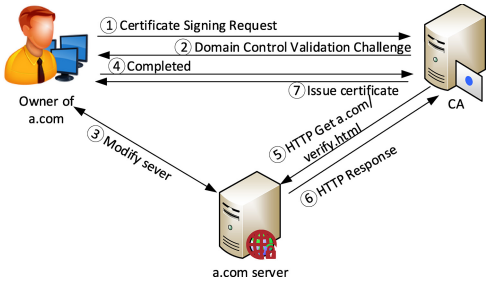


Fig. 6. A high-level overview of DV-certificate issuance.

certificate issuance process automatic, the ACME protocol is vulnerable to different cyber-attacks, which are discussed in the next section. More specifically, step 5 and step 6 can be attacked by an adversary to obtain a malicious certificate for a victim domain.

B. Validation Methods

1) Email-Based Validation: If a CA supports this validation method, it typically sends an email to the administrative domain contact chosen by the applicant out of the registered email addresses on the whois database for that domain. The email typically composes a code and link used for domain control validation. The applicant clicks on the link and enters the received code to prove the ownership of the domain. The CA issues a certificate if the applicant enters the correct code.

2) HTTP-Based Validation: Upon receiving a CSR, the CA responds with a challenge and asks the applicant to place it on the root of the Web server (at a well-known URL), as shown in Fig. 6. The CA makes a GET HTTP request to access the file and issues a certificate upon successful validation. In HTTP-based validation, the issuing CA can select HTTP or HTTPS to verify domain control. Unfortunately, the HTTPS-based validation mode can be downgraded to HTTP [73].

3) Whois-Based Validation: In this validation method, an email is used as the main validation source. However, the significant difference lies in the email selection process. In this method, CA randomly picks up an email address registered on the Whois database for that domain for validation, unlike the email-based validation method, where applicants provide an email.

4) DNS-Based Validation: Under this verification method, the CA asks an applicant to insert a challenge as a DNS Canonical NAME (CNAME) Resource Record (RR) for the domain server in the zone file. If the CA domain is ca-domain.com, then the CNAME record for the a.com domain would be challenge1.www.a.com CNAME challenge2.ca-domain.com. The CA checks the record using a DNS resolver and grants a certificate if the record is found. Table III lists some famous CAs with their supported domain validation methods.

C. The Array of Cyber-Attacks

In this section, a typical cyber-attack against PKIX is illustrated, and then the possible attack vectors against PKIX are discussed, which researchers have exploited to acquire

TABLE III
SOME LEADING CAS AND THEIR SUPPORTED VALIDATION METHOD

CA	DNS	HTTP(s)	Whois	Email
GoDaddy	✓	✓	✗	✓
GobalSign	✓	✓	✗	✓
InstantSSL	✓	✓	✗	✓
Certum	✓	✓	✗	✓
GeoTrust	✓	✓	✗	✓
SSL.com	✓	✓	✗	✓
RapidSSL	✓	✓	✗	✓
Amazon	✓	✓	✗	✓
Starfiled Technologies	✓	✓	✗	✓
COMODO	✓	✓	✗	✓
Let's Encrypt	✓	✓	✗	✗
Unizeto	✓	✗	✗	✓
NETLOCK	✓	✗	✗	✓
Thawte	✗	✗	✓	✗
Symantec	✗	✗	✓	✗
Entrust	✗	✗	✓	✗
DigiCert	✗	✗	✓	✗
IdenTrust	✗	✗	✓	✗
IdenTrust	✗	✗	✗	✓
StartSSL	✗	✗	✗	✓

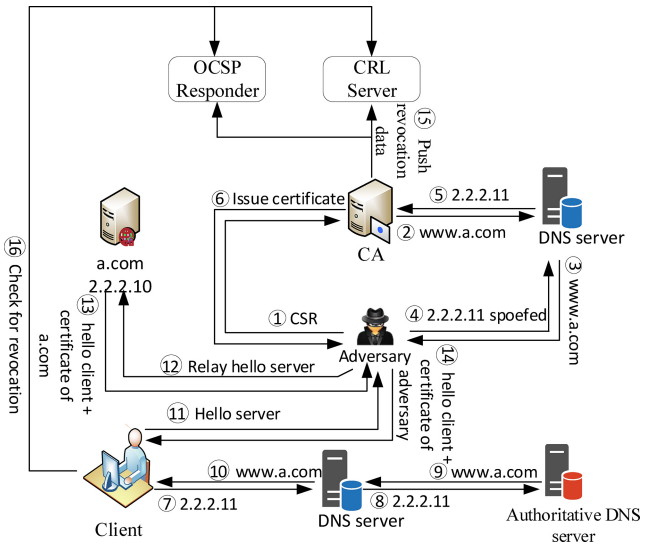


Fig. 7. An overview of a typical cyber-attack.

malicious certificates. The possible attacks on PKIX can arise from loopholes within the PKIX system, such as zombie certificates and malicious CA insertion attacks, or from the core network, such as BGP route and DNS hijacking attacks.

1) Cyber-Attack Illustration: A typical cyber-attack is conducted against PKIX clients using malicious certificates. An attacker can use rogue certificates to intercept, redirect, or alter victim Internet traffic. Fig. 7 illustrates a simplified version of a cyber-attack, where an adversary bypasses the CA validation process. The cyber-attack involves several steps, which are demonstrated in Fig. 7. In steps 1 to 6, the adversary acquires a fake but valid certificate for a victim domain using a DNS record spoofing (see Section IV-C6) attack[4] to bypass the CA validation process. The adversary installs the malicious certificate on a server to conduct a cyber-attack against the victim

[4]Note that DNS record spoofing attack is used only for demonstration purpose. The adversary can use other kinds of attack to bypass the CA validation process.

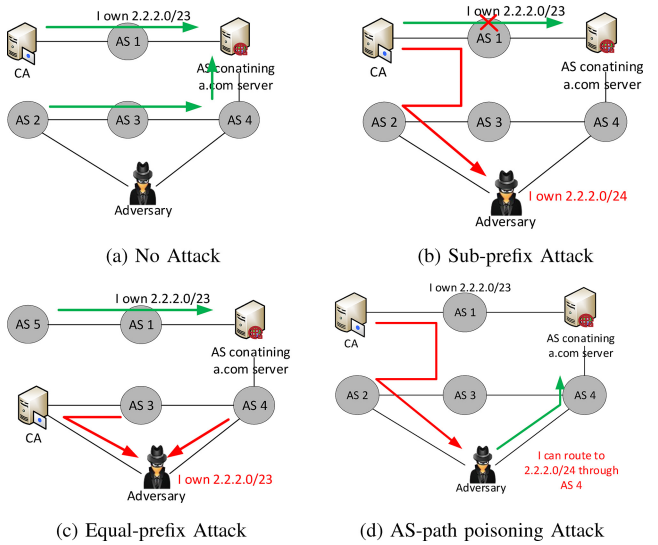
domain and its clients. The client obtains the IP address of the victim domain in steps 7 to 10, as shown in Fig. 7. When the client starts the TLS connection by sending a hello server message in step 11, the adversary starts conducting a MitM attack against the victim by controlling the communication between the victim client and the server, as shown in the steps between 12 and 14 of Fig. 7.

2) CA Compromises: It is an attack in which a trusted CA gets corrupted and issues false certificates to websites. Since any browser-trusted CA can issue certificates for any website acceptable by browsers, and thus a powerful attacker can take down any CA to acquire malicious certificates and conduct cyber-attacks on sites and their users. For example, two famous CAs, Comodo and DigiNotar, were compromised and certificates for the top sites were fraudulently acquired. In the case of DigiNotar, malicious certificates were used in MitM attacks.

3) Compelled Certificate Attack: It is an attack in which a government agency needs a domestic CA to provide them with fraudulent TLS certificates for surveillance purposes. In such attacks, the governmental entities can either force the compelled CA to issue them a false certificate for each Web server to be spoofed, or more probably, the root CA is compelled to grant them an intermediate CA certificate that can further be used again and again by the governmental entities without the further assistance and knowledge of the compelled CA. These concerns were theoretically raised in 2010 by the work in [74]. Unfortunately, recent attacks by different agencies against end-users showed that governmental agencies are often well positioned to spoof websites by compelling CAs and/or controlling the network infrastructure.

4) BGP Route Hijacking: Border Gateway Protocol (BGP) hijacking is malicious control of groups of IP addresses through the compromising routing tables maintained and managed by BGP routers. It is a form of application layer DDoS attack, which Internet Service Providers (ISPs) usually cause when they do not adequately screen the network prefix announcements received from their peer ISPs before relaying the announcements to others. It enables an adversary to maliciously impersonate a network by adopting the prefix of a legitimate network. Internet traffic is inadvertently forwarded to the adversary instead of its origin and proper destination if the adversary convinces other networks to accept the advertised network prefix.

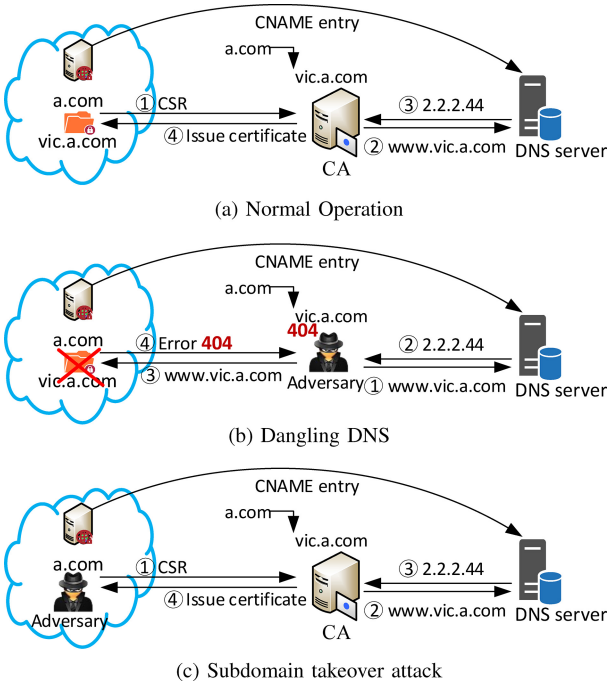
BGP hijacking can be used to masquerade domain control validation and acquire malicious certificates. The most effective types of BGP hijack used for domain control takeover are sub-prefix hijack, equally-specific hijack, and Autonomous System (AS) path poisoning attacks [11]. In the first type of hijacking, an attacker announces a sub-prefix that includes the IP address of the targeted domain. In the second case, an attacker announces a prefix of the same length as the victim domain server. In the last scenario, an adversary announces a sub-prefix of equal length to the victim domain prefix along an appended legitimate path that passes through the adversary's own Autonomous System Number (ASN) to the victim domain. Fig. 8 shows different attack scenarios. For details, we refer the readers to the work in [11].



(a) No Attack (b) Sub-prefix Attack

(c) Equal-prefix Attack (d) AS-path poisoning Attack

Fig. 8. BGP and path poisoning attack illustration source [11].



(a) Normal Operation

(b) Dangling DNS

(c) Subdomain takeover attack

Fig. 9. DNS dangling record and malicious certificate issuance through the subdomain takeover attack.

5) Dangling DNS Record Attack: Dangling DNS records point to expired services or resources in the authoritative DNS service. In this attack, stale DNS records pointing to free IP addresses that are not purged yet, are used to conduct domain takeover attacks. A typical dangling DNS record and domain takeover attack is shown in Fig. 9. The dangling DNS records can be further subdivided into expired domains, discontinued services, and deprovisioned cloud instances.

Deprovisioned cloud instance attack: The resources and services allocated in the cloud as an Infrastructure as a Service (IaaS) are well-known to cause the spread of unallocated IP addresses and stale DNS records. An attacker can exploit the unallocated IP addresses pointed out by DNS records in the

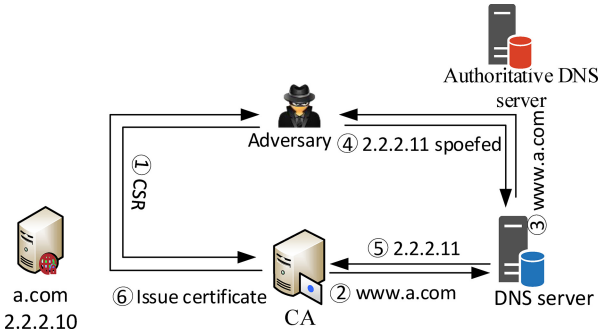


Fig. 10. A high-level overview of DNS record spoofing attack.

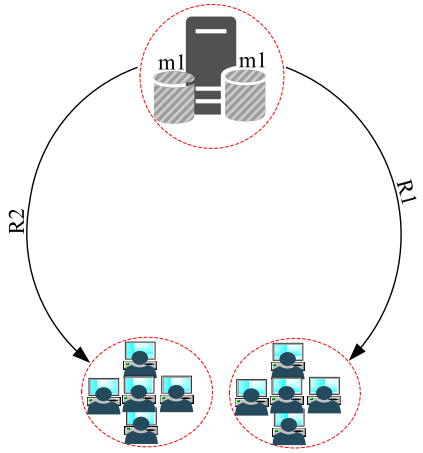


Fig. 11. A high-level overview of split-world attack.

cloud to impersonate a domain and bypass the CA domain validation process. The work in [75] identified dangling records, while the work in [12] exploited this vulnerability of the cloud and the integration of the ACME protocol. DNS records often refer to discontinued services that service providers had previously launched on the cloud.

Expired domain attack: In this attack, an expired CNAME of a domain is used to take control of a legitimate domain. If the CNAME is expired, any third party can register the domain and provide services under the alias domain name.

Discontinued service attack: In this attack, an adversary attacks third-party services used by websites to extend their functionality. Whenever domain owners use third-party services for the subdomain, the integrator has to configure the DNS record for the subdomain and explicitly validate the ownership of the subdomain. Suppose that the service provider does not validate the domain ownership. In that case, it exposes the domain to a takeover attack, and an adversary can take control by mapping any unclaimed domain to its own account by placing a valid DNS record [75].

6) DNS Record Spoofing Attack: In this attack, an adversary injects a spoofed DNS record into the DNS resolver's cache to impersonate victim domains to a CA and tricks the CA into issuing malicious certificates for domain servers owned by the adversary illegitimately. All the validation methods relying on DNS can be attacked through spoofing DNS records. Fig. 10 shows the DNS record attack on the domain validation process. This attack can be carried out mainly using a MitM attack or hijacking the DNS server. In the first case, the adversary learns the source port of the User Datagram Protocol (UDP) along with the communication ID to insert a spoofed DNS record into the response message. The victim accepts the response as a valid reply, as the UDP ID and communication ID are the same in the query. In the latter case, an adversary hacks a DNS server to perfectly modify the response.

7) Split-World Attack: In the literature, a split-world attack is also called equivocation, which is applicable if an adversary can show different views of records to different groups of clients. In this scenario, the adversary controls a trusted third party (e.g., CA and transparency log) and shows different copies of signed records to normal clients and victims. Fig. 11 shows a typical split-world scenario, where a trusted party shows the record R1 for a member m1 showing that m1 is invalid, and the record R2 for the same member m1 showing that m1 is valid to different sets of users simultaneously.

8) DNS Cache Poisoning Attack: In this attack, an adversary can insert a spoof record into the DNS cache even without knowing the DNS request. After poisoning the cache, the DNS resolver recalls the bad site for the victim, even if the issue is resolved or never happens on the server side. In 2005, five Unix servers caused the attack[5] while some attackers used BGP hijacking for DNS cache poisoning [76].

9) Vantage Point Downgrade Attack: In this attack, the adversary reduces the validation from multiple vantage points to a single adversary-selected nameserver by eliminating the nameservers. The adversary exploits a small set of vantage points selected by a CA and the methodology of selecting nameserver vulnerabilities of the DV-certificate issuance process. Using the mentioned vulnerabilities, the adversary can successfully downgrade the multi-vantage-points-to-multi-nameservers validation mechanism to multi-vantage-points-to-single-nameserver validation. For further details, we refer the readers to the base work in [77].

10) Truststore Attack: In this attack, an adversary replaces the truststore of a device with a malicious truststore that stores the certificates of malicious CAs. The malicious CAs are used to conduct cyber-attacks against users and breach their security and privacy.

11) Malicious CA Insertion Attack: In this attack, an adversary attacks users by inserting malicious CA certificates into their truststore. This attack has been launched by software and hardware vendors by inserting malicious CAs into their products.

12) Revocation Information Blocking Attack: In this attack, the attacker blocks victims' access to the certificate revocation status validation process to use the certificates revoked against them.

13) Zombie Certificate Attack: In this attack, an attacker uses a malicious DV-certificate that is supposed to be revoked by a CA. More specifically, a rogue CA helps the attacker to conduct attacks against victims by not revoking a certificate.

[5]http://isc.sans.org/presentations/dnspoisoning.php

TABLE IV
YEARLY CAS FAILURES

CA	Reason	Year
Stuxnet	Key leakage	2007
Thawte	Email spoofing	2008
StartCom	Email spoofing	2008
Comodo	RA failure	2008
ipsCA	Null prefix attack	2009
Etisalat	Internal system compromise	2009
VerSign	Unknown	2010
DigiNotar	CA system compromise	2011
TurkTrust	Internal error	2011
StartSSL	Impersonation/CA compromise	2008
Cyberroam	Key Signing leakage	2012
NICCA	CA certificate issuance compromise	2014
ANSSI	Internal system fault	2013
WoSign	Unpriviledged TCP port	2015
Let's Encrypt	Cryptographic flaw	2015
Symantec	Unknown	2017
GoDadd	Code vulnerability	2018
Certinomis	Unknown	2018
Trustico	Key disclosure	2018

TABLE V
CAS TAKEN DOWN BY RESEARCHERS

CA	Reason	Year
AlphaSSL	Email validation method attacked	2019
Amazon	Email validation method attacked	2019
Certum	Email validation method attacked	2019
Commodo	Email validation method attacked	2018
Let's Encrypt	HTTP validation method attacked	2018
SSL.com	Email validation method attacked	2019
GoDaddy	HTTP validation method attacked	2018
Starfields Technologies	Email validation method attacked	2018
Thwate	Email validation method attacked	2019
Symantec	Email validation method attacked	2018
GoDaddy	DNS validation method attacked	2019
GlobalSign	Email validation method attacked	2018

14) Efail Attack: In this attack, an adversary breaks OpenPGP and S/MIME email encryption by compelling email clients to send the entire plaintext of the email to the adversary. In Efail attacks, the adversary modifies an encrypted email in a special way and forwards the altered encrypted email to the victim. When the victim decrypts the malicious email and loads embedded external content (e.g., HTML link), it exfiltrates the plaintext to the attacker. For further details on Efail, we refer the readers to the work in [78].

D. CA Failures

Attacks on CAs and their failures were considered theoretical attacks on PKIX architecture during the early deployment stage. Unfortunately, different CA failures and cyber-attacks on them revealed their fragile ecosystem in practice. Table IV shows some leading compromises in practice, while Table V lists CAs taken down by researchers.

V. PKIX PROPOSALS

Classical PKIX, revocation proposals, and the recent proposals based on transparency logs and blockchain are discussed in this section. Classical PKIX proposals can be broadly divided into three main categories: CA-centered PKIX proposals, client-centered PKIX proposals, and domain-centered PKIX proposals. In addition to classical proposals, the modern proposal can be classified into two classes: log- and blockchain-based PKIX proposals.

A. CA-Centered Proposals

MECAI [21] is a certification framework that functions like notary-based schemes (e.g., Perspective [16] and Convergence [15]), and notary servers are run by participating CAs. Thus, CAs perform the dual functions of issuing server certificates and short-lived vouchers simultaneously. The short-lived voucher is a witnessed fact observed on the Web from the perspective of a notary server. Clients only accept certificates from Web servers accompanied by short-lived vouchers from CAs. The voucher contains the hostname, the server certificate from a TLS handshake, a timestamp, and a statement about the certificate revocation status.

Kasten et al. [79] proposed CAge to limit the scope of CA certification based on observations made by the study in [80] that states that only a small number of famous CAs have issued certificates for TLDs. It restricts a CA from endorsing certificates for a designated pool of TLDs, since CA behavior is mostly stable, allowing the restriction based on past behavior. They claimed that such a constraint could decrease the attack surface by 65–90%. Braun and Rynkowski [81] designed a tool called Rootopia to identify the list of CAs relevant to a specific client based on his/her browsing history. Braun et al. [82] presented trust views that work on a user-dependent local knowledge base to make trust decisions. The trust level and difference required for different applications are dynamically computed during trust validation. Braun et al. [83] restricted the scope of CAs at a user-specific level using the knowledge base of trust views [82]. Classen et al. [84] extended the concept of trust views by gathering CA reputation data from their users through the concept of peer-to-peer networking. It was observed that the proposals [83], [84] have the issue that a newly added CA with high compliance with community requirements may get lower reputation grades than an older CA with lower compliance with community requirements. Furthermore, proposals based on scope restrictions [79], [83], [84] cannot shield clients against malicious issuance by the CA, which is authorized to issue for the target domain.

Domain Validation++ (DV++) [85] is a recent domain identity verification protocol. DV++ complemented CAs' traditional domain validation, which is exposed to MitM attacks and mitigated through validation from multi-vantage points. In multi-vantage validation, each domain server identity is verified from different vantages to avoid path poisoning attacks and resist other on-path and off-path attackers.

Syta et al. [86] addressed the weakest-link issues of PKI by implementing a "Cothority" that federates all CAs of the current PKI. The Cothority uses the CoSi [87] signing mechanism to scale to many signatories and allows individual CA to detect fraudulent certificates before they are attested proactively. However, the scheme does not specify any mechanism for detecting and blocking fake certificates in practice [88].

Wang et al. [89] made CAs blind by introducing a privacy-preserving identity verification mechanism. In this work [89], blind CA can simultaneously validate the identity and issue a certificate without learning the information about the real identity. They introduced the Secure Channel Injection (SCI) protocol to validate clients' identities without learning their true identities. The SCI protocol includes generating several encrypted messages by the prover and verifier to conduct identity validation. The verifier can insert a secret message (challenge) at a designated point in the protocol. Later, the prover retrieves the challenge using a separate connection and relays it back to the verifier to complete the identity verification challenge. The work enabled traditional CAs to bootstrap an anonymous communication system without separating the identity validation and certificate issuance roles.

The Let's Encrypt CA [9] project was started by IETF to make certificate issuance, server configuration, and identity validation fast and accessible to encrypt web communication and data. This project [9] relies on ACME to automate the DV-certificate issuance process. This made the certificate issuance process quick and automatic. However, the CA is still fully trusted [90].

Perl et al. [91] analyzed ZMAP [92] and identified 140 unused CAs certificates that were never used in signing TLS certificates. They designed a solution and argued that removing or restricting the certificates of these unused CAs on the desktop and browser truststore would not invalidate any previously valid TLS/SSL certificates.

Jayaraman et al. [93] split the private key of a CA among multiple parties to reduce the risk that the private key can be misused in the event of a key compromise. The work in [93] uses a multiparty computation scheme to sign certificates that never expose the real signing key.

Lesson Learned: Despite various efforts to secure CAs against attacks, CAs still show the weakest link in the PKIX ecosystem. Recent attacks have shown that taking down CAs and bypassing their validation mechanism is practical. Additionally, CAs have low scalability and will not scale in case of future growth.

B. Client-Centered Proposals

Wendlandt et al. [16] implemented Perspective as an add-on for the Firefox browser in 2008. Perspective was designed to thwart PK cyber-attacks on the Trust-On-First-Use (TOFU) of PK during authentication by querying the PK status from specialized notary servers. Notaries are independent and distributed servers that monitor network traffic and provide their views about PKs to users. Notaries maintain a timestamped history of all observed PKs and periodically update their history by validating the revocation status of PKs. When an end-user receives a PK from a server, the end-user validates the PK from multiple notaries by querying them. If the results of the PK query are consistent, the end-user trusts and accepts the PK. Otherwise, it may reject the PK and consider a MitM attack by witnessing two different PKs for the server. Therefore, the discretion to trust a PK is transferred to end-users. It thwarts MitM attacks through malicious PKs (e.g., certificates). However, it does not support multiple PKs for a server, as end-users get warnings and alerts of conflicting views. Additionally, a fresh PK is exposed to a period of unavailability, as the PK can only be used after being witnessed by notaries.

DoubleCheck [94] solved the browsing history leakage and fresh PK unavailability problems of Perspective by introducing a double query of the PK from a target notary: one through a Tor secure connection and one through a TLS/SSL secured connection. DoubleCheck does not need to deploy new infrastructural components and conceals the user browsing history. However, it induces an additional 15 seconds of latency for each certificate verification [38].

Marlinspike concealed the user browsing history by implementing an add-on called Convergence in the Firefox browser [15]. Convergence introduced an onion routing-like mechanism to conceal browsing records, where users randomly select one notary of Perspective to pass the user request to other notaries. Therefore, the forwarding notary cannot know what the users are querying, while the notaries replying to the request cannot reveal the requester's identity. Clients keep a cache of certificates they encountered, and only query notaries for newly encountered certificates. It supports offline verification for credentials already encountered without querying the notary servers and protects browsing records. Like Perspective and DoubleCheck, Convergence does not support multiple PKs for a domain.

CertLock [74] is an add-on for monitoring the location of CAs for Firefox, where the add-on observes the country of CA that issues the certificate. CertLock alerts users if two different CAs issue certificates for the same domain from different countries. CertLock enables users to detect MitM attacks launched through valid but maliciously issued certificates by different CAs from different regions. However, it cannot shield and detect if the same CA gets corrupted and issues a fake certificate for the domain.

Crossbear [95] was designed to detect MitM attacks and locate them with a reasonable level of confidence. Crossbear uses notary servers to detect attacks and uses traceroutes to locate the attacks. To find traceroutes, the authors developed two types of hunter applications–a Mozilla Firefox add-on and a standalone hunter application. The add-on performs detection and localization tasks, while the standalone one can only locate the attacks. They maintain a list of servers that are under MitM attacks.

The Policy Engine [96] enables end-users to make user-specific decisions about which certificate to trust. The decision is made based on a policy of certificate sequence obtained from a Web server, which can be adopted by organizations or by individual end-users. The authors of this endeavor designed a repository that helped to interpret the policy and suggested an implementation language and integration of the policy to secure clients.

Bates [97] conducted a case study of deploying Convergence and forcing perspective-based validation on a complete university system in simulation. They found that a single Convergence deployment can meet the verification requirements of millions of TLS/SSL connections with 0.1%

communication overhead and 108 milliseconds (ms) verification delay on each connection.

CERTSHIM [98] designed easy-to-deploy dynamically shared objects to protect clients against TLS/SSL vulnerabilities. It further tried to transparently fix TLS/SSL problems and vulnerabilities in the validation applications. It includes a policy engine that enables clients to express complicated certificate validation procedures tailored to specific applications and domains or enforce the procedures system-wide. CERTSHIM was integrated with DANE, Convergence, and key-pinning proposals. It does not support browsers, and a local adversary and applications can bypass its security policies.

TrusBase [99] is a framework that offers certificate-based authentication as an Operating System (OS) service and enables system administrators to control the authentication policy. The framework includes eight services: CA validation, whitelist, key pinning, certificate revocation, CRLSets blocking, DANE, notary service, and cipher suite auditor. The certificate revocation check is performed through OCSP, while the CRLSets blocking check is used to extend the safety of the Chrome browser to all applications. The major disadvantage of this scheme is that it results in double validation for applications that perform certificate validation correctly.

Certificate Patrol [100] is another Firefox monitoring-based add-on, where all Web certificates are saved in the add-on encountered by the browser. The basic idea behind the Certificate Patrol add-on is that it is very unlikely that certificates will frequently change because they have an extended expiry time. Therefore, the add-on warns users if it encounters a new certificate for the domain that is different from the previously cached one. Accepting the newly encountered certificate is left to the user's discretion. The advantage of Certificate Patrol is that it is lightweight and offers an additional layer of defense against attacks. However, it is incompatible with multiple certificates.

Key pinning [22] cannot distinguish a fake certificate from a certificate update due to their use of the TOFU method. HTTPS Everywhere [101] is another client software add-on that makes it easy to enforce HTTPS-based communication with one click. The extension was deprecated at the end of 2021 due to widespread usage of HTTPS and HTTPS-only mode support by major browsers [102].

Lesson learned: It was observed that client-centered proposals need to query online servers to validate the certificate encountered. Thus, they need an additional connection to online servers, however, the update rate of servers is quite low and can lead to MitM in case of server infidelity. Furthermore, it introduces a window of unavailability and may not be compatible with short-lived certificates.

C. Domain-Centered Proposals

HPKP [103], [104] implemented PK Pinning (PKP) in Chrome browser, where domain servers specify a list of authorized CAs who can issue certificates for domains. Maintaining the mapping between CAs and domains causes the scalability issue of PKP, which was solved by declaring the CA list in the HTTP extension [104]. However, it does not protect domains if their designated CAs issue malicious certificates for them. Furthermore, Chrome intended to remove PKP after Google CT deployment [105].

Hodges et al. [106] thwarted MitM (e.g., protocol downgrade and cookie hijacking) by mandating HSTS and ignoring click-through warnings. It relies on the TOFU model and enables domain servers to enforce client applications to HTTPS connections only.

DANE [17], [18], [19] was proposed to bind a PK to a domain name through DNSSEC [41]. It allows domain servers to declare their PKs by recording them in the extension field of their DNS security. The record is valid if it is correctly endorsed as designated in DNSSEC [41] and only the parent domains are eligible for endorsing the DNSSEC record. It is scalable since domain servers assert trusted CAs in the DNSSEC field. Similarly, CAA [20] enables domain servers to assert the CAs authorized for attesting their PKs. However, DANE and CAA are reliant on DNS servers for their security.

Trust Assert for Certificate Keys (TACK) mitigated the need to place complete trust in CAs [107]. Each domain generates a TACK key pair and attests a TLS key pair using the TACK key pair. Each client caches and pins the PK to the domain name after multiple consistent observations of the same PK for the domain. The pinning is trusted for a certain period of time and accepts all PKs signed by the TACK key pair. If the TACK key pair is leaked or lost, then users need to delete the pinning history of the key pairs and refresh the pinning information. However, it does not protect clients on their first visit to domains.

Wang et al. [108] proposed an alternative approach to mitigate the damage of a failed CA and enhance the performance of certificate revocation through domain-based certification and revocation. Each individual domain operates a CA, which is responsible for the management of certificates, including the issuance and revocation of that particular domain. Each CA maintains a certificate and revocation database facilitating certificate validation and management. An entry is added to the former when a new certificate is granted and to the latter when a certificate is revoked. The work in [108] restricts the scope of CA, but it was observed that maintaining a CA per domain results in maintaining a huge number of CAs in the truststore of clients.

Elaphurus [109] is built on top of the PKP scheme. Elapshurus addresses secure initialization, TOFU, scalability, and update issues of the traditional PKP scheme. It initializes PKP entries if either one security scheme witnesses the certificate with the lowest false negative rate or two schemes with a low false positive rate. To update a pinned entry, the certificate should be signed by the same CA as the previous one and verified by the security system with a low false negative rate. The scheme supports multiple certificates per domain, introduces a mechanism to accept unpinned certificates through low and medium false negative security systems, and expiry time to remove pinned entries. However, Elaphurus does not provide a mechanism for updating a domain certificate if the domain changes the previous CA.

Borgolte et al. [12] identified attacks using stale DNS information associated with discontinued service. They showed that attackers could assign IP addresses to stale information pointed out by DNS records. Attackers can also abuse the trust placed in the domain name to phish, send and receive emails, or distribute malicious code to clients. Attackers could carry out such attacks in less than 70 seconds. They designed a domain validation method and proposed recommendations for domain owners and cloud operators to mitigate the exposure of their own and their users to the staleness of DNS records to avoid domain takeover attacks.

Lesson learned: DNS plays an important role during the Internet browsing process. Unfortunately, there are numerous attack vectors against DNS, and researchers have recently demonstrated attacks against DNS. A study revealed that domain takeover attacks increased by 20% in 2021 as compared to 2020. It was observed that the increasing trend of hosting domains on third-party services by large organizations and the lack of timely detection of dangling domain records is the main driver behind domain takeover attacks [110]. Worryingly, DNS itself is exposed to hijacking, while invalid certificates of the recursive resolver are also high [111], [112]. Furthermore, around 25% of DNS resolvers over HTPPs fail to meet minimum privacy requirements [111], [112].

D. Log-Based Proposals

SKI [23] tried to give control to domain owners over their PKs by defending against corrupted CAs. Each domain owner has two PKs: SKI key pair and TLS key pair. Domain owners cross-attest their TLS PKs by digitally signing them with SKI PKs. SKI PKs are recorded on specialized servers called "timeline servers", which are abundantly mirrored. When a user visits a domain server, it confirms the existence of the SKI key pair on a timeline server by visiting its mirror. The user accepts the server certificate if the SKI PK is witnessed on the timeline server and the CA-issued certificate is cross-endorsed by SKI PK. It eliminates browser warnings and is the first log-based approach to reduce the trust placed in CAs. However, it induces an extra latency in the handshake, raises privacy concerns, and does not support verifiable proof.

Google pioneered CT [5], which keeps a read-only history of CA-issued certificates in the form of transparency logs. CT introduced the following additional entities to mitigate SPoF and enhance traditional PKIX security.

- Transparency log: It is an auditable and append-only ledger maintained as a Chronological Merkle Hash Tree (CMHT). All CA-signed certificates are appended to it, and a Signed Certificate Timestamp (SCT) is inserted into each of them.
- Monitor: An entity that keeps a full copy of a transparency log and watches for suspicious entries by examining and validating each entry of the transparency log.
- Auditor: An entity that validates the correct functionality of a transparency log by checking that the certificates that the transparency log has promised to insert in the next update are present in the updated version of the transparency log. It can be a standalone entity or an integrated function of Web clients or monitors.

The logs are exposed to public scrutiny and monitoring to offer transparency and make CAs accountable to their clients for their conduct. The MHT is maintained in chronologically ascending order from left to right, enabling transparency logs to offer membership and extension proofs in logarithmic computational and communication costs. Any interested party can perform the role of monitor and auditor to watch transparency logs to identify if they are working correctly. CT can only detect malicious certificates and attacks through them, leaving revocation as an open problem. A typical PKIX framework with the Google CT log extension is depicted in Fig. 12. Eskandarian et al. [143] addressed the privacy issue of private subdomains caused by CT logs. Leibowitz et al. [144] addressed the security and revocation issues of CT. CTng uses Certificate Revocation Vectors (CRVs) [145] to complement CT revocation, while the threshold signature is used by monitors to sign the fingerprints of a log.

Accountable Key Infrastructure (AKI) [10] introduced a multi-CA-signed certificate managed on a transparency log to defend against the failure of a single trusted authority. It allows domain owners to assert their trust list of transparency logs and a list of CAs that can sign their certificates. Moreover, domain owners can define a threshold number of CA signatures on the multi-CA-signed certificate to be considered valid. Each domain owner in AKI acquires a multi-CA-signed certificate from the threshold number of CAs and registers the multi-CA-signed certificate on the transparency log. After the successful registration process, each domain gets membership proof for its certificate from the transparency log and staples the proof with the certificate. The staple proof shows that the certificate is valid and is refreshed each time the transparency logs update their database. AKI maintains a single certificate per domain in transparency logs under a lexical-ordered binary hash tree to offer a logarithmic certificate revocation proof. It introduces checks-and-balances among parties to watch each other's operations, and it identifies MitM attacks. However, each monitor needs to maintain a full copy of the transparency log to monitor the transparency log since the binary hash tree does not provide a logarithmic proof of extension from the previous one.

ARPKI [113], [114] identified attacks and loopholes in AKI and mitigated those loopholes by allowing a domain server to designate m service providers (e.g., transparency log and CA) during the certification process and request one signing CA to carry out the registration process in the transparency log. Each certificate is endorsed by no less than a threshold number of CAs, as in AKI, and the designated servers watch each other's operations to thwart MitM attacks even when m-1 parties are corrupted. The security of APRKI was proved using the Tamarin prover [146]. It eliminates the auditor's need for AKI and increases the resilience level against two compromised AKI authorities. However, the involvement of all parties in the certification and registration process slows down the performance of the system.

Policert [115] used the AKI multi-signature certificate concept and separated the list of trusted transparency logs and

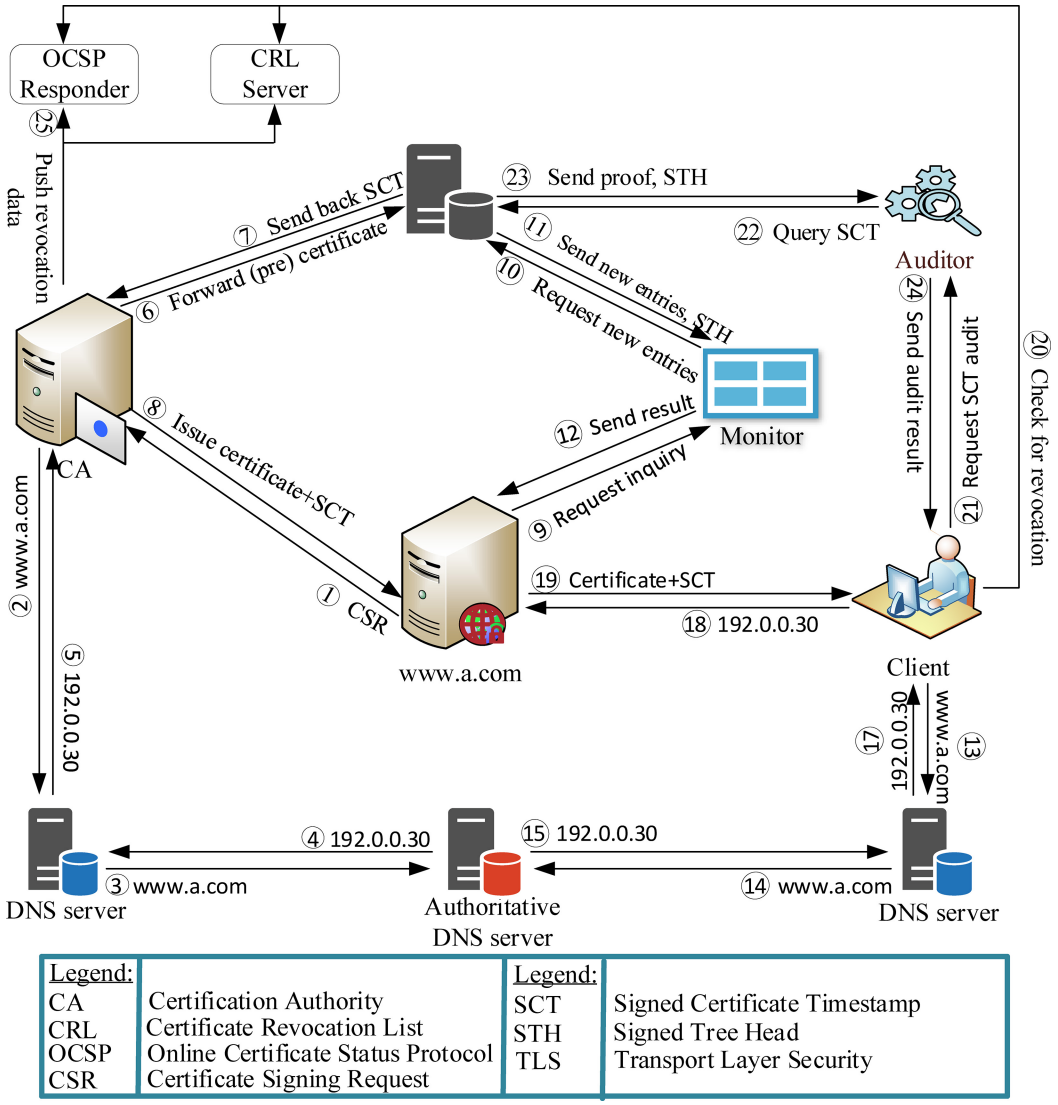


Fig. 12. A high-level overview of PKIX architecture with Google Certificate Transparency extension.

CAs from the TLS certificate by allowing domains to define a detailed policy for their use of certificates. The Policert protocol comprises policy registration, multi-signature certificate registration, certificate validation, transparency log proof auditing, and secure connection setup. Each domain defines the policy, binds it to the PK through attestation by multiple CAs, and registers on the transparency log. After policy registration, a multi-signature certificate is acquired from a pre-defined number of CAs and cross-attested with the policy key pair. Then, the multi-signature certificate is published on the transparency logs. Clients check policy parameters, the threshold number of CA signatures, and cross-signing by the policy-specific key pair. Various proofs are validated to ensure that the transparency log works correctly in the audit phase. The multi-signature certificate validation and proof audit phases are carried out during the TLS connection handshake. During multi-signature certificates, the policy-bound key pair is used to cross-sign each multi-signature TLS certificate, as in SKI.

DTKI [116] took advantage of cross-signing of the TLS certificate with the domain-specific key pair known as the master key. Clients only accept CA-issued certificates that are cross-signed by the master key. Hence, only CA and transparency logs cannot convince users to accept a PK unless they know the master key. The master and TLS key pairs are appended to the transparency log, which is maintained by a pair of MHTs as in CIRT [147]. DTKI has two layers of transparency log: Certificate Log Maintainer (CLM) and Mapping Log Maintainer (MLM). CLM stores all CA-issued PKs for a particular set of domains, while MLM holds the domain-to-CLM mapping information. Users must retrieve mapping information from MLM before connecting to a domain server. Like ARPKI, the security of DTKI is verified through the Tamarin prover [146]. However, DTKI does not offer any mechanism to recover the master key in the case of loss.

TripPKI [117] is a log-based PKIX proposal that introduces checks-and-balances among CA, transparency log, and DNS to ensure the detection of any single-party compromise. The checks-and-balances introduced in TriPKI are similar to checks-and-balances in AKI and ARPKI, except for introducing a DNS as a new party and threshold signature scheme. It

consists of initialization, certificate issuance, certificate update, and revocation in the case of a key leak/compromise. The certificate issuance phase also includes signing using a threshold signature, synchronization among the signing parties, and distribution to the requesting domain. TriPKI checks-and-balances among all parties during the certificate management process may cause extra latency and delay client connections to the domain. Furthermore, it is vulnerable to version attacks if CAs, transparency logs, and DNS collude together. We also observed that it has no mechanism to ensure the detection of version as in AKI and ARPKI.

Khan et al. [118] identified MitM attacks on Policert and mitigated the attacks through checks-and-balances and enhanced data structure. The protocol comprises certificate issuance, registration, revocation, and verifiable validation. The protocol supports CA hierarchy and sub-CA revocation through improved data structure and revocation policies. Furthermore, the protocol is verified using a Tamarin prover [146]. It was noted that ATCM introduces extra latency and is exposed to versioning attacks.

ARCT [119] addressed the weak identity verification and certificate miss-issuance process through collaborative identity verification and certificate examination before being issued and logged in the transparency log, respectively. In ARCT, each root CA maintains a separate transparency log that enforces sub-CAs to comply with community standards. ARCT consists of certificate issuance, registration, revocation, connection establishment, and cross-logging of log fingerprints owned by the root CA in the public logs. During certificate issuance, the collaborative identity verification process ensures that a set of CAs verifies the domain identity before issuing a certificate for the domain. At the same time, cross-logging of fingerprints prevents versioning attacks by transparency logs. It was identified that it could expose root CAs to DoS attacks if attackers take down a root CA-owned log.

SCM [120] is a hybrid scheme that leverages logs to manage domain server certificates, whereas the management of CAs and logs is handed over to a group of domain owners to balance power sharing in the PKIX architecture. Certificates of CA and transparency logs and fingerprints are managed on a blockchain platform. The protocol includes certificate creation, CA authorization, domain certificate publication, log update, publishing fingerprints of logs and revoked CAs to the blockchain platform, proof generation, auditing of logs, connection establishment, and domain certificate update. It was found that SCM does not support multi-domain certificates as well as induces additional latency to certificate issuance, which may result in the availability of domains.

Wang and El-Said [121] leveraged cosigning of the TLS certificate with the domain-owned master key. Clients only accept CA-issued certificates that are cosigned by a master key. Unlike conventional log-based proposals, each domain owns a log for logging its own master key and TLS certificates. Users need to retrieve certificate logging information from the domain-owned log before connecting to the domain server. Therefore, only CAs cannot convince users to accept a PK unless they know the master key and control the domain-owned log. Unfortunately, a client needs an extra connection to the transparency log each time the client connects to a domain. Moreover, Wang and El-Said [121] do not offer any mechanism to recover the master key in the case of loss.

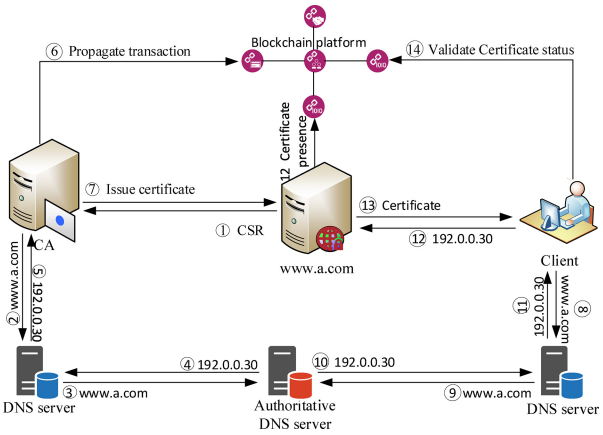


Fig. 13. A high-level overview of a typical blockchain-based PKIX architecture.

F-PKI [88] enables domains to assert policies for their certificates and users to place a different degree of trust in different CAs. F-PKI introduced a map server that maintains a comprehensive map of the certificates issued by the CAs supported by the map server. The map server aggregates certificate-related data and maintains the data in a sparse MHT. In addition, it provides meaningful certificate-related services to both domains and users. Users must retrieve certificate-related information from the map server before connecting to a domain. Therefore, it induces a connection delay in the authentication process. Fortunately, the certificate-related data provided by the map server complement the traditional authentication process as users gain a higher level of assurance by validating that there is no conflicting certificate for the communicating domain.

Lesson learned: Log-based proposals reduce the attack surface and window; however, they are still exposed to split-world attacks [128]. Furthermore, they increase the complexity of the certificate validation, implementation process, and operational overhead on CA and expose private domains to public audit (i.e., privacy concerns).

E. Blockchain-Based Proposals

Recently, blockchain emerged as an alternative to centralization by reducing trust in centralized authorities such as CAs. Fig. 13 illustrates a blockchain-based PKIX architecture, where certificate-related information is recorded on the blockchain platform. Blockchain-based proposals can be classified further as CA-centered, client-centered, and domain-centered.

1) CA-Centered Proposals: Matsumoto and Reischuk [148] investigated the lack of granting sufficient incentives to CAs for identity verification while issuing certificates. The work in [148] presented a new paradigm of certificate-as-an-insurance to keep CAs accountable for misconduct by using insurance policies and benefits bargained between the CA and the domain. Moreover, certificate-as-an-insurance tried to

automate the detection of CA misconduct and the subsequent reaction to the misconduct. CertChain [125] is another framework that conducts TLS certificate audits by introducing a new data structure to perform blockchain transactions. The data structure, namely CertOper, carries certificate-related information and operations such as certificate registration, validation, update, and revocation. It used the Bloom Filter (BF) to counter false positives during revocation validation and Ouroboros as a consensus algorithm, which selects a leader through a CA or a bookkeeper dependability rank. Users need to send a certificate verification request to a bookkeeper before connecting to a domain. Therefore, a dishonest bookkeeper can convince clients to accept fake certificates by sending incorrect responses to their certificate status validation queries [128]. Furthermore, bookkeepers can track the browsing history of clients [149]. PB-cert [149] addressed the privacy and storage issues of CertChain by separating the storage of the revocation history and the control plan. Additionally, the authors designed an efficient mechanism to obscure revocation response messages to client queries for the sake of hiding client privacy. It was observed that a dishonest bookkeeper could convince clients to accept fraudulent certificates, as in its predecessor.

Yakubov et al. [122] carried out the issuance, validation, and revocation of certificates on a blockchain platform. Each CA is instantiated on the blockchain platform by generating a smart contract for each CA. The smart contract includes the CA certificates and hashes of CA-issued certificates along with the revocation status. They inserted blockchain metadata into the certificate extension fields. They allowed CAs to set one of the extension fields during the issuance of certificates to include trust-related information, except for a root CA certificate. Later, the extension field is used in the authentication process to verify the certificate path/chain. The proposed PKI scheme does not specify how to ensure rigorous identity verification of root CA before contract deployment and does not support the detection of fake certificates due to storing only the hash values of end-entity certificates on the blockchain. Yasin Kubilay et al. [139] identified privacy and security vulnerabilities in the scheme of [122]. Korgan is based on a permissioned blockchain with an enhanced Practical Byzantine Fault Tolerance (PBFT) consensus method. The modified PBFT consensus with threshold signature is used to sign blocks that eliminate the need to trust external entities during the issuance and validation of certificates. The modified PBFT with threshold signature uses the concept that all nodes maintain a part of the signing key. The elected leader can add a new transaction after receiving signatures from other consensus nodes. Karaarslan and Adiguzel [150] shed light on the strength of the blockchain application to DNS and PKIX.

Wang et al. [123], [151] designed a blockchain-based certificate transparency and revocation framework. The framework inserts a CA-issued certificate and stapled revocation information in the blockchain transaction with a pre-defined validity period. Each domain server has two key pairs: the publishing key pair and the TLS key pair. The publishing key pair is initially certified by a group of web servers residing on the blockchain before the owner of the publishing key pair. After certification, the publishing key pair is used to insert the TLS certificate and revocation information into the blockchain. The TLS certificate and its revocation status insertion process are repeated periodically, since the transaction has a validity period. The TLS key pair (certificate) is then used during the authentication and signing process, and clients are presented with the TLS certificate and its authentication path of MHT. The authentication path verifies the revocation status of the TLS certificate through locally maintained block headers (MHT root hashes). Although the proposed work tried to make the revocation process accountable, it relies on old-fashioned CRL-based revocation by CAs and cannot prevent CAs from malfunctioning. Furthermore, it has a window of attacks during revocation of key leaks, where MitM can be launched against clients with revoked TLS certificates (in case of a key leak) having unexpired certificate transactions.

BKI [152] designed a blockchain-based PKIX framework in which a domain certificate is endorsed by multiple CAs, as in AKI and ARPKI. It includes the initialization, certificate issuance, update, and revocation phases, where the last three phases are carried out on a blockchain maintained by multiple trusted maintainers. BKI shields clients against a corrupted CA and eliminates versioning attacks. However, it does not specify any mechanism for offline verification of domain certificates. Moreover, clients need to contact blockchain maintainers to check the certificate validation status; hence, the maintainer can track the browsing history of clients.

Certificate Transparency using Blockchain (CTB) [126] leverages the Hyperledger Fabric blockchain of IBM to prevent CAs from issuing certificates for domains without getting the consent of domain owners. CTB incorporates a revocation mechanism to complement the CT revocation process. CTB strengthens the security of PKIX against corrupted CAs. Nevertheless, it does not preserve the main aim of CT as Hyperledger Fabric is a private blockchain. The private blockchain does not ensure public scrutiny and monitoring. It was observed that detecting fraudulent certificates is difficult once issued because of the design of the CTB and the private nature of Hyperledger Fabric.

Conifer [127] integrated CONIKS [153] and Catena log [154] to eliminate log- and blockchain-based PKIX issues. The architectural design of Conifer is similar to the enhanced CONIKS architecture proposed by [154]. Conifer enhanced CONIKS [153] design by leveraging two enhanced data structures: the transaction chain—linking the consistency of name-to-key binding to the Bitcoin blockchain and operation tree forest—enabling cost-effective and safe search operations against untrusted transparency logs. Conifer has advantages over CONIKS in eliminating the need for watching by transitioning the responsibility of auditing a name history from the name owner to everyone who watches the name and decreasing the communication cost through the efficient data structure.

Dykcik et al. [155] proposed BlockPKI that saves and manages CA-signed domain certificates on a blockchain. In BlockPKI, a domain sends a CSR to multiple CAs to endorse the domain certificate. The group of CAs uses a modified version of the Schnorr multi-signature algorithm to defend against CA failures and eliminate the weakest link security

issue of PKIX. The multi-CA signed certificate is then published on the blockchain platform, which clients can validate later. BlockPKI leaves certificate revocation as an open issue.

CertLedger [128] managed CAs and domain certificates on a permissionless blockchain to make both the CA and domain certificate management transparent to the public. It also mitigated the split-world attack issue of log-based PKIX proposals. A group of trusted members, named board members, share a signing key, and has the authority to insert a CA into the blockchain if the CA complies with standards and security requirements defined by them. This proposal can thwart attacks on the truststore of clients. However, group member management requires manual effort. CertLedger has no mechanism to block CAs from issuing valid but malicious certificates.

Xu and Joshi [156] instantiated CA on a distributed ledger to allow audit, introduce transparency, and enhance the trustworthiness of CA. The scheme comprises CA, ledger, owner, and client. The distributed ledger maintains a history of CA's operations, making CA auditable to the public per the system's security requirements. Zhao et al. [132] used smart contracts to instantiate a new party that works as a CA proxy to watch and manage the lifecycle of TLS certificates. They designed the proxy in a manner that can be easily integrated with the traditional CA ecosystem through blockchain Oracle services. The proxy helps the CA in the identity verification of domains, keeps metadata of TLS certificates on the blockchain, and uses the blockchain Oracle to bridge the CA proxy instantiated on the blockchain platform with identity verification and signing of the certificate.

Cecoin [157] instantiated PKIX on a blockchain platform. It consists of three entities: miner, certificate owner, and certificate user. The role of CA is replaced by miners, who use a proof-of-work consensus algorithm to record certificates. The protocol includes the registration of certificates on the blockchain platform, revocation, and renewal by the certificate owner. It also includes identity transfer from one entity to another through the identity assignment protocol. It was found that Cecoin does not specify the method of how the miner carries out proof of identity ownership.

Meta-PKI [140] distributed trust among multiple parties to make CA decentralize and connect parties through cross-certification. The cross-certification process uses Hyberledger Fabric smart contracts. Meta-PKI uses a layering approach and comprises three layers: the end-entity layer, the CA layer, and the meta-CA layer. End-entities request certificates, which CAs grant from the CA layer. The CAs at the CA layer are created by CAs from the meta-CA layer connected to the blockchain, which consists of a group of meta-CAs. Slepak [158] proposed a DNSchain that integrates DNS with Namecoin [159] to authenticate domain names. Kfoury et al. [90] instantiated Let's Encrypt CA on a blockchain platform to eliminate the SPoF problem. They created an ACME protocol based on an Ethereum blockchain to conduct the domain control validation and certificate issuance process on the blockchain platform.

Hwang et al. [133] proposed a semi-decentralized PKIX framework comprising four steps: request, clearing, auditing and appeal, and verification. In the request phase, they use a modified proof-of-violation cloud protocol to enable domains to supervise CA and ensure nonrepudiation between domains and CAs. In the clearing step, a CA computes the hash root of MHT and uploads the root hash along with the certificate-related information to the Interplanetary File System (IPFS) [160] once a day. In the auditing and appealing step, domains check the operations of CAs for correctness and upload data for missing information related to their certificates. In the last step, a user receives a certificate and MHT proof (uploaded by CA daily) from a domain. The user validates this information to test the validity of the domain certificate.

Sermpinis et al. [141] proposed a blockchain-based decentralized PKIX called DeTract that targets DV-certificates. DeTract comprises three phases: 1) identity creation, 2) certificate generation and update, and 3) revocation steps. In the identity creation phase, domain owners create digital identities and link them to their real identities using Uport[6]. DeTract domain owners create and update the certificates in the second phase, where new certificates are generated, and their hash values are stored in the blockchain while they are stored off-chain. A prefix is leveraged before the hash value of the certificate to differentiate between the certificate creation and the update process. Finally, a certificate is revoked either softly by draining the address balance to zero or hardly by sending a revocation transaction from a revocation address with a stated reason for revocation. When a client browser with an installed DeTract add-on initiates a connection to a DeTract domain server, the server provides the transaction ID and the download link of her certificate. The client downloads the hash value of the certificate and accepts it if it matches the hash value stored on the blockchain.

BPKI [142] was proposed to thwart domain name preemption cyber-attack caused by a corrupted CA. In this attack, a fake but valid certificate is issued to a domain before the owner of the domain applies for a certificate. BPKI introduces auditors to supervise CA operations during certificate issuance and registration. BPKI comprises a data layer, an extension layer, and an application layer. The data layer handles the structure and format of the transactions. The extension layer improves the transaction rate of the underlying blockchain through a delegated PBFT based on a verifiable pseudorandom function and double-chain structure. The last one designs the verification logic for certificate issuance, update, and revocation.

Lesson Learned: It was observed that the blockchain has the potential to address PKIX problems. However, scalability limits the application of blockchain when applied to the Internet PKIX. None of the existing work addresses this issue. It would be interesting to design a hybrid solution by integrating a log-based PKIX solution with blockchain, where CA-signed domain certificates are recorded on the logs, and then log fingerprints are propagated to the blockchain platform each time the logs update their database. Another interesting design would be a careful design of an alternative for PKIX by instantiating PKIX using blockchain, sharding, and layer 2 techniques. However, it may increase complexity and reduce transparency and security.

[6]https://www.uport.me

2) Client-Centered Proposals: PADVA [129] instantiated notary servers [16] on a blockchain platform and recorded every transaction on the ledger. By leveraging blockchain technology, notary servers are no longer trusted, as every action is exposed to public scrutiny. The protocol starts with the system setup phase, where a requester forwards a transaction to a notary server by sending a message consisting of a domain name, a whitelist of keys, a fee, and a reference time (optional). After accepting the validation request, the notary server publishes the list of domain servers on the blockchain platform. PADVA uses timestamping to ensure reliable monitoring of domain PKs. Like the traditional notary system, clients require an extra connection to validate the domain key, causing additional latency.

Tewari et al. [124] designed a cloud-based PKI based on blockchain technology to authenticate end-users to financial organizations, websites, and other services. The scheme eliminates the need to purchase expensive certificates and follow complex identity validation processes. It uses a password for end-user authentication. The end-user identity verification and certification process is carried out by organizations that act as intermediate CAs. End-user certificates are then locked in the blockchain ledgers used by organizations, which can be later used by other entities who wish to utilize them to validate and securely communicate with end-users on the Internet.

Lesson Learned: Clients still need to query blockchain-based servers (i.e., full node) to validate the certificate encountered, similar to traditional client-centered solutions. In general, the additional complexity and overhead of blockchain-based PKI due to which users need to query full nodes is characterized by two factors. First, the substantial growth of the blockchain's size makes it challenging to maintain the ledger on the client side. Second, the size of the handshake message increases significantly (e.g., 14kb size of the handshake message in the work [128]) if the certificate is stapled with related information showing the logging of the encountered certificate on a blockchain platform. No work in this class focused on a thin, lightweight, and PKIX-compatible client for blockchain-based PKIX.

3) Domain-Centered Proposals: Ethereum Name Service (ENS) [134] is the first solution for mapping human-readable Ethereum names to machine-readable Ethereum addresses. It comprises two smart contracts: registrars' smart contracts and resolvers' smart contracts. The first type of smart contract owns TLDs such as ".eth" or ".test", and maintains a history of all domains, including subdomains on the Ethereum chain, while the latter acts as a DNS resolver. BitAlias [161] offers a secure mapping of human-readable Bitcoin names to machine-readable Bitcoin addresses. ENS and BitAlias have fee and scalability problems caused by the underlying blockchains.

In [162], the authors built a framework, namely OneName, to allow users to register and create profiles based on their blockchain ID. In a similar manner, Onename is extended to Openname by basing its solution on a blockchain and performing authentication on a blockchain. Authledger [135] is a blockchain-based domain name authentication scheme using Ethereum smart contracts. It allows a client to trust that a CA can issue a certificate for the domain using blockchain technology. The Authledger approach explained how to authenticate certificates efficiently and reliably. Authledger lacks a specific implementation to show the implementation of the scheme in the real world.

Xiong et al. [130] proposed SSHTDNS to mitigate the security issues of the DNS scheme. They used the consortium blockchain to manage TLD and leveraged linkable ring signatures to hide identity and ensure fair voting of consortium nodes. SSHTDNS maintains data in multichain form, where the main chain maintains light information, while the subchains maintain complete information under each specific TLD. Moreover, they also integrate the sharding technique with the multichain structure to ensure high throughput. Unfortunately, SSHTDNS has redundant storage issues and uses an inefficient ring signature.

BB-PKI [131] addressed problems caused by RAs of CAs when attackers take them down. In BB-PKI, each domain ownership verification is carried out by several RAs. After verification, one of the RAs propagates the certificate signing request signed by them to the blockchain. After receiving the request, multiple CAs sign the certificate out-of-band. Certificates can be revoked by an issuing CA or by the domain. BB-PKI safeguards clients in the face of a stronger adversary capable of controlling n-1 entities involved in the verification and certification process. We observed that BB-PKI makes verification and certification longer and more tedious, which may not be acceptable by some CAs, such as Let's Encrypt CA.

Li et al. [163] proposed an RSA-based ring signature to conceal the identities of the signatories. Then, they integrated their signature protocol with a privacy-preserving blockchain-based PKI [164] to address the storage and leakage issues of client keys. Their proposed PKI comprises the registration and update phases. In the registration stage, peers generate registration and master key pairs, where PK and real identity information are recorded on the blockchain. After registration, peers get pseudonyms and key pairs in the update phase to hide their real identities during communication.

LightLedger [137] is a blockchain-based domain certificate authentication and validation mechanism for identity identification that is efficient, verifiable, and trustworthy without using a trusted third party. CAs are also held accountable for issuing and updating certificates under the proposed system, and their actions are visible and governed by smart contracts. Moreover, the scheme keeps track of several trusted CAs, each of which is linked to a certain blockchain domain. To put it another way, each CA must first determine whether it can be trusted to carry out the actual issuing process. For constrained devices, the proposed approach also includes a robust and scalable domain authentication scheme based on blockchain technology with privacy-preserving features (e.g., mobile, browser, and IoT devices). One of the limitations of this scheme is the exponential growth of blockchain data size due to the consideration of public blockchain.

DNSBA [138] enabled the mapping of blockchain addresses to associated domain names in a similar manner to the mapping of IP to the domain name. It offers RR in two forms: text

TABLE VI
THE SUMMARY OF PKIX PROPOSALS

Class	Proposals	Problem addressed	Strengths	Weaknesses
CA-centered	Kasten et al. [79]	CA scope	Reduces attack surface through malicious certificates.	Cannot protect clients against locally trusted CAs compromises.
	Braun et al. [81]	CA scope	Client-specific CAs identification based on clients' browsing history.	Monitoring client browsing can leak client browsing data.
	Braun et al. [82]	CA scope	Dynamic trust decision for different applications.	Monitoring client browsing can leak client browsing data.
	Braun et al. [83]	CA scope	Dynamic CAs' scope restriction.	Cannot protect clients against locally trusted CAs compromises.
	Classen et al. [84]	CA scope	Peer-to-peer CAs reputation.	Older CAs get higher scores than newly added CAs with higher levels of compliance with community standards.
	(DV++) [85]	Domain identity validation.	Prevented path poisoning attacks as well as on-path and off-path attackers.	Cannot protect clients against CAs' compromises.
	Syta et al. [86]	CAs compromise	Made trust decentralized among various parties.	No detection and prevention mechanism in practice.
	Wang et al. [89]	Identity privacy	Enabled CAs to validate identity without learning it.	Only applicable to email-based identity validation.
	Let's Encrypt [9]	Automate DV-certificate issuance.	Free of cost certificate issuance.	exposed to different cyber-attacks.
	Perl et al. [91]	CAs scope	Identified unused CAs in the TLS ecosystem.	Did not specify any mechanism to deal with when a client encounters certificates signed by these unused CAs.
	Jayaraman et al. [93]	CAs' key leakage	Distributing trust among several parties.	Higher signing cost.
Client-centered	Wendlandt et al. [16]	TOFU	Trust management at end-users level	Unavailability and privacy issues.
	DoubleCheck [94]	TOFU	Resolved privacy and fresh PK unavailability problems of Perspective.	Introduced extra latency to connection establishment.
	Marlinspike [15]	TOFU	Resolved privacy issue of Perspective through onion routing.	Does not support multiple PKs for a domain.
	CertLock [74]	CA change attack	Enables users to detect fake PK issued by a different region CA.	Expose to same CA-issued malicious certificate attacks.
	Crossbear [95]	MitM attacks detection	Detection of under attack servers.	Expose privacy of attacked servers.
	Policy Engine [96]	Trust decision	Enable user-centered local trust decision.	Does not protect clients against malicious certificates.
	CERTSHIM [98]	TLS/SSL vulnerabilities	Transparent validation process.	Guests and applications can bypass security policies.
	Certificate Patrol [100]	Malicious certificates detection	Offers lightweight extra layer of security.	Does not protect users on first connection to a server and does not support multiple certificates for a domain.
	TrusBase [99]	certificate validation.	It offers client-controlled certificate authentication.	Local adversaries and applications can bypass the authentication.
Domain-centered	HPKP [103], [104]	CA scope	Enables domains to assert a list of trusted CAs for certificate issuance.	Helpless against declared CAs issued malicious certificates.
	DANE [17]–[19]	Domain control over PK	Enables domains to declare domain-related information in their certificates.	Reliant on DNS servers security.
	CAA [20]	CAs scope	Allows domain to declare CAs authorized for signing their certificates.	Reliant on DNS servers security.
	TACK [107]	Trust model	Domains signed TLS certificates.	No security on the first connection to a website.
	Elaphurus [109]	TOFU	Addresses update and multiple PKs issues of PKP.	Lack mechanism to update pinned entries if a domain changes it's previous CA.
	Borgolte et al. [12]	Dangling DNS records.	Eliminates domain takeover attacks.	
Log-based	SKI [23]	CAs compromises	Defense against CAs compromises.	Leaks client connection information.
	CT [5]	CAs compromises	Exposes CAs' operation to public scrutiny.	Does not support certificate revocation.
	AKI [10]	CAs compromises	Defense against CAs compromises and enables domains to designate trusted CAs.	Monitoring needs to download a full copy of the transparency log.
	ARPKI [113], [114]	Counters CAs compromises	Defense against n-1 trusted parties collusion.	Slower system performance due to the participation of all parties in the certification and registration process.
	Policert [115]	CAs compromises	Empowers domains to define detailed policy for their certificate usage.	Monitoring needs to download a full copy of the transparency log.
	DTKI [116]	Oligopoly	Eliminates the oligopoly problem of log-based PKIX.	Clients need extra connections to retrieve mapping information before connecting to servers.
	TripPKI [117]	CAs compromises	Addresses denial of service attacks against CAs.	Slower system performance due to all parties' involvement in the certification and registration process.
	Khan et al. [118]	CAs compromises	Eliminates attacks on Policert and enhances log structure.	Vulnerable to versioning attacks.
	ARCT [119]	Identity validation	Collaborative identity verification by a group of CAs.	Exposed to denial of services attacks.
	SCM [120]	CAs compromises	Reduces certificate storage cost on blockchain and mitigates versioning attacks.	May cause unavailability of domains due to additional latency during certificate issuance.
	Wang et al. [121]	CAs compromises	Domains controlled certificate issuance.	Lacks mechanism to recover the master key in case of loss.
	F-PKI [88]	Trust model	Complementing authentication process by feeding additional authentication data from a third-party map server.	Induces extra latency to the authentication process.
Blockchain-based	Yakubov et al. [122]	Trust model	Manages clients truststore on a blockchain, thus eliminating malicious CA insertion attacks.	
	Wang et al. [123]	CAs compromises	Domain controlled certificate issuance and revocation.	Relies on old-fashioned revocation method.
	Tewari et al. [124]	End-users certification	Free of cost certificate issuance to end-users.	End-user real identity is learned by the organization.
	CertChain [125]	certificate auditing	Introduced improved data structure for certificate auditing.	A dishonest bookkeeper can breach the security and privacy of clients.
	CTB [128]	MitM prevention	Prevents CAs from malicious certificates issuance.	Difficult detection of malicious certificate after issuance.
	Conifer [127]	Auditing	Cost effective auditing of users' keys.	
	Certledger [128]	Split-world attack	Eliminates split-world attack and protects truststore against adversaries.	Still, a CA can issue malicious certificates.
	PADVA [129]	TOFU	Transparency to notary server-based trust management.	Extra latency.
	Xiong et al. [130]	DNS attacks	Multichain structure to ensure high throughput.	Redundant storage problem.
	BB-PKI [132]	Identity validation	Defense against stronger adversary capable of controlling n-1 parties.	Longer and tedious validation process.
	Zhao et al. [131]	Domain control validation	Watching and monitoring of malicious certificates.	Can not prevent malicious certificate issuance.
	Hwang et al. [133]	certificate auditing	Enabled domains to supervise CAs' operation.	Extra storage overhead and extra burden on CAs.
	ENS [134]	Mapping	Enabled human-readable Ethereum addresses mapping to machine-readable addresses.	Not avail
	Authledger [135]	Domain validation	User-based trust management.	Usability and performance issues.
	Lewison et al. [136]	Identity validation	Introduced new metrics for identity verification during certificate issuance.	
	LightLedger [137]	Identity validation	Smart contract-based scope restriction of CAs.	Enormous growth of ledger size.
	DNSBA [138]	Mapping	Mapping of blockchain addresses to domain names.	Does not consider the privacy of student information.
	Korgan [139]	Privacy	It eliminates privacy issues of Yakubov et al. [123] scheme.	It introduces extra latency to certificate issuance.
	Meta-PKI [140]	Privacy	Eliminates trust on external entities.	Does not achieve full transparency goal.
	Sermpinis et al. [141]	Mapping	Soft revocation of a certificate by draining its address balance to zero.	Additional latency for downloading off-chain certificates during each connection establishment.
	BPKI [142]	Name preemption attack	Introduces layering approach to fix low transaction throughput.	Additional delay due to the participation of multiple auditors in the certification process.

RR and blockchain RR. The authors used BIND9[7] to implement the blockchain RR and assessed their proposed solution for the registration of diploma of students. Table VI provides a comprehensive summary of the PKIX schemes.

Lesson Learned: The integration of blockchain and DNS is a fascinating topic of ongoing research due to its potential

[7]https:www.isc.org/bind

to solve many problems of conventional DNS. Unfortunately, the instantiation of DNS on blockchain may be limited by the well-known scalability of blockchain issues. Conventional DNS resolves billions of queries from Internet users, and blockchain-based DNS may not be able to handle such a large number of queries. Integrating blockchain and DNS may also bring some security issues, such as losing control over a domain in the case of a private key compromise.

F. Revocation Proposals

Many TLS/SSL certificate revocation schemes have been put forward to resolve the revocation problem of TLS certificates. They can be broadly classified into pull-based revocation, push-based revocation, network-assisted revocation, log-based revocation, and blockchain-based revocation.

1) Pull-Based Revocation Proposals: CRLs were introduced along with X.509 certificates in 1988 by ITU-T. Later, the specifications were made available as an RFC [165], and version 2 is based on the second edition of the X.509 recommendations published in 1993 by ITU-T and ISO [166]. This is one of the most widely adopted and straightforward methods for revocation of keys. CRLs are timestamped lists containing the serial number of revoked certificates, which are signed and released periodically by CAs. They are provided freely and support the offline validation of the revocation status. Clients check the serial number of certificates on CRLs while verifying TLS certificates. The primary benefit of this revocation mechanism is that the CRL distribution does not require trusted communications and server systems [167]. Delta-CRLs were used to reduce the size of CRLs and ensure the freshness of the revocation information [165]. Cooper [168] mitigated the explosion of CRL requests and CRL size by introducing overissued CRL and segmented CRL concepts, where CRLs are replicated on different servers. Cooper [169] introduced sliding-window-based delta-CRLs that combine delta-CRL with an over-issuing technique to mitigate the request explosion problem of delta-CRLs. The blacklist CRLs [170] used the idea that the issuers would watch the size of the CLRs or a clock. Issuers invalidate all certificates regardless of validity status and reissue them when the CRL size or the clock approaches the threshold. The blacklist contains re-issuance time, which shows that all certificates issued before that time are revoked.

Micali proposed the Certificate Revocation System (CRS) [171] to improve CRL communication overhead and enhanced it in 1996 [171]. CRS uses online/offline signatures and generates a signed message for each TLS certificate, showing whether it was already revoked or not. CRS associates two 100-bit numbers, Y_t and N, with the TLS certificate to show "non-revoked" and "revoked" status, respectively. CA associates and computes these two values using a one-way hash function (H) as CA determines the lifetime interval t for the certificate and generates two (pseudo) random numbers, Y_0 and N_0 representing the non-revoked and revoked status, respectively. CA uses H to get $Y_t = H^t(Y_0)$ and $N = H(N_0)$. CA performs two operations to update revocation information, such as for a non-revoked certificate, it sets entry to $Y_{t-i} = H_{t-i}(Y_0)$, where i $(0 \leq i < t)$ is a counter variable, and sets it to N_0 for revoked one. Micali [172] further revised and improved the CRS by incorporating one minor and one major modification. The minor one used SHA [173] as a hash function, and the major one replaced the centralized directory with distributed management.

Aiello et al. [174] enhanced the CRS scheme by decreasing update costs on the revocation maintainer while maintaining its reduced query costs. In contrast to CRS, where 1 token per certificate was inserted, their hierarchical and generalized scheme inserts $logN$, $(2^{c-1}-1)log_cN$ tokens instead, respectively, where N represents the number of valid certificates and R represents the revoked ones. On the other hand, their scheme reduces the update cost on the revocation directory maintainer from $N-R$ to $Rlog(N/R)$ and $Rlog_c(N/R)+R$, respectively.

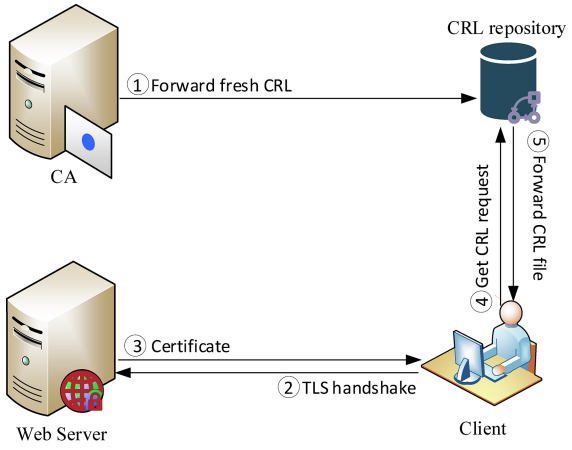
Kocher [175] presented the Certificate Revocation Tree (CRT) using MHT [176] to revoke TLS certificates. The idea behind using MHT is as follows: given any serial number of a TLS certificate, the prover can offer a short proof of the status of the TLS certificate that the verifier can verify. CRT is maintained as MHT, where the leaves correspond to statements about the serial number of the TLS certificate X_{SNo} issued by a trusted CA, say CA_X. CA_X breaks its certificate list into serial ranges, indicating the revocation status of certificates. For example, the range (7, 15) specifies that certificate 7 is revoked, but any certificate with a serial number greater than 7 and less than 15 is not revoked yet. ValiCert Inc. deployed and implemented CRT commercially.

Nissim et al. [177], [178] extended CRT by replacing 2 child nodes of each internal node in CRT with 2 or 3 child nodes and is named 2-3CRT. 2-3CRT leaf node indicates the serial number of the revoked certificate in sorted form, and the path from the leaves to the root has the same length. Additionally, it solves the reconstruction problem of the CRT during each update.

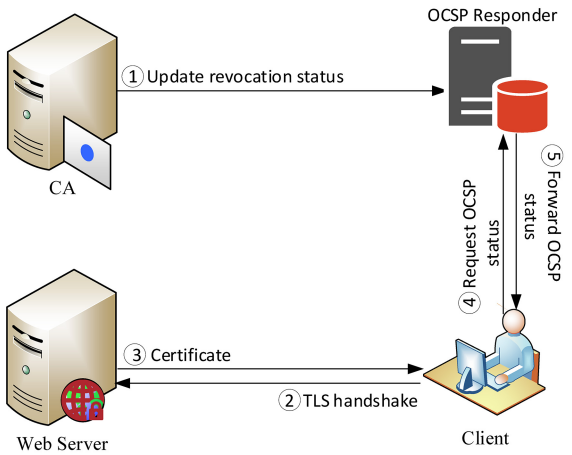
The Online Certificate Status Protocol (OCSP) [179] introduced a trusted third party named the OCSP responder. It provides online certificate revocation information and ensures real-time status validation. When a client queries a certificate status, the OCSP responder sends a signed response message showing its validity condition. ValiCert, Verisign, and Entrust have commercially implemented it. The OCSP responder has scalability issues in the case of a heavy system load, as the responder must sign each response. Furthermore, the respondent can violate the privacy of clients [180]. Fig. 14 displays CRLs and OCSP revocation schemes.

Window Certificate Revocation (WCR) [181] used the reference locality principle (i.e., the verifier might query the same certificate) to rescind the certificate. In WCR, each certificate mentions its release period $((1/T))$ and an eviction window size s that asserts the time when the certificate would be inserted in the scheduled release of CRL after revocation. Additionally, the verifier defines a vulnerability window when sending online status queries. The revocation and vulnerability window helps to reduce the size of the CRL.

Buldas et al. [182] proposed an Authenticated Search Tree (AST), similar to MHT, to introduce accountability to certificate management. They reduced the communication and



(a) CRL-based revocation



(b) OCSP-based revocation

Fig. 14. An overview of pull-based revocation schemes.

validation costs of certificates using an efficient tree. It includes a hashing algorithm to generate a short digest for a given set of certificates, a prover algorithm to generate short proofs for certificates (non)existence, and a verifier algorithm that validates the proofs. It exposes CA operations to public scrutiny and reduces the trust placed in them. However, there is no working scheme based on it because of the complex theory and open questions raised in the paper.

Easy Fast Efficient Technique (EFECT) [183] was presented to replace the individual signing of the certificate by a CA. It uses the Certificate Verification Tree (CVT) to maintain all certificates in a binary tree. A leaf node is a certificate statement, and CA timestamps the certificate along with the authentication path. When a client discovers a certificate, the client is provided the certificate in question along a signed authentication path. One of the drawbacks of this scheme is that newly acquired certificates undergo an unavailability period of update (once a day) [184].

Faldella and Prandini [185] proposed one-way accumulator-based revocation management to replace the signing of each response by the OCSP responder. In this case, the membership proof is constructed as in the CRT for valid certificates, and the digest is made public for verification. They showed that their scheme maintains the same performance as OCSP by comparing it with OCSP in terms of security, revocation update timeliness, and scalability while mitigating the computation load on the responder.

Wright et al. [186] used a directed root-based acyclic graph to disseminate revocation information. Wright et al. [186] used the property of a k redundant graph, where each node and its dependent nodes have k parents, except the root node of the graph. The k redundancy ensures fault tolerance in the worst situation, where a path exists from the root to other nodes even in the face of removal of k-1 non-root nodes from the graph. The scheme works in a peer-to-peer manner, where each new joining user must rely on the root or k already existing normal nodes. After the network is formed, the participants push the revocation data and update each other instead of relying on a trusted centralized server.

Boneh et al. [187] designed the SEcurity Mediator (SEM) framework to ensure rapid revocation. The scheme is based on a threshold type of RSA algorithm named mediated RSA. A CA generates key pairs and divides the private parts of key pairs into two parts: one granted to users and the other maintained by the SEM maintainer. SEM cancels users' certificates by voiding the private key part; thus, users cannot sign, encrypt/decrypt messages, since they need to cooperate with SEM for the second part of the private key each time while signing and encrypting/decrypting messages. The SEM architecture ensures prompt revocation; however, it has trust and privacy issues, as SEM can learn the communications of users [184].

Munoz et al. [188] implemented the Nissim et al. [177], [178] schemes in the Java programming language. They also mitigated the issues left by Naor and Nissim [177], [178] in their original work, such as revoking a certificate, deleting an expired certificate, and responding to a client request. Elwailly et al. [189] presented a QuasiModo tree similar to MHT with two major differences to ensure constant verification and proof size of O(1). To say the first difference, the leaves are hash chains of length 2. Regarding the second difference, their design supports proof generation with alternate internal nodes of the QuasiModo tree.

Lesson Learned: Generally, clients in pull-based schemes need to cache revocation information and responses to improve revocation check time. It was found that pull-based schemes generally lead to soft failure if clients cannot retrieve the revocation information in case the request is not cached or expired, which exposes them to MitM attacks.

2) Push-Based Revocation Proposals: CRLSets [190] are used by Google to push periodic updates to Chrome in the form of a small list of revoked certificates. CRLSets are built into Chrome and filter Extended Validation (EV) certificates. The maximum size of CRLSets is set to 250 KB and has the capacity to hold 40,000 revoked certificates. CRLSets protect clients against EV certificates; hence, they do not defend clients against DV-certificates. Similarly, Mozilla generates a list of revocations of intermediate CA certificates called OneCRL, which has a greater negative impact if abused [191].

Larisch et al. [192] based the CRLite revocation scheme on the BF cascade to reduce the size of the CRL. CRLite enables clients to download revocation information in a compressed

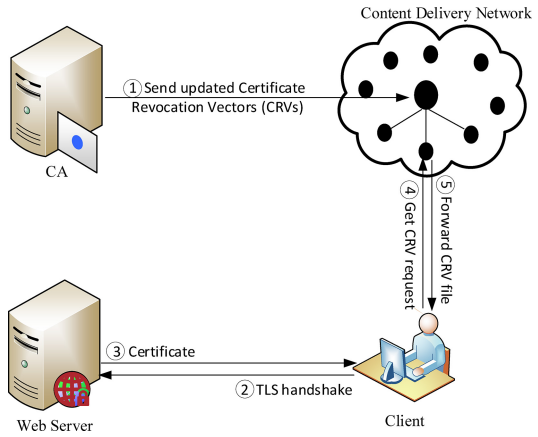


Fig. 15. An overview of a push-based revocation scheme.

and deterministic form. The cascade filters false-positive queries in another BF that checks for the opposite queries. The filtering process is repeated until the BF is free from false positive anomalies. The construction of the filter cascade requires checking the complete set of valid and evicted TLS certificates using the data structure while consuming enormous computational and communicational resources.

Let's Revoke [145] is another push-based method that minimizes the revocation overhead, especially the communication cost. It uses bit-vector-based CRVs to achieve communication efficiency. CRVs use binary bits to show a certificate revocation status (0-no, 1-yes), and each CA is assigned one CRV. Furthermore, the authors also proposed a new identifier for each certificate, which consists of three parts. The three parts show the issuing CA, the expiry date, and the revocation number. The last part, known as the revocation number, is assigned by the issuing CA. Revocation of a certificate requires the issuing CA to set 1 at the location of the revocation number. Fig. 15 illustrates Let's Revoke as a demonstration of a push-based revocation scheme.

Lesson Learned: It was learned that push-based revocation schemes are generally bandwidth-extensive. It is worth noting that, despite being bandwidth-intensive, CRLite has made improvements over CRL and OCSP and is currently being adopted by Mozilla.

3) Network-Assisted Revocation Proposals: Short-lived certificates [193], [194] were introduced to eliminate certificate revocation checks. Short-lived certificates have a short life ranging from a few hours to a few days [194], [195]. This technique requires domains to change their TLS certificates [195] frequently. Eliminating the need for revocation validation does induce an additional burden on CAs, CT logs, and monitoring parties.

OCSP stapling [196] was proposed to enforce servers to download and staple certificate status information to their certificates. Stapling eliminates the need to download and test the status of certificates online on the client side. This approach mitigates the delay and privacy concerns of conventional OCSP responders. However, attackers can conduct MitM attacks against clients by bypassing OCSP stapling, since attackers can conduct downgrade attacks by offering TLS certificate missing OCSP stapling while handshaking with clients. OCSP Must-Staple [197] addressed downgrade attacks by enabling browsers to block connections for certificates missing OCSP stapling through the X.509 certificate extension check. Must-Staple eliminated the security loophole of OCSP stapling through the certificate extension. However, it may result in server unavailability if the administrator fails to update the stapling information correctly.

Hu et al. [198] proposed the Certificate Revocation Gaurd (CRG), which uses the idea of a middlebox. The middlebox intercepts TLS traffic and tests certificate revocation status using OCSP requests for end-entities, such as an organizational gateway. If the encountered certificate is revoked, an invalid certificate is returned, and the connection is blocked. However, this approach exposes mobile nodes, such as phones and laptops, to MitM after leaving the network secured by CRG.

Revocation In The Middle (RITM) [199] depends on the middlebox as in CRG. However, it uses CDN to distribute revocation information to middleboxes. The middleboxes intercept TLS connections, test the validity status of certificates, and inject the validity information into TLS connections as a TLS extension. It eliminates the delay caused by the revocation status check, as the boxes installed along the connection route staple the information. Unfortunately, it requires the installation of middleboxes throughout the infrastructure, an extra CDN layer to update, and changes on the client and server side to operate. Fig. 16 illustrates the OCSP stapling and RITM schemes.

RevCast [200] used an FM radio to disseminate revocation information to multiple users. Each CA broadcasts certificate revocation data to clients on FM radio after the eviction of a certificate. Therefore, client devices need integrated hardware receivers to get the information. If a client loses a transmission, the client must download CRLs over the Internet.

SecureGuard [201] was proposed to complement certificate validation during TLS handshakes. It uses middleboxes, called collector servers, to push revocation information to ISPs. The proxy servers of the ISPs maintain a cache of domain certificates, and clients access the Internet through the proxy servers. CAs update collector servers in real-time when they revoke any TLS certificate. The collector servers, in turn, relay the information to the ISPs' proxy servers. Proxy servers intercept connections along with TLS traffic and alert clients about the validity of domain certificates.

Lesson Learned: It was noticed that revocation checking is an overcomplicated process in PKIX. Short-lived certificates can greatly simplify this overcomplicated process if the server side is modified to renew and install short-lived certificates without manual effort.

4) Log-Based Revocation Proposals: Revocation transparency (RT) [202] extended Google CT to deal with certificate revocation. Two mechanisms are presented to store revocation data transparently. The first one relies on the sparse MHT, which is the MHT with most nodes initialized with 0. Each proof consists of 25 hash values ending in 1 or 0 leaf nodes based on the eviction status of the certificate. Revoking a certificate requires altering the sparse MHT to end in 1 and then inserting the record into the Google CT log. The second

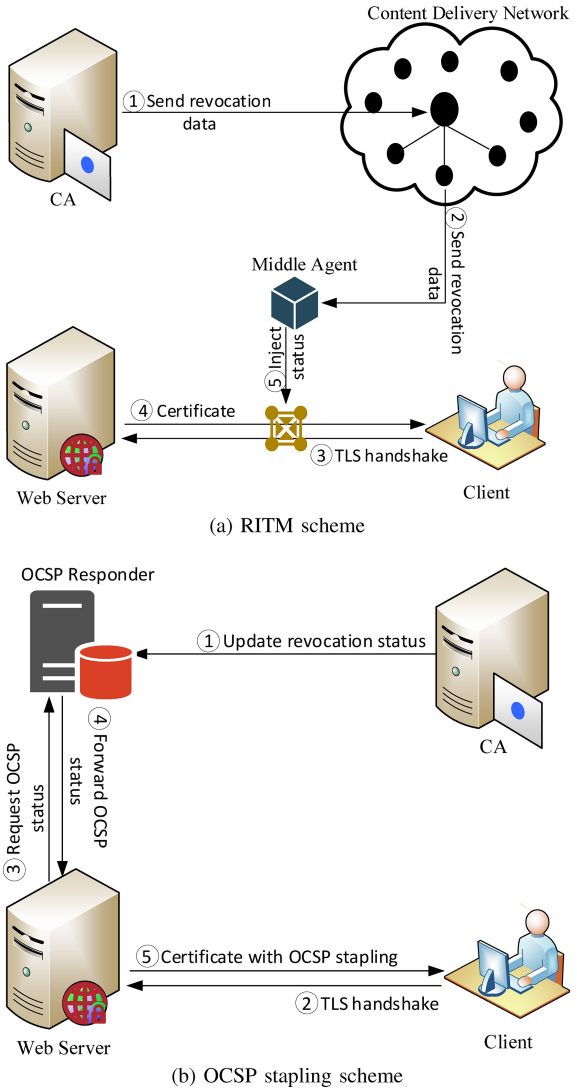


Fig. 16. An overview of network-assisted revocation schemes.

one is based on creating a sorted list of certificates in the form of a search tree. However, the revocation check in both mechanisms remains linear rather than logarithmic in the number of TLS certificates. Fig. 17 shows a log-based revocation process.

CIRT [147] introduced lexical ordered MHT, which can provide efficient proof of non-membership of a certificate. Thus, the CIRT log can efficiently provide the proof of membership, non-membership, and extension that states the current version is an extension of the previous, in contrast to the CT log, which can only provide proof of membership and extension.

PKISN [203] addressed the CA certificate revocation problem by designing an efficient method to avoid collateral invalidation of TLS certificates caused by a CA revocation. They based their solution on a public transparency log. They extended the traditional log and revocation request through timestamping, which makes it possible to revoke a CA certificate without invalidating legitimate domain certificates issued before that particular CA compromise time. They also proposed a design to better express the hierarchical structure of the certificate chain in the context of public transparency logs.

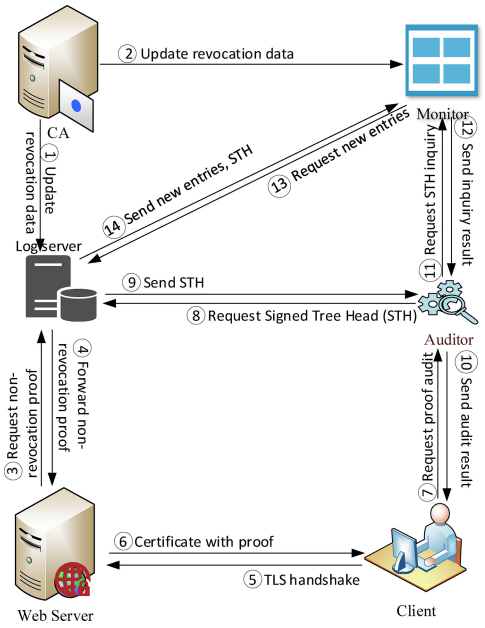


Fig. 17. An overview of log-based revocation schemes.

Singh et al. [204] extended the CT log by making the proof shorter than the CT log. They maintained their log using two types of tree structure: MHT and accumulation tree. The MHTs are maintained and offered proofs as in CIRT, while the accumulation tree is used to provide constant short proofs of currency and non-membership. The accumulated value is inserted as the last node in the MHT.

Lesson Learned: It was learned that log-based revocation schemes generally enable clients to watch and monitor the revocation information by exposing them to public scrutiny. However, the designed log has a higher monitoring cost, as monitors need to download all entries from the log.

5) Blockchain-Based Revocation Proposals: BlockVoke [205] designed a decentralized certification revocation, allowing certificate owners and CAs to revoke certificates and distribute revocation information rapidly. The proposed scheme also allows for the revocation of CA root certificates, which is impossible with traditional approaches. Using a blockchain as an underlying layer ensures the continued availability and immutability of revocation information. BlockVoke interacts favorably with approaches, such as CRLite and CRVs, allowing organizations to update revocation filters with a little delay according to the requirements of their security policies. BlockVoke has several limitations. First, it lacks details on dealing with privacy issues while implementing the proposed system on Bitcoin and Ethereum. Second, it lacks a proof-of-concept implementation to demonstrate its practicality. Third, the authors did not provide details on the reward/incentive cost to make it faster and more efficient.

Process [206] designed a protocol for revocation status validation on a blockchain platform. Process integrates Counting Garbled BF (CGBF) with the blockchain to ensure timely updates and privacy of clients by querying the revocation status from the blockchain nodes. Additionally, the authors designed

TABLE VII
THE SUMMARY OF REVOCATION PROPOSALS

Class	Proposals	Problem addressed	Strengths	Weaknesses
Pull-based	CRL [165]	Revocation	Simplest revocation method with offline validation support.	Size and search time increase linearly with an increasing number of revoked certificates.
	Blacklist CRLs [170]	CRL size	Reduced CRL size.	Exerted extra pressure on certificate issuers.
	Cooper [169]	CRL request explosion	Reduced delta-CRL request explosion through over-issueing technique.	Over-issuing causes distribution issues.
	CRS [171]	CRL size	Enables revocation servers to respond promptly to revocation queries.	Increases communication costs from CAs to distribution centers.
	Aiello et al. [175]	Communication cost	Reduces communication cost from CAs to distribution centers.	Large certificate size.
	Nissim et al. [179]	Update cost of MHT	Search, insertion, and deletion operations are logarithmic in the number of elements.	Needs both CAs and revocation distribution centers to construct the revocation tree.
	Kocher [176]	Scalability	Commerically used by ValiCert Inc.	Update might require a complete construction of MHT.
	OCSP [180]	Revocation distribution	Supports online real-time revocation status validation.	Scalability and privacy issues.
	WCR [182]	CLR size	Optimizes average case cost at the cost of exceptional cases.	Complex and complicated revocation management that could limit its deployment.
	Buldas et al. [183]	Accountability	Reduces communication and validation cost.	Complex theory made it difficult to implement.
	EFECT [184]	Revocation stapling	Batching of certificate revocation status information signing.	Unavailability issues for newly acquired certificates.
	Boneh et al. [188]	Prompt revocation	Eliminates revocation checking on client side.	Trust and privacy issues.
Push-based	CRLSets [191]	EV certificate revocation	Smaller size CRL.	Does not offer DV-certificate validation.
	OneCRL [192]	Intermediate CAs revocation	Prevents intermediate CAs from abusing system resources.	Works only on the Mozilla Firefox browser.
	Larisch et al. [193]	CRL size	False positive free BF-based light CRL design.	Needs additional computational resources to construct a filter cascade.
	Let's revok [145]	CLR size	CRVs reduces CRL size through CA centered revocation mechanism.	
Network-assisted	Short-lived certificates [194], [195]	Revocation checking	Eliminates revocation checking on the client side.	Increases the certificate issuance rate on the CA side.
	OCSP stapling [197]	Privacy	Eliminates the need to download certificate status information online.	Exposed to version downgrade attacks.
	RevCast [201]	Revocation distribution	Ensures updated revocation data through broadcast.	Clients need an additional FM receiver to receive information.
	OCSP Must-Staple [198]	Version downgrade attack	Successfully mitigates version downgrade attacks.	Server unavailability issues in case of update failures.
	Hu et al. [199]	Timely revocation information delivery	Assists end-users during certificate revocation status checking.	Exposes mobile devices to MitM attacks through the revoked certificate after leaving the CRG secured network.
	RITM [200]	Timely revocation information delivery	Reduces the delay caused by the revocation validation check.	Needs changes on the client and server side.
	SecureGuard [202]	Revocation	Reduces trust in trusted parties.	Proxy can learn about the client's connection.
Log-based	RT [203]	Revocation	Introduces transparency to certificate revocation.	
	CIRT [147]	Revocation	Complements Google CT revocation process.	Expose to split-world attack.
	PKISN [203]	Intermediate CA revocation	Makes it possible to revoke intermediate CAs without causing collateral damage.	
	Singh et al. [204]	Revocation	Complements Google CT revocation process and introduces short proofs.	Expose to split-world attack.
Blockchain	BlockVoke [205]	CA revocation	Compitable with CRLite and CRVs.	Privacy issues.
	Process [206]	Timely update	Preserve the freshness of user data and privacy.	
	Adja et al. [207]	Revocation information distribution	Ensure transparency and efficient revocation status validation.	Higher delay.
	CRchain [208]	Validation and storage cost	Enables domains to revoke their keys without CA involvement.	The ledger is mutable, so a forging attack is easy to conduct.

a Block-Oriented Revocation List (BORL) to mitigate the division and forking of CGBF updates. The BORL is maintained in the form of a double-linked list.

Elloh Adja et al. [207] used the extension field of the X.509 certificate to insert a revocation-related field that denotes the distribution center on which the certificate will be recorded in the case of revocation. Each distribution center maintains revoked certificates in BF form. The BF and the revoked data are saved in a permissionless blockchain platform. Furthermore, the authors designed Revocation Status Information (RSI) to complement the validation process when the BF returns a positive response. However, the RSI mechanism induces a very high latency, even higher than CRL and OCSP [208].

CRchain [208] is another blockchain-based certificate revocation protocol that aims to reduce storage cost and validation latency. The authors designed a dual cuckoo filter called validCertCF and revokedCertCF to save the fingerprints of valid and revoked certificates and support the dynamic deletion of elements. CRchain also includes a server and CA co-controlled revocation mechanism that enables the server to revoke its compromised certificate without the involvement of CA. However, the CRchain does not preserve the immutability of the ledger, as the revocation data can be deleted dynamically. Table VII provides a summary of the revocation schemes.

Lesson Learned: It was learned that blockchain-based revocation schemes violate the privacy of clients during revocation checks, in general.

VI. Comparison

This section presents a set of evaluation metrics before moving on to the comparison that would help to thoroughly examine the reviewed schemes. After conducting the systematic and in-depth comparison, the lessons learned and future research directions are presented.

A. Evaluation Metrics

Detects malicious certificate issuance: A PKIX scheme should incorporate the ability to detect valid but malicious PK issued to end-entities. We assign partial points to schemes that locally support detection but do not offer public detection and monitoring facilities for malicious issuance.

Prevents malicious certificate issuance: Schemes offering this feature prevent a single party from issuing a valid but fake certificate and guarantee a higher level of security.

Connection privacy: Schemes that offer this feature hide browsing and validate the certificate history of clients from third-party access.

Conceal identity: This privacy level is offered by schemes that validate and demonstrate ownership of the identity without learning it.

Extra connection: Schemes that provide this feature require support from servers during the validation of the certificate.

Client side changes: This feature shows modifications needed to client browsers/applications to take advantage of the security offered by a scheme.

Server side changes: This metric shows the modification needed on the server side to take advantage of services offered by a scheme.

CA side changes: This metric shows the modifications needed on the CA side to take advantage of the services offered by a scheme. Experience has revealed that a scheme is less likely to gain major adoption by the Internet community if it requires considerable changes to the business model of CA, but has limited financial incentives. The same applies to server side changes for a scheme with higher maintenance costs.

Limited trust: Schemes that offer this feature rely on semi-trusted centralized servers instead of relying on fully trusted centralized servers for certificate management.

Decentralized trust: Schemes that offer this feature do not have centralized trusted servers. Trust is distributed among various parties.

No TOFU: Schemes that offer this feature do not trust an unknown PK the first time it is encountered.

CA revocation: Schemes offering this feature have unique mechanisms and parties to invalidate the certificates of intermediate and root CA.

No extra burden on CA: Schemes offering this feature do not require CA participation beyond the regular certificate management process. For example, ARPKI does not support this feature, as CAs are involved in certificate registration and monitoring beyond the standard certificate management process.

Scalability: This metric measures the ability of a scheme to support current and future identity management load in the case of Internet expansion.

No infrastructural changes: Schemes that offer this feature do not change the business model of the CA, the server, and the client. Such schemes offer services without altering the underlying infrastructure (e.g., certificate issuance and changes to the TLS handshake mechanism).

B. Comparison Through Evaluation Metrics

This section provides a detailed evaluation of PKIX and revocation proposals and their modern implementations using the aforementioned evaluation metrics.

1) CA-Centered Proposals: As can be observed from Table VIII, this class performs identity verification before issuance, but lacks a post-miss-issuance detection mechanism. Therefore, it partially supports this feature (e.g., Kasten et al. [79] alert users if a restricted CA issues a malicious DV-certificate for a domain) except Let' Encrypt and Wang et al. [89] schemes. The schemes in [86] and [93] can detect and prevent because they do not rely on a single authority for verification and certification. This class does not require clients and servers to change their business models, except the scheme in [86]. The work in [86] also substantially alters the business model of clients, servers, and CAs, as clients and servers need to maintain an aggregated group PK, and the CA role is split between multiple parties. The remaining schemes require slight modifications to the CA business model. All proposals do not support the "limited trust" and "decentralized trust" features, as CAs are fully trusted and responsible for verification and certification, except the scheme in [86], while the scheme in [93] supports the "limited trust" feature. CA-centered proposals fulfill "no TOFU" and leave "CA revocation" features unfulfilled. They are scalable and slightly alter the underlying infrastructure, except for the scheme in [86].

2) Client-Centered PKIX Proposals: Perspective, DoubleCheck, and Convergence in this class are based on multipath probing, where clients accept PKs witnessed by certified observers. The remaining three use client side PK pinning to protect clients from malicious certificates. Thus, multipath probing-based proposals can detect malicious PKs at the cost of an extra connection, while the pioneer Perspective also learns browsing history. This class does not require changes on the server and CA side. However, clients must install and maintain the add-ons, thus partially fulfilling the "client side changes" metric. This class is entirely based on trusted servers for certificate management. The first three proposals visit certificate observers upon encountering a new certificate, while the latter three leave "no TOFU" metric unfulfilled. It lacks the CA revocation method while fulfilling the features in the context of "no extra connection, scalable" and "no Internet-side changes".

3) Domain-centered PKIX proposals: In this class, the server-based pinning schemes, i.e., PKP [103], [104] and TACK [107], rely on blind TOFU, the DNS-based pinning proposal; that is, DANE relies on DNSSEC, while Elaphurus [109] integrates multipath probing with server-based pinning. Therefore, server-based pinning schemes partially fulfill the "detects malicious certificate issuance" feature,

TABLE VIII
A COMPARISON OF LEADING CONVENTIONAL PKIX, REVOCATION PROPOSALS, AND THEIR MODERN IMPLEMENTATION ON BLOCKCHAIN AND DISTRIBUTED LEDGERS BASED ON METRICS DEFINED IN SECTION VI. NOTE: ●=SUPPORTS THE FEATURE; ◑= PARTIALLY SUPPORTS THE FEATURE; ○= DOES NOT SUPPORT THE FEATURE

Class	Subclass	Proposals	Detects malicious certificate issuance	Prevents malicious certificate issuance	Connection privacy	Conceal identity	No extra connection	Client side changes	Server side changes	CA side changes	Limited trust	Decentralized trust	No TOFU	CA revocation	No Extra burden on CA	Scalable	No Internet-side Changes
PXIX proposals	CA-centered	Kasten et al. [79]	◑	○	●	○	●	●	○	○	○	○	○	○	●	○	
		Braun et al. [81]	◑	○	●	○	●	●	○	○	○	○	○	○	○	●	○
		Braun et al. [83]	◑	○	●	○	●	●	○	○	○	○	○	○	○	●	○
		Classen et al. [84]	◑	○	●	○	●	●	○	○	○	○	○	○	○	●	○
		Syta et al. [86]	●	●	●	○	○	●	●	●	●	●	○	○	●	◑	●
		Wang et al. [89]	○	○	●	●	●	●	○	●	○	○	○	○	◑	◑	○
		Let's Encrypt	○	○	●	○	●	○	◑	●	○	○	○	○	○	●	○
		Jayaraman et al. [93]	●	●	●	○	●	○	○	●	●	○	○	○	●	◑	○
	Client-centered	Wendlandt et al. [16]	●	○	○	○	○	◑	○	○	○	○	●	○	●	◑	●
		DoubleCheck [94]	●	○	●	○	○	◑	○	○	○	○	●	○	●	◑	●
		Marlinspike [15]	●	○	●	○	○	◑	○	○	○	○	●	○	●	◑	●
		CertLock [74]	◑	○	●	○	●	◑	○	○	○	○	○	○	●	◑	●
		Policy Engine [96]	◑	○	●	○	●	◑	○	○	○	○	○	○	●	◑	●
		Certificate Patrol [100]	◑	○	●	○	●	◑	○	○	○	○	○	○	●	◑	●
	Domain-centered	PKP [103], [104]	◑	○	●	○	●	○	◑	○	○	○	○	○	●	◑	●
		TACK [107]	◑	○	●	○	●	○	◑	○	○	○	○	○	●	●	●
		DANE [17]–[19]	◑	○	●	○	●	○	◑	○	○	○	●	○	●	●	●
		Elaphurus [109]	◑	○	○	○	○	○	◑	○	○	○	●	○	●	●	●
	Log-based	SKI [23]	●	●	○	○	○	◑	◑	◑	●	◑	●	○	○	●	◑
		CT [5]	●	○	●	○	●	◑	◑	◑	●	◑	●	○	●	●	◑
		AKI [10]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		ARPKI [113], [114]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		Policert [115]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		DTKI [116]	●	●	○	○	○	◑	◑	◑	●	◑	●	○	○	●	◑
		TripPKI [117]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		Khan et al. [118]	●	●	●	○	●	◑	◑	◑	●	◑	●	●	○	●	◑
		ARCT [119]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		SCM [120]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		Wang et al. [121]	●	●	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
		F-PKI [88]	●	○	●	○	●	◑	◑	◑	●	◑	●	○	○	●	◑
	Blockchain-based	Yakubov et al. [122]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Wang et al. [123]	●	●	○	○	○	●	●	●	●	◑	●	○	●	○	○
		BlockPKI [155]	●	●	○	○	○	●	●	●	●	◑	●	○	●	○	○
		CertChain [125]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		CTB [126]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Conifer [127]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Certledger [128]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		PADVA [129]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Xiong et al. [130]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		BB-PKI [131]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○

while leaving the "prevents malicious certificate issuance" feature unfulfilled.

They fully conceal browsing history and do not need an additional connection due to local PK validation support. But they need extra confirmation on the server side without any special need for modifications on the client and CA side. The server-based pinning proposals partially satisfy the "server side changes" feature while avoiding changes on the client

TABLE VIII
(Continued.) A COMPARISON OF LEADING CONVENTIONAL PKIX, REVOCATION PROPOSALS, AND THEIR MODERN IMPLEMENTATION ON BLOCKCHAIN AND DISTRIBUTED LEDGERS BASED ON METRICS DEFINED IN SECTION VI. NOTE: ●=SUPPORTS THE FEATURE; ◑= PARTIALLY SUPPORTS THE FEATURE; ○= DOES NOT SUPPORT THE FEATURE

		Hwang et al. [133]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		ENS [134]	●	○	●	○	●	●	●	●	●	◑	●	○	●	○	○
		Authledger [135]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Lewison et al. [136]	●	○	●	○	●	●	●	●	●	◑	●	○	●	○	○
		LightLedger [137]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		DNSBA [138]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Korgan [139]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
		Meta-PKI [140]	●	○	○	○	○	●	●	●	●	◑	●	○	●	○	○
Revocation proposals	Pull-based	CRL [165]	○	○	●	○	●	○	○	○	○	○	●	○	●	●	●
		Blacklist CRLs [170]	○	○	●	○	●	○	○	○	○	○	●	○	○	◑	●
		CRS [171]	○	○	●	○	●	○	○	◑	○	○	●	○	●	●	●
		Aiello et al. [175]	○	○	●	○	●	○	○	◑	○	○	●	○	●	●	●
		Nissim et al. [178]	○	○	●	○	●	◑	○	○	○	○	●	○	●	●	●
		Kocher [176]	○	○	●	○	●	◑	○	○	○	○	●	○	●	●	●
		OCSP [180]	○	○	○	○	○	◑	○	○	○	○	●	○	●	◑	●
		WCR [182]	○	○	●	○	○	○	○	●	○	○	●	○	●	●	●
		Boneh et al. [188]	●	●	○	○	○	○	●	●	○	○	●	○	○	○	○
	Push-based	CRLSets [191]	○	○	●	○	●	○	○	○	○	○	●	○	●	●	●
		OneCRL [192]	○	○	●	○	●	○	○	○	○	○	●	●	●	●	●
		Larisch et al. [193]	○	○	●	○	●	○	○	○	○	○	●	○	●	●	●
		Let's revok [145	○	○	●	○	●	○	○	○	○	○	●	○	●	●	●
	Network-assisted	Short-lived certificates [194]	○	○	●	○	●	○	●	●	○	○	●	○	○	●	◑
		OCSP stapling [197]	○	○	●	○	●	○	●	○	○	○	●	○	●	●	●
		RevCast [201]	○	○	●	○	●	●	●	●	○	○	●	○	●	●	◑
		OCSP Must-Staple [198]	○	○	●	○	●	○	●	○	○	○	●	○	●	●	●
		Hu et al. [199]	○	○	●	○	●	○	○	○	○	○	●	○	●	●	◑
		RITM [200]	○	○	●	○	●	○	○	○	○	○	●	○	●	●	◑
		SecureGuard [202]	○	○	●	○	●	○	○	○	○	○	●	○	●	●	◑
	Log-based	RT [203]	●	○	●	○	●	◑	◑	◑	●	◑	●	○	●	●	◑
		CIRT [147]	●	○	●	○	●	◑	◑	◑	●	◑	●	○	●	●	◑
		PKISN [204]	●	○	●	○	●	◑	◑	◑	●	◑	●	●	●	●	◑
		Singh et al. [205]	●	○	●	○	●	◑	◑	◑	●	◑	●	○	●	●	◑
	Blockchain-based	BlockVoke [206]	●	○	●	○	●	●	●	●	●	●	●	●	●	○	○
		Process [207]	●	○	●	○	●	●	●	●	●	●	●	○	●	○	○

and CA side. Regarding trust, server-based pinning proposals rely on TOFU without offering limited and decentralized trust benefits. They also do not support the revocation of CA. Fortunately, they do not induce additional workload on the CA side, maintaining a high scalability level and no changes to the underlying Internet. On the other hand, DNS-based pinning and multipath probing proposals protect clients on their first visit to a domain by extending the trust model and offering the "no TOFU" benefit. Pinning with multipath probing offers the "detects malicious certificate issuance" benefit; nonetheless, it introduces an additional connection and causes browsing history leakage.

4) Log-Based PKIX Proposals: This class fully supports the "detects malicious certificate issuance" and "prevents malicious certificate issuance" features, except for CT and F-PKI, which leave the "prevents malicious certificate issuance" feature unfulfilled. The proposals that prevent malicious issuance rely on multi-signed certificates (e.g., ARPKI, AKI) or cross-signed certificates (e.g., SKI, DTKI).

In a similar manner, this class fully hides browsing history. It does not require an extra connection, except for SKI and DTKI, where users must connect to mirror servers and mapping log maintainers before every connection. Therefore, mirror servers and log maintainers know the browsing history of users. Log-based proposals require moderate changes on the client, server, and CA sides, as clients, servers, and CAs need to slightly change their business to take full advantage of their security benefits. For example, clients must actively or passively validate some additional credentials stapled with CA-signed certificates. Log-based proposals improve the conventional CA trust model by limiting the absolute trust placed in CAs and distributing it among several parties. Therefore, they fulfill the "limited trust" metric while partially fulfilling the "decentralized trust" feature. They also offer the "no TOFU" benefit. In addition to issuing certificates, CAs also need to participate in the monitoring process for malicious activity; therefore, log-based proposals leave the "no extra burden on CA" metric unfulfilled except for CT. Log-based proposals are designed to scale to current and future Internet traffic with minor changes to the underlying Internet. Thus, they fulfill the "scalable" feature while partially fulfilling the "no Internet-side changes" feature.

5) Blockchain-Based PKIX Proposals: This class fully supports the "detects malicious certificate issuance" feature while leaving the "prevents malicious certificate issuance" feature unfulfilled except for BlockPKI [155] and the scheme in [123], which fulfill the last feature. The proposals that prevent malicious issuance rely on multi-CA signed certificates (e.g., BlockPKI) or cross-signed certificates (e.g., Wang et al. [123]). This class does not hide browsing history, and clients require an extra connection, as light nodes need to contact full nodes while validating a certificate. Thus, the full nodes can learn the browsing history of the light nodes. Blockchain-based proposals require changes on the client, server and CA side as they need to completely change their business models to take full advantage of the security benefits offered by the blockchain platform. Blockchain-based proposals improve the conventional CA trust model by limiting the absolute trust placed in CAs and distributing it among several parties. Thus, they offer "limited trust" and "no TOFU" benefits. However, they partially fulfill the "decentralized trust" property since CAs remain the certificate issuers in this class of primitives. Similarly, blockchain-based PKIX proposals do not impose an extra burden on CA, as CA is not required to actively participate in the monitoring process. Unfortunately, blockchain-based PKIX proposals leave the "scalable" feature unfulfilled due to the well-known scalability issues of current blockchain technologies, such as low transaction throughput and rapid growth of ledger size. Similarly, deploying a blockchain-based PKIX proposal needs substantial changes to online communication. Therefore, the "no Internet-side changes" feature remains unfulfilled.

6) Pull-Based Revocation Proposals: This class leaves the "detects malicious certificate issuance" and "prevents malicious certificate issuance" features unfulfilled, except for the scheme in [187], as this class does not offer any mechanism to detect and prevent malicious PKs. This class preserves the privacy of clients and generally does not require additional connection, except for OCSP and the scheme in [187]. In OCSP, clients need to query online OCSP servers to validate revocation status, while in the scheme in [187], CA countersigns each server message. Thus, CA can learn the browsing history of clients and servers as well as need an extra connection for each signed message. This class of primitives generally does not need any changes on the client, server, and CA side, as they do not need to alter their business model to use security services, with few exceptions, such as the scheme in [187] and WCR [181]. In work [187], the CA and the server must sign each message cooperatively. The negative side of this class of primitives is that this class does not enhance the trust model by leaving the "limited trust" and "decentralized trust" features unmet. On the positive side, this class satisfies the "no TOFU" property. Similarly, this class fulfills the "no extra burden on CA", "scalable", and "no Internet-side changes" features, as CAs periodically issue revocation information that scales well without changing the underlying Internet architecture. As can be observed from Table VIII, the scheme in [187] deviates in terms of these characteristics due to the active participation of CA in the signing/decrypting of messages, which greatly affects the scalability of this scheme.

7) Push-Based Revocation Proposals: This class is discussed in a less verbose way. This class offers the same benefits as pull-based proposals but with an extra edge regarding the dissemination method. The push-based dissemination method can promptly ensure fresh revocation data, in general, compared to the pull-based method. Moreover, OneCRL allows the revocation of intermediate CAs to prevent compromised intermediate CAs from abusing system security and secure communication.

8) Network-Assisted Revocation Proposals: As it can be observed from Table VIII, this class of proposals offers the same benefits as push-based schemes for the first-sex evaluation metrics. This class shows slightly different traits with respect to changes on the client, server, and CA side. For example, short-lived certificates need changes on the server and CA side, while RevCast needs changes on three sides. On the other hand, OCSP stapling and OCSP Must-Staple need changes only on the server side. This class is similar to pushed-based proposals regarding the "limited trust", "decentralized trust", "no TOFU", and "CA revocation" metrics. The network-assisted proposals show mixed behaviors while examining the "no burden on CA, scalable", and "no Internet-side changes" properties. For example, short-lived certificates exert extra pressure on CAs by forcing them to issue several orders of magnitude greater numbers of certificates than traditional issuance. Similarly, RevCast, Hu et al. [198], RITM, and SecureGuard mandate the installation of additional infrastructure components to provide revocation information.

9) Log-Based Revocation Proposals: Log-based revocation proposals are presented to complement the revocation left open by Google CT. Thus, this class offers almost the same features as Google CT. Fortunately, some proposals fixed some additional issues of certificate revocation. For instance, the scheme in [204] reduced the proof size of Google CT, while PKISN also introduced the revocation of intermediate CAs without causing collateral damage through timestamping.

10) Blockchain-Based Revocation Proposals: It can be observed from Table VIII that this class of proposals offers the same benefits as log-based schemes for the first five parameters. They change the entire business model of the client, server, and CA. This class improves the trust deficit of the CA trust model by limiting and distributing trust among different parties, as well as by not relying on the TOFU model. This class also lacks CA revocation features and does not exert an extra burden on CAs. Unfortunately, the underlying Internet architecture needs substantial changes and has scalability issues, such as blockchain-based PKIX proposals.

C. Defense Comparison

In this section, the defense offered by the PKIX proposals, the revocation proposals, and their modern implementations are investigated against cyber-attacks discussed in Section IV-C in a less verbose manner.

1) CA-Centered Proposals: Schemes in [79], [81], [83], [84] restrict the certificate issuance scope of CAs for domains in this class. Thus, they can only prevent the first six types of attack if a restricted CA is taken down. Syta et al. [86] can

thwart all attacks except the last two, as a group of witnesses validates a document and signs it after successful validation. Additionally, the group signing key is aggregated, distributed, and maintained on the client side; therefore, the truststore is protected, and a malicious witness (CA) does not affect the validation process. On the other hand, Let's Encrypt leaves the "CA compromise" and "compelled certificate" attacks unfilled while preventing the "BGP route hijacking, deprovisioned cloud instances, expired domain", and "DNS record spoofing" attacks by incorporating different defense mechanisms proposed in the works [11], [12], [85]. It can be observed from Table IX that CA-centered proposals leave "truststore attack", "malicious CA insertion", "revocation information blocking", and "Efail" attacks unfulfilled with only the exception of the scope-based scheme, which can prevent it through scope restriction.

2) Client-Centered PKIX Proposals: This class of proposals is defenseless against strong attackers, except for the Certificate Patrol proposal. The Certificate Patrol proposal has the same level of defense as the scope-based proposals.

3) Domain-Centered PKIX Proposals: This class can defend against the first seven cyber-attacks similar to the scope-based proposals, except for DANE, which can completely prevent DNS record spoofing attacks.

4) Log-Based PKIX Proposals: This class fully defends against the "CA compromise, compelled certificate, BGP route hijacking, deprovisioned cloud instance, expired domain, discontinued service, DNS record spoofing", and "revocation information blocking" attacks through checks-and-balances, publicly auditable ledger, and incorporating enhanced certificate revocation mechanisms while leaving the rest unfulfilled. On the other hand, Google CT can only detect the first seven attacks through successful detection of malicious issuance, but cannot prevent them, while the rest are unfulfilled along with the revocation information blocking attacks.

5) Blockchain-Based PKIX Proposals: As can be observed from Table IX, this class of proposals offers almost the same level of defense as that of Google CT, with additional protection against split-world attacks. However, few proposals, such as the proposals in [123], [131], [139], [140], [155], offer a stronger defense than those of advanced log-based proposals (e.g., AKI), while PADVA and Xiong et al. [130] offer a low-level defense.

6) Pull-Based/Push-Based Revocation Proposals: Table IX shows that pull-based/push-based schemes are defenseless against all new types of attacks. They only offer the basic functionality of key revocation without offering extra security.

7) Network-Assisted Revocation Proposals: It can be observed from Table VIII that this class of proposals offers the same benefits as the push-based/pushed-based schemes with slightly different traits regarding revocation information blocking attacks. Similarly, the SecureGuard proposal offers a higher level of security properties over the other proposals in the same class.

8) Log-Based Revocation Proposals: Log-based revocation proposals are presented to complement the revocation left open by Google CT. Thus, this class offers almost the same level of defense as Google CT. Fortunately, some proposals advanced their defense through design and checks-and-balances.

9) Blockchain-Based Revocation Proposals: It can be observed from Table IX that this class has the same defense against that of the log-based proposals, with a slight improvement in defense against the split-world attack.

D. Revocation Evaluation Metrics

Revocation and validation of the revocation status is an overcomplicated process. The following metrics are defined to assess the revocation proposals thoroughly, and a comparison among them is provided in the next section.

Storage cost: It represents the storage cost at the end-devices, and the revocation should require minimal storage.

Revocation check latency: It shows the time taken by a client application to validate the revocation status of an encountered certificate.

Extra connection: It shows whether the client software needs to query an online server to validate the revocation status of the encountered certificate.

Timeliness: It shows the time taken to update the revocation information.

Failure model: This metric shows the mode adopted by a revocation scheme in case of missing/incorrect revocation information.

Bandwidth cost: This metric shows the cost consumed by a revocation scheme per connection while sending revocation information.

PKIX Compatible: This metric measures whether the scheme is compatible with the X.509 standard or not.

E. Comparison of Revocation Proposals

It can be seen in Table X that CRLs have a higher storage cost in pull-based schemes, while CRLite is leading among push-based schemes. RevCast induces the highest storage cost among network-assisted schemes. Log-based schemes require storage in kilobytes, whereas blockchain-based schemes do not shed light on storage costs.

Revocation checking delay is another important factor that influences handshakes during TLS communication. Most schemes induce a delay in milliseconds to the validation process. The highest latency observed was induced by the scheme in [198], which is up to 2 seconds in the worst-case scenario. As can be observed from Table X, some schemes also need to establish a connection to online servers to validate revocation status, which induces a Round Trip Time (RTT) delay.

Regarding timeliness, blockchain-based schemes ensure the freshness of revocation information, while network-assisted, pull- and push-based schemes take longer to update revocation information. In a similar way, log-based proposals also offer a fair level of timeliness in hours.

Blockchain- and log-based revocation proposals result in a hard failure in case of missing revocation information, which can guard clients against MitM. On the other hand, network-assisted, pull- and push-based vary in terms of response to invalid/missing revocation information.

TABLE IX
COMPARING DEFENSE OF LEADING CONVENTIONAL PKIX, REVOCATION PROPOSALS, AND THEIR MODERN IMPLEMENTATION ON BLOCKCHAIN AND DISTRIBUTED LEDGERS AGAINST ATTACKS DISCUSSED IN SECTION IV-C. NOTE: ●=SUPPORTS THE FEATURE; ◑= PARTIALLY SUPPORTS THE FEATURE; ○=DOES NOT SUPPORT THE FEATURE

Class	Subclass	Proposals	CA compromise	Compelled certificate attack	BGP route hijacking	Deprovisioned Cloud Instance attack	Expired domain attack	Discontinued service attack	DNS record spoofing attack	Split-world attack	Truststore attack	Malicious CA insertion attack	Revocation information blocking attack	Efail attack	DNS cache poisoning attack	Vantage downgrade attack	Zombie certificate attack
PXIX proposals	CA-centered	Kasten et al. [79]	◑	◑	◑	◑	◑	◑	◑	○	○	●	○	○	◑	○	○
		Braun et al. [81]	◑	◑	◑	◑	◑	◑	◑	○	○	●	○	○	◑	○	○
		Braun et al. [83]	◑	◑	◑	◑	◑	◑	◑	○	○	●	○	○	◑	○	○
		Classen et al. [84]	◑	◑	◑	◑	◑	◑	◑	○	○	●	○	○	◑	○	○
		Syta et al. [86]	●	●	●	●	●	●	●	●	●	●	○	○	●	●	○
		Wang et al. [89]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		Let's Encrypt	○	○	●	●	●	●	●	○	○	○	○	○	○	○	○
		Jayaraman et al. [93]	●	●	●	●	●	●	●	○	○	○	○	○	●	●	○
		Zhao et al. [132]	◑	◑	◑	◑	◑	◑	◑	●	○	○	○	○	◑	◑	○
	Client-centered	Wendlandt et al. [16]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		DoubleCheck [94]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		Marlinspike [15]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		CertLock [74]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		Policy Engine [96]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		Certificate Patrol [100]	◑	◑	◑	◑	◑	◑	◑	○	○	○	○	○	○	○	○
	Domain-centered	PKP [103], [104]	◑	◑	◑	◑	◑	◑	◑	○	○	○	○	○	◑	○	○
		TACK [108]	●	●	●	●	●	●	●	○	○	○	○	○	●	○	○
		DANE [17]–[19]	◑	◑	◑	◑	◑	◑	●	○	○	○	○	○	◑	○	○
		Elaphurus [109]	◑	◑	◑	◑	◑	◑	◑	○	○	○	○	○	◑	○	○
	Log-based	SKI [23]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	○
		CT [5]	◑	◑	◑	◑	◑	◑	◑	○	○	○	○	○	○	○	○
		AKI [10]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		ARPKI [113], [114]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		Policert [115]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		DTKI [116]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		TripPKI [117]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		Khan et al. [118]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		ARCT [119]	●	●	●	●	●	●	●	○	○	○	●	○	○	○	○
		SCM [120]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	○
		Wang et al. [121]	●	●	●	●	●	●	●	○	○	○	●	○	●	●	●
		F-PKI [88]	●	●	●	●	●	●	●	○	○	○	●	○	●	○	●
	Blockchain-based	Yakubov et al. [122]	◑	◑	◑	◑	◑	◑	◑	●	●	●	●	○	○	○	○
		Wang et al. [123]	●	●	●	●	●	●	●	●	◑	◑	●	○	●	●	●
		BlockPKI [155]	●	●	●	●	●	●	●	●	◑	◑	●	○	●	●	●
		Tewari et al. [124]	◑	◑	◑	◑	◑	◑	◑	●	○	○	○	○	○	○	○
		CertChain [125]	◑	◑	◑	◑	◑	◑	◑	●	○	○	●	○	◑	◑	●
		CTB [126]	◑	◑	◑	◑	◑	◑	◑	●	●	●	●	○	○	○	○
		Conifer [127]	◑	◑	◑	◑	◑	◑	◑	●	○	○	○	○	○	○	●
		Certledger [128]	◑	◑	◑	◑	◑	◑	◑	●	●	●	●	○	◑	◑	○

Table X also investigates the bandwidth cost of each scheme per connection. From the table, it can be derived that OneCRL and Let's Revoke consume higher bandwidth per connection. Finally, all methods are compatible with PKIX except Let's Revoke, CRS, Aiello et al. [174], Boneh et al. [187] and blockchain-based schemes. Any scheme that needs changes

TABLE IX
(Continued.) COMPARING DEFENSE OF LEADING CONVENTIONAL PKIX, REVOCATION PROPOSALS, AND THEIR MODERN IMPLEMENTATION ON BLOCKCHAIN AND DISTRIBUTED LEDGERS AGAINST ATTACKS DISCUSSED IN SECTION IV-C. NOTE: ●=SUPPORTS THE FEATURE; ◑= PARTIALLY SUPPORTS THE FEATURE; ○=DOES NOT SUPPORT THE FEATURE

		PADVA [129]	○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ○ ○ ○ ○
		Xiong et al. [130]	○ ○ ○ ○ ○ ○ ● ● ○ ○ ○ ○ ○ ○ ○
		BB-PKI [131]	● ● ● ● ● ● ● ● ○ ○ ○ ○ ● ● ●
		Hwang et al. [133]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ● ○ ○ ○ ○ ◑ ◑ ○
		Authledger [135]	◑ ◑ ◑ ◑ ◑ ◑ ● ● ○ ○ ○ ○ ○ ○ ○
		Lewison et al. [136]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ● ○ ○ ○ ○ ○ ○ ○
		LightLedger [137]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ● ○ ○ ● ○ ◑ ◑ ◑
		DNSBA [138]	◑ ◑ ◑ ◑ ◑ ◑ ● ● ○ ● ◑ ○ ○ ○ ○
		Korgan [139]	● ● ● ● ● ● ● ● ○ ○ ● ○ ● ● ●
		Meta-PKI [140]	● ● ● ● ● ● ● ● ○ ○ ● ○ ● ● ●
Revocation proposals	Pull-based	CRL [165]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		Blacklist CRLs [170]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		CRS [171]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		Aiello et al. [175]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		Nissim et al. [178]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ○
		Kocher [176]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ○
		OCSP [180]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		WCR [182]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		Boneh et al. [188]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ○
	Push-based	CRLSets [191]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		OneCRL [192]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		Larisch et al. [193]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
		Let's revok [145]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
	Network-assisted	Short-lived certificates [194]	○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ○ ○ ○ ●
		OCSP stapling [197]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ●
		RevCast [201]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ●
		OCSP Must-Staple [198]	○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ○ ○ ○ ●
		Hu et al. [199]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ●
		RITM [200]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ●
		SecureGuard [202]	○ ○ ○ ● ○ ● ● ● ● ○ ○ ● ○ ○ ●
	Log-based	RT [203]	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○ ○ ○
		CIRT [147]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ○ ○ ○ ● ○ ○ ○ ●
		PKISN [204]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ○ ○ ○ ● ○ ○ ○ ●
		Singh et al. [205]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ○ ○ ○ ● ○ ○ ○ ●
	Blockchain-based	BlockVoke [206]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ● ○ ○ ● ○ ○ ○ ●
		Process [207]	◑ ◑ ◑ ◑ ◑ ◑ ◑ ● ○ ○ ● ○ ○ ○ ●

to the underlying infrastructure is less likely to be adopted by the Web community.

VII. LESSONS LEARNED, RESEARCH GAP AND FUTURE PERSPECTIVES

In this section, key lessons learned from our comprehensive survey on the PKIX proposals are presented first. Next, the research gaps and directions for future work are briefly discussed.

A. Lessons Learned

Despite efforts to secure the fragile CA ecosystem through the use of different security mechanisms, CAs are still not fault proof. The enhancement of their security mechanism and the adoption rate to deal with stronger attackers are relatively slow. In the last two decades, attackers and researchers have taken down various famous CAs by bypassing their security mechanisms. However, none of the CAs developed a robust mechanism to stop attacks, and their security relied mainly on third-party-developed security mechanisms. For example, Bog et al. took down CAs and presented countermeasures to defend against such attacks. Only a few CAs incorporated the countermeasures in their original form without advancing and enhancing them. Consequently, a modified version of the same attack was carried out against different CAs, including those that incorporate the countermeasures presented in [11]. CAs should be agile in adopting newer and stronger security to defend their own infrastructure against powerful adversaries (e.g., state-level adversaries) and keep their reputation of providing robust security services. Furthermore, most small CAs do not comply with security requirements and standards [209].

Various state-level actors have been noticed to be directly involved in monitoring and intercepting Internet traffic to analyze the data of Internet clients. This kind of state-level actor raises concerns about the conventional security protocol, which relies on a weaker attacker model by assuming that the trusted third-party servers are secure and trusted. Unfortunately, this assumption does not seem to hold against stronger state-level adversaries. The NASA report revealed that

TABLE X
A COMPARISON OF LEADING REVOCATION PROPOSALS BASED ON METRICS DEFINED IN SECTION VI-D

Class	Subclass	Proposals	Storage cost	Revocation check delay	Extra connection	Timeliness	Failure model	Bandwidth cost	PKIX compatible
Revocation proposals	Pull-based	CRL [165]	173kb-76MB	5ms-200ms	No	4-7 days	soft	q.173kb	Yes
		CRS [171]	0	N/A	Yes	1 day	hard	100 bits	No
		Aiello et al. [175]	0	N/A	Yes	1 day	hard	100 bits	No
		Nissim et al. [178]	100 bytes.nlog	N/A	No	day	hard	960bytes-1.2kb	Yes
		Kocher [176]	100 bytes.nlog	N/A	No	day	hard	960bytes-1.2kb	Yes
		OCSP [181]	0	5ms-250ms	Yes	1-4 days	soft	1.3kb	Yes
		Boneh et al. [188]	0	8ms-42ms	No	0	hard	N/A	No
	Push-based	CRLSets [191]	250kb	5ms-20ms	No	1 day	soft	250kb	Yes
		OneCRL [192]	600kb	5ms-20ms	No	1 day	hard	600kb	Yes
		Larisch et al. [193]	3.1 MB	6-10ms	No	1 day	hard	408kb	Yes
		Let's Revok [145]	2.2MB	10ms	No	1 day	hard	114kb	No
	Network-assisted	Short-lived certificates [194]	0	0	No	1-3 days	hard	0	Yes
		OCSP Stapling [197]	0	N/A	No	1-4 days	soft	500-1000 bytes	Yes
		RevCast [201]	173kb	0	No	hours	soft	173kb	No
		OCSP Must-Staple [198]	0	N/A	No	1-4 days	hard	500-1000 bytes	Yes
		Hu et al. [199]	0	2ms-2s	Yes	1-4 days	soft	300bytes	Yes
		RITM [200]	0	250μs	No	hours	hard	960bytes-1.2kb	Yes
		SecureGuard [202]	0	0.1ms-1ms	No	hours	soft	1kb	Yes
	Log-based	RT [203]	100bytes-5kb	N/A	No	hours	soft	960bytes-1.2kb	Yes
		CIRT [147]	100bytes-5kb	N/A	No	hours	soft	960bytes-1.2kb	Yes
		PKISN [204]	100bytes-5kb	N/A	No	hours	soft	960bytes-1.2kb	Yes
		Singh et al. [205]	100bytes-5kb	3.06ms	No	hours	soft	32bytes	Yes
	Blockchain-based	BlockVoke [206]	N/A	N/A	Yes	minutes	hard	N/A	No
		Process [207]	N/A	3.75ms	Yes	minutes	hard	N/A	No

the U.S. government is monitoring user connections, while the Kazakhstan government recently intercepted the connections of users by forcing them to install and trust the root CA certificate of the Kazakistan government. In addition, it was found that 49 out of 61 of root CA certificates of different governments (e.g.,the governments of India, Brazil, France, Saudi Arabia, South Africa, and Uruguay) are installed in the truststore of Microsoft [210]. These root CA certificates and incidents question the fundamental assumption behind the current working principle of PKIX and highlight the need to reevaluate the security assumptions. Therefore, researchers, experts, and academicians must move to the next level of the attacker model, which can hold against state-level adversaries.

Although the issuance of DV-certificates and DV-based CAs dominate the certificate ecosystem, there are different attack vectors against the issuance process of DV-certificate. The issuance of the DV-certificate involves minimal level validation, as it only validates that the entity applying for the certificate has control over the domain. Additionally, the method used to issue DV-certificates and identity verification is insecure and vulnerable to MitM. It was learned that using one validation method is not strong enough to block attacks on DV-certificate issuance. A study revealed that around 1 million malicious certificates exist in the TLS ecosystem for the top 10k Alex websites. Relying on more than one identity would introduce a delay to the certificate issuance process; however, it would greatly reduce the attack surface on the issuance process of the DV-certificate.

Certificate transparency is an elegant mechanism for proactively detecting the issuance of malicious certificates. It does not prevent the issuance of malicious certificates and can only detect them once logged into the CT logs. CAs can monitor CT logs to identify malicious entries; however, it cannot be applied to detect attacks on the identity validation process. Thus, it has a window of vulnerabilities and leaves space for attackers to conduct cyber-attacks against CAs and their clients. Moreover, the browser support for CT is surprisingly poor [211]. CAs should rely on different intrusion detection methods to proactively identify attacks during the validation process and prevent malicious issuance.

Integration of different technological solutions comes up with new vulnerabilities. Hosting secure services on cloud computing gave rise to dangling DNS record loopholes, while S/MIME gave rise to the decryption of encrypted email attacks. The loopholes remained undetected, and clients relied on fragile and insecure ecosystems for years. We learned that faulty design or faulty implementation of the security protocol can sabotage the security of the entire system.

It was learned that several attacks, which were theoretical a few years ago, turned into practice. For example, rogue CA (e.g., DigiNotar) and CA compromise through a software bug (e.g., Heartbleed bug discovered in 2014) attacks were theoretical a few years ago until attacks happened. Currently, a versioning attack on CT is theoretical, where an attacker compromises CA and CT logs to show different versions of their logs to different sets of users. These attacks remain theoretical in the face of weaker attackers and are turned into practice if the attackers enhance their potential.

It was discovered that the adoption of any security solution is determined by various factors on the Web, where multiple factors such as standards organization (e.g., IETF), regulatory requirements from regulatory bodies (e.g., government), user demand, and security incidents can mandate the adoption of a security mechanism. Unfortunately, giant stakeholders play an important role in adopting a security mechanism. Several elegant mechanisms and solutions (e.g., DANE) are proposed to enhance the security of the Web. However, these solutions lack widespread adoption due to a lack of support from major browser and CA vendors.

The Web is a very diverse and open community that is open to diverse attackers with different attack capabilities. The diverse group of attackers, which can include cyber criminals, state-sponsored actors, insider attackers, and advanced persistent attackers, poses unique challenges and needs different levels of defense strategies. These attackers can launch different attacks simultaneously against Web users. Implementing and patching a single security solution cannot ensure security against the diverse nature of attackers. Clients need to use various robust and layered security protocols to defend themselves against these diverse attackers.

B. Research Gap and Future Perspectives

As discussed previously, PKIX is a widely deployed PKI, and different cyber-attacks are successfully conducted against PKIX clients. In this survey article, some issues are emphasized in the PKIX and revocation process, and the following strategies and directions are presented to improve the overall security and performance of the PKIX and revocation process in the future.

1) Immature certificate revocation process: It was found that CRL and OCSP dominate the Web for checking the certificate revocation process. Unfortunately, the certificate revocation method only checks that the certificate has not yet been revoked; however, it does not include any mechanism to ensure defense against malicious certificates. The current revocation process is overcomplicated, and none of them actually solves the problem. Furthermore, the browser support for the revocation process and the detection of revoked certificates is pretty low [111], [211]. For example, a study revealed that browsers can successfully detect approximately 35- 40% of revoked certificates [212]. Incorporating simplicity, security, and defense into revocation mechanisms can greatly improve the security of the Internet.

2) Security and privacy requirements for PKI in Web 3.0 scenarios: No work was found that focused on the detailed security, privacy requirements, and design of the PKI, which can meet the communication requirements of Web 3.0. Most of the proposals tried to instantiate the PKI design on blockchain technology and fix the problems of the PKIX design. Additionally, no work was found that came up with a new identity validation method focusing on Web 3.0 communication, since the main objective of this standard is to eliminate centralized authorization and validation systems in the future.

3) Client identity validation: Client authentication is still an open problem on the Internet. The work in [124] only designed client-based PKIX to issue free certificates for end-users. However, this work does not specify any mechanism for end-user identity validation. PKIX still lacks an identity validation mechanism for end-users using services and still relies on mechanisms such as password and two-factor authentication. The IETF Let's Encrypt project that aims to encrypt Internet traffic would be incomplete without providing a free and user-friendly certificate issuance process for end-users.

4) CA revocation: Revoking a CA certificate invalidates all domains' certificates signed by that particular CA. Some proposals presented mechanisms to invalidate an intermediate CA certificate without causing collateral damage. Unfortunately, PKIX has no practical mechanism to revoke the CA certificate without causing collateral damage. Moreover, trutstore must be updated on all client software if a root CA certificate is revoked to remove the rogue CA certificate. Updating truststores on client software requires updating the operating system, browser, and other applications, since truststores are maintained by them, which usually takes a few days to weeks. It is still an open research problem to design methods that safely revoke CA certificates without collateral damage and in a timely manner.

5) Elimination of certificate chaining: It was identified during our article on vehicular PKI [67] that certificate chain verification involves at least two signature verification operations to authenticate a message. This induces an extra delay caused by the hierarchical CA model adopted by PKIX to reduce the size of the truststore at the cost of verification cost. The extra delay can be introduced by certificate chain depth, leaf certificate, intermediate CA certificate revocation checking, and occasionally by certificate path discovery if the server does not include an intermediate CA certificate. The certificate chain problem significantly affects various application deployments designed for real-time and IoT services, as these real-time and IoT services need time-efficient authentication processes and resource-constrained devices, respectively. In addition, certificate chaining can cause security breaches. For example, browsers can accept certificates that violate constraints (e.g., NameContraints and PathlenContraint) and thus accept invalid certificates [212]. Redesigning truststore and authentication protocols to eliminate certificate chain validation without compromising security would enable real-time and IoT services to use PKIX for authentication.

6) It was observed that the secure and rigorous identity validation remains an open problem of PKIX design. There exist various attack vectors against the validation methods that attackers exploit to issue malicious certificates. Additionally, the validation of an identity that preserves privacy is also a challenging task. Only the work of Wang et al. [89] was found that aims to offer validation and certification of private identities. However, the method is only applicable to email-based identity validation scenarios, which is insecure in itself, and an attacker can easily obtain malicious certificates on behalf of organizational employees.

VIII. Conclusion

The security of online communication is bootstrapped from PKI, which provides the foundation for PKC realization. PKIX is the most widely used PKI to secure Web and online communication. In this survey, we discussed the PKIX architecture, Web evolution, and revocation proposals that complement the revocation process. This survey focused on classifying PKIX proposals, revocation proposals, and proposals based on blockchain and transparency log (ledger) technologies. The article compares the PKIX proposals, the certificate revocation proposals, and the recent blockchain- and log-based proposals using 15 evaluation metrics in a uniform way. In addition, the defense of leading proposals is investigated against recent attack vectors. The systematic comparison and defense investigation helped to identify the research gap despite existing proposals. Finally, we presented the lessons learned and highlighted certain security loopholes and performance issues of the PKIX schemes.

Acknowledgment

The authors thank the anonymous reviewers for their feedback and comments.

REFERENCES

[1] "Digital 2022: April global statshot report." Accessed: Jun. 30, 2022. [Online]. Available: https://datareportal.com/reports/digital-2022-april-global-statshot

[2] "Vulnerability and threat trends report 2022." Accessed: Jun. 30, 2022. [Online]. Available: https://www.skyboxsecurity.com/resources/report/vulnerability-threat-trends-report-2022/

[3] G. Greenwald et al., "NSA collecting phone records of millions of Verizon customers daily," *Guardian*, vol. 6, no. 6, p. 13, 2013.

[4] R. S. Raman, L. Evdokimov, E. Wurstrow, J. A. Halderman, and R. Ensafi, "Investigating large scale HTTPs interception in Kazakhstan," in *Proc. ACM Internet Meas. Conf. (IMC)*, 2020, pp. 125–132. [Online]. Available: https://doi.org/10.1145/3419394.3423665

[5] B. Laurie, "Certificate transparency," *Commun. ACM*, vol. 57, no. 10, pp. 40–46, 2014. [Online]. Available: https://doi.org/10.1145/2659897

[6] C. Welch. "Google encrypts gmail between data centers to keep the NSA out of your inbox." Mar. 2014. [Online]. Available: https://www.theverge.com/2014/3/20/5530072/google-encrypts-gmail-between-data-centers-to-keep-out-nsa

[7] C. Farivar, "Apple expands data encryption under iOS 8, making handover to cops moot." Sep. 2014. [Online]. Available: https://arstechnica.com/gadgets/2014/09/apple-expands-data-encryption-under-ios-8-making-handover-to-cops-moot/

[8] S. Farrell and H. Tschofenig, "Pervasive monitoring is an attack," IETF, Rep. RFC 7258, 2014.

[9] J. Aas et al., "Let's encrypt: An automated certificate authority to encrypt the entire Web," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2019, pp. 2473–2487. [Online]. Available: https://doi.org/10.1145/3319535.3363192

[10] T. H.-J. Kim, L.-S. Huang, A. Perrig, C. Jackson, and V. Gligor, "Accountable key infrastructure (AKI): A proposal for a public-key validation infrastructure," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, May 2013, pp. 679–690. [Online]. Available: https://doi.org/10.1145/2488388.2488448

[11] H. Birge-Lee, Y. Sun, A. Edmundson, J. Rexford, and P. Mittal, "Bamboozling certificate authorities with BGP," in *Proc. 27th USENIX Conf. Security Symp. (SEC)*, 2018, pp. 833–849.

[12] K. Borgolte, T. Fiebig, S. Hao, C. Kruegel, and G. Vigna, "Cloud strife: Mitigating the security risks of domain-validated certificates," in *Proc. Appl. Netw. Res. Workshop (ANRW)*, 2018, p. 4. [Online]. Available: https://doi.org/10.1145/3232755.3232859

[13] R. A. Rueppel and B. Wildhaber, "Public key infrastructure—Survey and issues," in *Trust Center*. Cham, Switzerland: Springer, 1995, pp. 197–212.

[14] A. C. Grant. "Search for trust: An analysis and comparison of ca system alternatives and enhancements." 2012. [Online]. Available: https://digitalcommons.dartmouth.edu/senior_theses/78/

[15] M. Marlinspike, "SSL and the future of authenticity," in *Proc. Black Hat USA*, 2011, p. 6.

[16] D. Wendlandt, D. G. Andersen, and A. Perrig, "Perspectives: Improving SSH-style host authentication with multi-path probing," in *Proc. USENIX Annu. Tech. Conf. (ATC)*, 2008, pp. 321–334.

[17] R. Barnes, "DANE: Taking TLS authentication to the next level using DNSSEC," *IETF J.*, vol. 7, no. 2, p. 360, 2011.

[18] R. Barnes, "Use cases and requirements for DNS-based authentication of named entities (DANE)," Internet Eng. Task Force, RFC 6394, 2011.

[19] P. Hoffman and J. Schlyter, "The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA," IETF, RFC 6698, Aug. 2012.

[20] P. Hallam-Baker, R. Stradling, and B. Laurie, "DNS certification authority authorization (CAA) resource record," Internet Eng. Task Force, RFC 6844, 2013.

[21] K. Engert. "MECAI-mutually endorsing CA infrastructure." 2013. [Online]. Available: https://kuix.de/mecai/mecai-proposal-v2.pdf

[22] G. Toth and T. Vlieg, "Public key pinning for TLS using a trust on first use model," 2013. [Online]. Available: https://rp.os3.nl/2012-2013/p56/report.pdf

[23] P. Eckersley. "Sovereign key cryptography for Internet domains." 2012. [Online]. Available: https://git.eff.org

[24] F. Amin, A. Jahangir, and H. Rasifard, "Analysis of public-key cryptography for wireless sensor networks security," *Int. J. Comput. Inf. Eng.*, vol. 2, no. 5, pp. 1448–1453, 2008.

[25] J. Clark and P. C. Van Oorschot, "SoK: SSL and HTTPs: Revisiting past challenges and evaluating certificate trust model enhancements," in *Proc. IEEE Symp. Security Privacy*, 2013, pp. 511–525.

[26] A. Parsovs, "Practical issues with TLS client certificate authentication," in *Proc. Cryptol. ePrint Archive*, 2013, pp. 1–8.

[27] A. Delignat-Lavaud, M. Abadi, A. Birrell, I. Mironov, T. Wobber, and Y. Xie, "Web PKI: Closing the gap between guidelines and practices," in *Proc. NDSS*, 2014, pp. 1–15.

[28] A. Albarqi et al., "Public key infrastructure: A survey," *J. Inf. Security*, vol. 6, no. 1, p. 31, 2014.

[29] L. S. Huang, A. Rice, E. Ellingsen, and C. Jackson, "Analyzing forged SSL certificates in the wild," in *Proc. IEEE Symp. Security Privacy*, 2014, pp. 83–97.

[30] L. Zhang et al., "Analysis of SSL certificate reissues and revocations in the wake of heartbleed," in *Proc. Conf. Internet Meas. Conf.*, 2014, pp. 489–502.

[31] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar, "TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2016, pp. 1–8.

[32] K. Michael and B. Joseph, "Upgrading HTTPs in mid-air: An empirical study of strict transport security and key pinning," in *Proc. NDSS Symp.*, 2015, pp. 1–8.

[33] J. Hodges, C. Jackson, and A. Barth, "HTTP strict transport security (HSTS)." 2012. [Online]. Available: https://https.cio.gov/hsts/

[34] S. de los Santos, C. Torrano, Y. Rubio, and F. Brezo, "Implementation state of HSTS and HPKP in both browsers and servers," in *Proc. Int. Conf. Cryptol. Netw. Security*, 2016, pp. 192–207.

[35] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, "Towards a complete view of the certificate ecosystem," in *Proc. Internet Meas. Conf.*, 2016, pp. 543–549.

[36] R. Heiland, W. C. Garrison, Y. Qiao, A. J. Lee, and V. Welch. "The Web's PKI: An expository review and certificate validation cost simulation." 2016. [Online]. Available: https://scholarworks.iu.edu/dspace/bitstream/handle/2022/21038/CACR-ASAC-PKI.pdf?sequence=1

[37] K.-A. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 577–601, 1st Quart., 2016.

[38] J. Yu and M. Ryan, "Chapter 7—Evaluating Web PKIS," in *Software Architecture for Big Data and the Cloud*, 2017, pp. 105–126. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128054673000077

[39] J. Gustafsson, G. Overier, M. Arlitt, and N. Carlsson, "A first look at the CT landscape: Certificate transparency logs in practice," in *Proc. Int. Conf. Passive Active Netw. Meas.*, 2017, pp. 87–99.

[40] J. Amann, O. Gasser, Q. Scheitle, L. Brent, G. Carle, and R. Holz, "Mission accomplished? HTTPs security after diginotar," in *Proc. Internet Meas. Conf.*, 2017, pp. 325–340.

[41] S. Weiler and D. Blacka, "Clarifications and implementation notes for DNS security (DNSSEC)," IETF, RFC 6840, Feb. 2013.

[42] C. Nykvist, L. Sjöström, J. Gustafsson, and N. Carlsson, "Server-side adoption of certificate transparency," in *Proc. Int. Conf. Passive Active Netw. Meas.*, 2018, pp. 186–199.

[43] O. Gasser, B. Hof, M. Helm, M. Korczynski, R. Holz, and G. Carle, "In log we trust: Revealing poor security practices with certificate transparency logs and Internet measurements," in *Proc. Int. Conf. Passive Active Network Meas.*, 2018, pp. 173–185.

[44] B. Li et al., "Certificate transparency in the wild: Exploring the reliability of monitors," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2019, pp. 2505–2520.

[45] M. Malik, M. Dutta, and J. Granjal, "A survey of key bootstrapping protocols based on public key cryptography in the Internet of Things," *IEEE Access*, vol. 7, pp. 27443–27464, 2019.

[46] B. Li, D. Chu, J. Lin, Q. Cai, C. Wang, and L. Meng, "The weakest link of certificate transparency: Exploring the TLS/HTTPs configurations of third-party monitors," in *Proc. 18th IEEE Int. Conf. Trust Security Privacy Comput. Commun. 13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, 2019, pp. 216–223.

[47] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 196–248, 1st Quart., 2019.

[48] B. Amann, R. Sommer, M. Vallentin, and S. Hall, "No attack necessary: The surprising dynamics of SSL trust relationships," in *Proc. 29th Annu. Comput. Security Appl. Conf.*, 2013, pp. 179–188.

[49] D. Akhawe, B. Amann, M. Vallentin, and R. Sommer, "Here's my cert, so trust me, maybe? Understanding TLS errors on the Web," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 59–70.

[50] L. Zhu, D. Wessels, A. Mankin, and J. Heidemann, "Measuring DANE TLSA deployment," in *Proc. Int. Workshop Traffic Monitor. Anal.*, 2015, pp. 219–232.

[51] C. Aishwarya et al., "DANE: An inbuilt security extension," in *Proc. IEEE Int. Conf. Green Comput. Internet Things (ICGCIoT)*, 2015, pp. 1571–1576.

[52] H. Lee et al., "A longitudinal and comprehensive study of the DANE ecosystem in email," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, 2020, p. 6.

[53] C. Brunner, F. Knirsch, A. Unterweger, and D. Engel, "A comparison of blockchain-based PKI implementations," in *Proc. ICISSP*, 2020, pp. 333–340.

[54] L. Chuat, A. Abdou, R. Sasse, C. Sprenger, D. Basin, and A. Perrig, "SoK: Delegation and revocation, the missing links in the Web's chain of trust," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, 2020, pp. 624–638.

[55] N. Aldahwan and D. Alghazzawi, "Use of blockchain in public key infrastructure (PKI): A systematic literature review," *Int. J. Comput. Sci. Inf. Security*, vol. 18, no. 6, pp. 106–111, 2020.

[56] G. Schmid, "Thirty years of DNS insecurity: Current issues and perspectives," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2429–2459, 4th Quart., 2021.

[57] O. Albogami, M. Alruqi, K. Almalki, and A. Aljahdali, "Public key infrastructure traditional and modern implementation," *Int. J. Netw. Security*, vol. 23, no. 2, pp. 343–350, 2021.

[58] S. Meiklejohn, J. DeBlasio, D. O'Brien, C. Thompson, K. Yeo, and E. Stark, "SoK: SCT auditing in certificate transparency," 2022, *arxiv.abs/2203.01661*.

[59] D. Maldonado-Ruiz, J. Torres, N. E. Madhoun, and M. Badra, "Current trends in blockchain implementations on the paradigm of public key infrastructure: A survey," *IEEE Access*, vol. 10, pp. 17641–17655, 2022.

[60] X. D. C. de Carnavalet and P. C. van Oorschot, "A survey and analysis of TLS interception mechanisms and motivations," *ACM Comput. Surveys*, vol. 55, no. 13s, pp. 1–40, 2023. [Online]. Available: https://doi.org/10.1145/3580522

[61] M. S. Pour, C. Nader, K. Friday, and E. Bou-Harb, "A comprehensive survey of recent Internet measurement techniques for cyber security," *Comput. Security*, vol. 128, May 2023, Art. no. 103123. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823000330

[62] M. K. Bansal and M. Sethumadhavan, "Survey on domain name system security problems-DNS and blockchain solutions," in *Proc. Int. Conf. Futuristic Trends Netw. Comput. Technol.*, 2019, pp. 634–647.

[63] *Recommendations x.509 and ISO 9594-8*, ITU, Geneva, Switzerland, 1988.

[64] "The directory—Overview of concepts, models and services, Melbourne, Fascicle VIII. 8—Rec," ITU, Geneva, Switzerland, ITU Recommendation 500, 1988.

[65] H. Leibowitz, A. Herzberg, and E. Syta, "Provable security for PKI schemes," in*Proc. IACR Cryptol. ePrint Arch.*, 2019, p. 807.

[66] P. R. Zimmermann, *The Official PGP User's Guide*. Cambridge, MA, USA: MIT Press, 1995.

[67] S. Khan, F. Luo, Z. Zhang, M. A. Rahim, M. Ahmad, and K. Wu, "Survey on issues and recent advances in vehicular public-key infrastructure (VPKI)," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1574–1601, 3rd Quart., 2022.

[68] J. Höglund, S. Lindemer, M. Furuhed, and S. Raza, "PKI4IoT: Towards public key infrastructure for the Internet of Things," *Comput. Security*, vol. 89, Feb. 2020, Art. no. 101658. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404819302019

[69] X. Shi, S. Shi, M. Wang, J. Kaunisto, and C. Qian, "On-device IoT certificate revocation checking with small memory and low latency," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2021, pp. 1118–1134. [Online]. Available: https://doi.org/10.1145/3460120.3484580

[70] J. Höglund, M. Furuhed, and S. Raza, "Lightweight certificate revocation for low-power IoT with end-to-end security," *J. Inf. Security Appl.*, vol. 73, Mar. 2023, Art. no. 103424. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212623000091

[71] A. Garba et al., "LightCERT4IoTs: Blockchain-based lightweight certificates authentication for IoT applications," *IEEE Access*, vol. 11, pp. 28370–28383, 2023.

[72] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, "Automatic certificate management environment (ACME)," IETF, RFC 8555, Mar. 2019.

[73] L. Schwittmann, M. Wander, and T. Weis, "Domain impersonation is feasible: A study of CA domain validation vulnerabilities," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, 2019, pp. 544–559.

[74] C. Soghoian and S. Stamm, "Certified lies: Detecting and defeating government interception attacks against SSL (short paper)," in *Proc. Int. Conf. Financial Cryptography Data Security*, 2011, pp. 250–259.

[75] D. Liu, S. Hao, and H. Wang, "All your DNS records point to us: Understanding the security threats of dangling DNS records," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2016, pp. 1414–1425. [Online]. Available: https://doi.org/10.1145/2976749.2978387

[76] S. Goldberg. "The myetherwallet.com Hijack and why it's risky to hold cryptocurrency in a Webapp." 2018. [Online]. Available: https://medium.com/@goldbe/the-myetherwallet-com-hijack-and-why-its-risky-to-hold-cryptocurrency-in-a-webapp-261131fad278

[77] T. Dai, H. Shulman, and M. Waidner, "Let's downgrade let's encrypt," in *Proc. ACM SIGSAC Conf. Comput. Commub. Security (CCS)*, 2021, pp. 1421–1440. [Online]. Available: https://doi.org/10.1145/3460120.3484815

[78] D. Poddebniak et al., "EFAIL: Breaking S/MIME and OpenPGP email encryption using exfiltration channels," in *Proc. 27th USENIX Security Symp. (USENIX Security)*, Aug. 2018, pp. 549–566. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/poddebniak

[79] J. Kasten, E. Wustrow, and J. A. Halderman, "CAGE: Taming certificate authorities by inferring restricted scopes," in *Proc. Int. Conf. Financ. Cryptography Data Security*, 2013, pp. 329–337.

[80] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining your PS and QS: Detection of widespread weak keys in network devices," in *Proc. 21st USENIX Security Symp. (USENIX Security)*, 2012, pp. 205–220.

[81] J. Braun and G. Rynkowski, "The potential of an individualized set of trusted CAS: Defending against CA failures in the Web PKI," in *Proc. IEEE Int. Conf. Social Comput.*, 2013, pp. 600–605.

[82] J. Braun, F. Volk, J. Buchmann, and M. Mühlhäuser, "Trust views for the Web PKI," in *Proc. Eur. Public Key Infrastruct. Workshop*, 2013, pp. 134–151.

[83] J. Braun, F. Volk, J. Classen, J. Buchmann, and M. Mühlhäuser, "CA trust management for the Web PKI," *J. Comput. Security*, vol. 22, no. 6, pp. 913–959, 2014.

[84] J. Classen, J. Braun, F. Volk, M. Hollick, J. Buchmann, and M. Mühlhäuser, "A distributed reputation system for certification authority trust management," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, vol. 1, 2015, pp. 1349–1356.

[85] M. Brandt, T. Dai, A. Klein, H. Shulman, and M. Waidner, "Domain validation++ for MITM-resilient PKI," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 2060–2076.

[86] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, and B. Ford, "Certificate cothority: Towards trustworthy collective CAS," in *Proc. Hot Topics Privacy Enhanc. Technol. (HotPETs)*, vol. 7, 2015, pp. 1–2.

[87] E. Syta et al., "Keeping authorities 'honest or bust' with decentralized witness cosigning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 526–545.

[88] L. Chuat, C. Krähenbühl, P. Mittal, and A. Perrig, "F-PKI: Enabling innovation and trust flexibility in the HTTPs public-key infrastructure," 2021, *arXiv:2108.08581*.

[89] L. Wang, G. Asharov, R. Pass, T. Ristenpart, and A. Shelat, "Blind certificate authorities," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 1015–1032.

[90] E. F. Kfoury, D. Khoury, A. AlSabeh, J. Gomez, J. Crichigno, and E. Bou-Harb, "A blockchain-based method for decentralizing the ACME protocol to enhance trust in PKI," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, 2020, pp. 461–465.

[91] H. Perl, S. Fahl, and M. Smith, "You won't be needing these any more: On removing unused certificates from trust stores," in *Proc. Int. Conf. Financial Cryptography Data Security*, 2014, pp. 307–315.

[92] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMAP: Fast Internet-wide scanning and its security applications," in *Proc. 22nd USENIX Conf. Security (SEC)*, 2013, pp. 605–620.

[93] B. Jayaraman, H. Li, and D. Evans, "Decentralized certificate authorities," 2017, *arXiv:1706.03370*.

[94] M. Alicherry and A. D. Keromytis, "DoubleCheck: Multi-path verification against man-in-the-middle attacks," in *Proc. IEEE Symp. Comput. Commun.*, 2009, pp. 557–563.

[95] R. Holz, T. Riedmaier, N. Kammenhuber, and G. Carle, "X.509 forensics: Detecting and localising the SSL/TLS men-in-the-middle," in *Proc. Eur. Symp. Res. Comput. Security*, 2012, pp. 217–234.

[96] M. Abadi, A. Birrell, I. Mironov, T. Wobber, and Y. Xie, "Global authentication in an untrustworthy world," in *Proc. 14th Workshop Hot Topics Oper. Syst. (HotOS XIV)*, 2013, p. 18.
[97] A. Bates, J. Pletcher, T. Nichols, B. Hollembaek, and K. R. Butler, "Forced perspectives: Evaluating an SSL trust enhancement at scale," in *Proc. Conf. Internet Meas. Conf.*, 2014, pp. 503–510.
[98] A. Bates et al., "Securing SSL certificate verification through dynamic linking," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 394–405.
[99] M. O'Neill et al., "TrustBase: An architecture to repair and strengthen certificate-based authentication," in *Proc. 26th USENIX Security Symp. (USENIX Security)*, 2017, pp. 609–624.
[100] M. Modell, A. Barz, G. Toth, and C. Loesch. "Certificate patrol." 2014. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/certificate-patrol
[101] H. Everywhere. "Electronic frontier foundation." 2014. [Online]. Available: https://www.eff.org/https-everywhere
[102] H. Alexis. "HTTPs is actually everywhere." 2021. [Online]. Available: https://www.eff.org/deeplinks/2021/09/https-actually-everywhere
[103] A. Langley. "Public key pinning." 2011. [Online]. Available: https://www.imperialviolet.org/2011/05/04/pinning.html
[104] C. Evans, C. Palmer, and R. Sleevi, "Public key pinning extension for HTTP," IETF, RFC 7469, Apr. 2015. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7469.txt
[105] C. Palmer. "Intent to deprecate and remove: Public key pinning." 2017. [Online]. Available: https://groups.google.com/a/chromium.org/g/blink-dev/c/he9tr7p3rZ8?pli=1
[106] J. Hodges, C. Jackson, and A. Barth, "HTTP strict transport security (HSTS)," Internet Eng. Task Force, RFC 6797, 2018.
[107] M. Marlinspike and T. Perrin. "Trust assertions for certificate keys." 2013. [Online]. Available: http://tools.ietf.org/id/draft-perrin-tls-tack-02.txt
[108] X. Wang, Y. Bai, and L. Hu, "Domain based certification and revocation," in *Proc. Int. Conf. Security Manag. (SAM)*, 2015, p. 272.
[109] B. Li, W. Wang, L. Meng, J. Lin, X. Liu, and C. Wang, "ELAPHURUS: Ensemble defense against fraudulent certificates in TLS," in *Proc. Int. Conf. Inf. Security Cryptol.*, 2019, pp. 246–259.
[110] M. Zhang et al., "Detecting and measuring security risks of hosting-based dangling domains," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 1, p. 247, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3579440
[111] R. Li et al., "A longitudinal and comprehensive measurement of DNS strict privacy," *IEEE/ACM Trans. Netw.*, early access, Apr. 3, 2023, doi: 10.1109/TNET.2023.3262651.
[112] X. Li et al., "Ghost domain reloaded: Vulnerable links in domain name delegation and revocation," in *Proc. 30th Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*. 2013, pp. 1–8. [Online]. Available: https://doi.org/10.14722/ndss
[113] D. Basin, C. Cremers, T. H.-J. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "ARPKI: Attack resilient public-key infrastructure," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, Nov. 2014, pp. 382–393. [Online]. Available: https://doi.org/10.1145/2660267.2660298
[114] D. Basin, C. Cremers, T. H.-J. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "Design, analysis, and implementation of ARPKI: An attack-resilient public-key infrastructure," *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 3, pp. 393–408, May/Jun. 2016.
[115] P. Szalachowski, S. Matsumoto, and A. Perrig, "PoliCert: Secure and flexible TLS certificate management," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 406–417.
[116] J. Yu, V. Cheval, and M. Ryan, "DTKI: A new formalized PKI with verifiable trusted parties," *Comput. J.*, vol. 59, no. 11, pp. 1695–1713, 2016.
[117] J. Chen, S. Yao, Q. Yuan, R. Du, and G. Xue, "Checks and balances: A tripartite public key infrastructure for secure Web-based connections," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2017, pp. 1–9.
[118] S. Khan, Z. Zhang, L. Zhu, M. Li, Q. G. K. Safi, and X. Chen, "Accountable and transparent TLS certificate management: An alternate public-key infrastructure with verifiable trusted parties," *Security Commun. Netw.*, vol. 2018, Jun. 2018, Art. no. 8527010. [Online]. Available: https://doi.org/10.1155/2018/8527010
[119] S. Khan, L. Zhu, Z. Zhang, M. A. Rahim, K. Khan, and M. Li, "Attack-resilient TLS certificate transparency," *IEEE Access*, vol. 8, pp. 98958–98973, 2020.
[120] S. Khan, Z. Zhang, L. Zhu, M. A. Rahim, S. Ahmad, and R. Chen, "SCM: Secure and accountable TLS certificate management," *Int. J. Commun. Syst.*, vol. 33, no. 15, Jul. 2020, Art. no. e4503. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4503
[121] X. Wang and M. El-Said, "DomainPKI: Domain aware certificate management," in *Proc. 21st Annu. Conf. Inf. Technol. Educ.*, 2020, pp. 419–425.
[122] A. Yakubov et al., "A blockchain-based PKI management framework," in *Proc. 1st IEEE/IFIP Int. Workshop Manag. Manag. Blockchain (Man2Block) IEEE/IFIP NOMS*, Apr. 2018, pp. 1–7.
[123] Z. Wang, J. Lin, Q. Cai, Q. Wang, D. Zha, and J. Jing, "Blockchain-based certificate transparency and revocation transparency," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 1, pp. 681–697, Jan./Feb. 2017.
[124] H. Tewari, A. Hughes, S. Weber, and T. Barry, "X509cloud—Framework for a ubiquitous PKI," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2017, pp. 225–230.
[125] J. Chen, S. Yao, Q. Yuan, K. He, S. Ji, and R. Du, "CertChain: Public and efficient certificate audit based on blockchain for TLS connections," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2018, pp. 2060–2068.
[126] D. Madala, M. P. Jhanwar, and A. Chattopadhyay, "Certificate transparency using blockchain," in *Proc. IEEE Int. Conf. Data Min. Workshops (ICDMW)*, 2018, pp. 71–80.
[127] Y. Dong, W. Kim, and R. Boutaba, "ConiFER: Centrally-managed PKI with blockchain-rooted trust," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2018, pp. 1092–1099.
[128] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "CertLedger: A new PKI model with certificate transparency based on blockchain," *Comput. Security*, vol. 85, pp. 333–352, Aug. 2019.
[129] P. Szalachowski, "PADVA: A blockchain-based TLS notary service," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, 2019, pp. 836–843.
[130] Z. Xiong, Z. L. Jiang, S. Yang, X. Wang, and J. Fang, "SSHTDNS: A secure, scalable and high-throughput domain name system via blockchain technique," in *Proc. Int. Conf. Netw. Syst. Security*, 2019, pp. 272–287.
[131] A. Garba, Q. Hu, Z. Chen, and M. R. Asghar, "BB-PKI: Blockchain-based public key infrastructure certificate management," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun. IEEE 18th Int. Conf. Smart City IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, 2020, pp. 824–829.
[132] J. Zhao, Z. Lin, X. Huang, Y. Zhang, and S. Xiang, "TRUSTCA: Achieving certificate transparency through smart contract in blockchain platforms," in *Proc. Int. Conf. High Perform. Big Data Intell. Syst. (HPBD&IS)*, 2020, pp. 1–6.
[133] G.-H. Hwang, T.-K. Chang, and H.-W. Chiang, "A semidecentralized PKI system based on public blockchains with automatic indemnification mechanism," *Security Commun. Netw.*, vol. 2021, Oct. 2021, Art. no. 7400466.
[134] N. Johnson and V. Griffith. "Ethereum name service." 2018. [Online]. Available: https://docs.ens.domains
[135] Z. Guan, A. Garba, A. Li, Z. Chen, and N. Kaaniche, "AUTHLEDGER: A novel blockchain-based domain name authentication scheme," in *Proc. ICISSP*, 2019, pp. 345–352.
[136] K. Lewison and F. Corella. "Backing rich credentials with a blockchain PKI." 2016. [Online]. Available: https://pomcor.com/
[137] A. Garba, Z. Chen, Z. Guan, and G. Srivastava, "LightLedger: A novel blockchain-based domain certificate authentication and validation scheme," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1698–1710, Apr.–Jun. 2021.
[138] M. Caldeira and M. Correia, "Blockchain address transparency with DNS," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2021, pp. 1–7.
[139] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "KORGAN: An efficient PKI architecture based on PBFT through dynamic threshold signatures," *Comput. J.*, vol. 64, no. 4, pp. 564–574, 2021.
[140] S. Kakei, Y. Shiraishi, M. Mohri, T. Nakamura, M. Hashimoto, and S. Saito, "Cross-certification towards distributed authentication infrastructure: A case of hyperledger fabric," *IEEE Access*, vol. 8, pp. 135742–135757, 2020.
[141] T. Sermpinis, G. Vlahavas, K. Karasavvas, and A. Vakali, "DETRACT: A decentralized, transparent, immutable and Open PKI certificate framework," *Int. J. Inf. Security*, vol. 20, no. 4, pp. 553–570, 2021.
[142] Z. Zhai, S. Shen, and Y. Mao, "BPKI: A secure and scalable blockchain-based public key infrastructure system for Web services," *J. Inf. Security Appl.*, vol. 68, Aug. 2022, Art. no. 103226. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212622000990

[143] S. Eskandarian, E. Messeri, J. Bonneau, and D. Boneh, "Certificate transparency with privacy," 2017, *arXiv:1703.02209*.

[144] H. Leibowitz, H. Ghalwash, E. Syta, and A. Herzberg, "CTNG: Secure certificate and revocation transparency," in *Proc. Cryptol. ePrint Archive*, 2021, p. 4.

[145] T. Smith, L. Dickinson, and K. Seamons, "Let's revoke: Scalable global certificate revocation," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, 2020, pp. 1–7.

[146] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *Proc. 25th Int. Conf. Comput.-Aided Verification (CAV)*, vol. 8044, Jul. 2013, pp. 696–701.

[147] M. D. Ryan, "Enhanced certificate transparency and end-to-end encrypted mail," in *Proc. NDSS*, 2014, pp. 1–14.

[148] S. Matsumoto and R. M. Reischuk, "Certificates-as-an-insurance: Incentivizing accountability in SSL/TLS," in *Proc. NDSS Workshop Security Emerg. Netw. Technol. (SENT)*, 2015, p. 9.

[149] S. Yao, J. Chen, K. He, R. Du, T. Zhu, and X. Chen, "PBCERT: Privacy-preserving blockchain-based certificate status validation toward mass storage management," *IEEE Access*, vol. 7, pp. 6117–6128, 2018.

[150] E. Karaarslan and E. Adiguzel, "Blockchain based DNS and PKI solutions," *IEEE Commun. Stand. Mag.*, vol. 2, no. 3, pp. 52–57, Sep. 2018.

[151] Z. Wang, J. Lin, Q. Cai, Q. Wang, J. Jing, and D. Zha, "Blockchain-based certificate transparency and revocation transparency," in *Proc. Int. Conf. Financial Cryptography Data Security*, 2018, pp. 144–162.

[152] Z. Wan, Z. Guan, F. Zhuo, and H. Xian, "BKI: Towards accountable and decentralized public-key infrastructure with blockchain," in *Proc. Int. Conf. Security Privacy Commun. Syst.*, 2017, pp. 644–658.

[153] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman, "CONIKS: Bringing key transparency to end users," in *Proc. 24th USENIX Security Symp. (USENIX Security)*, 2015, pp. 383–398.

[154] A. Tomescu and S. Devadas, "CATENA: Efficient non-equivocation via bitcoin," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 393–409.

[155] L. Dykcik, L. Chuat, P. Szalachowski, and A. Perrig, "BlockPKI: An automated, resilient, and transparent public-key infrastructure," in *Proc. IEEE Int. Conf. Data Min. Workshops (ICDMW)*, 2018, pp. 105–114.

[156] R. Xu and J. Joshi, "Trustworthy and transparent third-party authority," *ACM Trans. Internet Technol.*, vol. 20, no. 4, p. 31, Oct. 2020. [Online]. Available: https://doi.org/10.1145/3386262

[157] B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, and W. Shi, "CeCoin: A decentralized PKI mitigating MITM attacks," *Future Gener. Comput. Syst.*, vol. 107, pp. 805–815, Jun. 2020.

[158] G. Slepak. "DNSchain + okturtles." 2013. [Online]. Available: http://okturtles.com/other/dnschain_okturtles_overview.pdf

[159] A. Loibl and J. Naab. "Namecoin." 2014. [Online]. Available: https://www.namecoin.org/

[160] J. Benet, "IPFS-content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

[161] Y. G. Malahov. "Bitalias 1, AKA usernames for bitcoin, a new, simple naming system for bitcoin addresses." Mar. 2017. [Online]. Available: https://medium.com/bitalias-decentralized-naming-and-identity-service/bitalias-7b66bffed9d8

[162] A. Muneeb and S. Ryan. "ONENAME—Bringing decentralization to identity with blockchain id." Accessed: May 12, 2022. [Online]. Available: https://epicenter.tv/episodes/101/

[163] F. Li, Z. Liu, T. Li, H. Ju, H. Wang, and H. Zhou, "Privacy-aware PKI model with strong forward security," *Int. J. Intell. Syst.*, vol. 37, no. 12, pp. 10049–10065, 2022.

[164] P. Plessing and O. Omolola, "Revisiting privacy-aware blockchain public key infrastructure," in *Proc. ICISSP*, 2020, pp. 415–423.

[165] R. Housley et al., "Internet x.509 public key infrastructure certificate and CRL profile," IETF, RFC 2459, Jan. 1999.

[166] P. Wohlmacher, "Digital certificates: A survey of revocation methods," in *Proc. ACM Workshops Multimedia*, 2000, pp. 111–114.

[167] G. Jain. "Certificate revocation: A survey." 2000. [Online]. Available: http://citeseer.ist.psu.edu/511985.html

[168] D. A. Cooper, "A model of certificate revocation," in *Proc. 15h Annu. Comput. Security Appl. Conf. (ACSAC)*, 1999, pp. 256–264.

[169] D. A. Cooper, "A more efficient use of delta-CRLs," in *Proc. IEEE Symp. Security Privacy (S&P)*, 2000, pp. 190–202.

[170] R. J. Perlman and C. W. Kaufman, "Method of issuance and revocation of certificates of authenticity used in public key networks and other systems," U.S. Patent 5 261 002, Nov. 9, 1993.

[171] S. Micali, "Efficient certificate revocation," Lab. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Rep. MIT/LCS/TM-542b, 1995.

[172] S. Micali, "Scalable certificate validation and simplified PKI management," in *Proc. 1st Annu. PKI Res. Workshop*, vol. 15, 2002, p. 8.

[173] *FIPS PUB 180-1*," National Inst. Stand. Technol., Gaithersburg, MD, USA, 1995.

[174] W. Aiello, S. Lodha, and R. Ostrovsky, "Fast digital identity revocation," in *Proc. Annu. Int. Cryptol. Conf.*, 1998, pp. 137–152.

[175] P. C. Kocher, "On certificate revocation and validation," in *Proc. Int. Conf. Financial Cryptography*, 1998, pp. 172–177.

[176] R. C. Merkle, "A certified digital signature," in *Proc. Conf. Theory Appl. Cryptol.*, Jul. 1989, pp. 218–238.

[177] M. Naor and K. Nissim, "Certificate revocation and certificate update," in *Proc. 7th Conf. USENIX Security Symp.*, vol. 7, 1998, p. 17.

[178] M. Naor and K. Nissim, "Certificate revocation and certificate update," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 561–570, Apr. 2000.

[179] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet public key infrastructure online certificate status protocol-OCSP," IETF, RFC 6960, 1999.

[180] C. Ekechukwu, D. Lindskog, and R. Ruhl, "A notary extension for the online certificate status protocol," in *Proc. IEEE Int. Conf. Soc. Comput.*, 2013, pp. 1016–1021.

[181] P. McDaniel and S. Jamin, "Windowed certificate revocation," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 9th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 3, 2000, pp. 1406–1414.

[182] A. Buldas, P. Laud, and H. Lipmaa, "Accountable certificate management using undeniable attestations," in *Proc. 7th ACM Conf. Comput. Commun. Security*, 2000, pp. 9–17.

[183] I. Gassko, P. S. Gemmell, and P. MacKenzie, "Efficient and fresh certification," in *Proc. Int. Workshop Public Key Cryptography*, 2000, pp. 342–353.

[184] P. Zheng, "Tradeoffs in certificate revocation schemes," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 103–112, Apr. 2003. [Online]. Available: https://doi.org/10.1145/956981.956991

[185] E. Faldella and M. Prandini, "A novel approach to on-line status authentication of public-key certificates," in *Proc. IEEE 16th Annu. Comput. Security Appl. Conf. (ACSAC)*, 2000, pp. 270–277.

[186] R. N. Wright, P. D. Lincoln, and J. K. Millen, "Efficient fault-tolerant certificate revocation," in *Proc. 7th ACM Conf. Comput. Commun. Security*, 2000, pp. 19–24.

[187] D. Boneh, X. Ding, G. Tsudik, and C.-M. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Proc. USENIX Security Symp.*, 2001, p. 22.

[188] J. L. Munoz, J. Forne, O. Esparza, and M. Soriano, "Certificate revocation system implementation based on the Merkle hash tree," *Int. J. Inf. Security*, vol. 2, no. 2, pp. 110–124, 2004. [Online]. Available: https://doi.org/10.1007/s10207-003-0026-4

[189] F. F. Elwailly, C. Gentry, and Z. Ramzan, "QuasiModo: Efficient certificate validation and revocation," in *Proc. Public Key Cryptography (PKC)*, 2004, pp. 375–388.

[190] A. Langley. "Revocation checking and Chrome's CRL." 2012. [Online]. Available: https://www.imperialviolet.org/2012/02/05/crlsets.html

[191] M. Goodwin. "Revoking intermediate certificates: Introducing OneCRL." 2015. [Online]. Available: https://wiki.mozilla.org/CA:RevocationPlan#OneCRL

[192] J. Larisch, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "CRLITE: A scalable system for pushing all TLS revocations to all browsers," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 539–556.

[193] R. L. Rivest, "Can we eliminate certificate revocation lists?" in *Proc. Int. Conf. Financial Cryptography*, 1998, pp. 178–183.

[194] Y.-K. Hsu and S. Seymour, "Intranet security framework based on short-lived certificates," in *Proc. IEEE 6th Workshop Enabling Technol. Infrastruct. Collaborative Enterprises*, 1997, pp. 228–234.

[195] E. Topalovic, B. Saeta, L.-S. Huang, C. Jackson, and D. Boneh. "Towards short-lived certificates." 2012. [Online]. Available: https://www.ieee-security.org/TC/W2SP/2012/papers/w2sp12-final9.pdf

[196] D. Eastlake et al., "Transport layer security (TLS) extensions: Extension definitions," IETF, RFC 6066, Jan. 2011.

[197] P. Hallam-Baker, "X.509v3 transport layer security (TLS) feature extension," IETF, RFC 7633, 2015.

[198] Q. Hu, M. R. Asghar, and N. Brownlee, "Certificate revocation guard (CRG): An efficient mechanism for checking certificate revocation," in *Proc. IEEE 41st Conf. Local Comput. Netw. (LCN)*, 2016, pp. 527–530.

[199] P. Szalachowski, L. Chuat, T. Lee, and A. Perrig, "RITM: Revocation in the middle," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2016, pp. 189–200.
[200] A. Schulman, D. Levin, and N. Spring, "REVCAST: Fast, private certificate revocation over FM radio," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 799–810.
[201] A. Alrawais, A. Alhothaily, X. Cheng, C. Hu, and J. Yu, "SecureGuard: A certificate validation system in public key infrastructure," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5399–5408, Jun. 2018.
[202] B. Laurie and E. Kasper, *Revocation Transparency*, Google Res., Menlo Park, CA, USA, Sep. 2012.
[203] P. Szalachowski, L. Chuat, and A. Perrig, "PKI safety net (PKISN): Addressing the too-big-to-be-revoked problem of the TLS ecosystem," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, 2016, pp. 407–422.
[204] A. Singh, B. Sengupta, and S. Ruj, "Certificate transparency with enhancements and short proofs," in *Proc. Aust. Conf. Inf. Security Privacy*, 2017, pp. 381–389.
[205] A. Garba, A. Bochem, and B. Leiding, "BlockVoke—Fast, blockchain-based certificate revocation for PKIS and the Web of trust," in *Proc. Int. Conf. Inf. Security*, 2020, pp. 315–333.
[206] M. Jia et al., "PROCESS: Privacy-preserving on-chain certificate status service," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2021, pp. 1–10.
[207] Y. C. E. Adja, B. Hammi, A. Serhrouchni, and S. Zeadally, "A blockchain-based certificate revocation management and status verification system," *Comput. Security*, vol. 104, May 2021, Art. no. 102209. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740482100033X
[208] X. Ge, L. Wang, W. An, X. Zhou, and B. Li, "CRchain: An efficient certificate revocation scheme based on blockchain," in *Algorithms and Architectures for Parallel Processing*, Y. Lai, T. Wang, M. Jiang, G. Xu, W. Liang, and A. Castiglione, Eds. Cham, Switzerland: Springer Int., 2022, pp. 453–472.
[209] D. Kumar et al., "Tracking certificate Misissuance in the wild," in *Proc. IEEE Symp. Security Privacy (SP)*, Jul. 2018, pp. 785–798.
[210] J. Purushothaman, E. Thompson, and A. Abdou, "Position paper: Certificate root stores—An area of unity or disparity?" in *Proc. 15th Workshop Cyber Security Exp. Test (CSET)*, 2022, pp. 105–110. [Online]. Available: https://doi.org/10.1145/3546096.3546110
[211] R. Li, Z. Zhang, J. Shao, R. Lu, X. Jia, and G. Wei, "The potential harm of email delivery: Investigating the HTTPs configurations of webmail services," *IEEE Trans. Depend. Secure Comput.*, early access, Feb. 20, 2023, doi: 10.1109/TDSC.2023.3246600.
[212] M. Luo, B. Feng, L. Lu, E. Kirda, and K. Ren, "On the complexity of the Web's PKI: Evaluating certificate validation of mobile browsers," *IEEE Trans. Depend. Secure Comput.*, early access, Mar. 13, 2023, doi: 10.1109/TDSC.2023.3255869.

Salabat Khan received the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China. He was a Postdoctoral Fellowship with the College of Computer Science and Software Engineering, Shenzhen University, China, where he is currently working as an Associate Researcher with the College of Computer Science and Software Engineering. His current research interests include security and privacy, VPKI, PKIX, cryptographic algorithms, blockchain, and distributed ledger technologies.

Fei Luo received the B.Sc. degree from the Jiangxi University of Science and Technology, the M.Sc. degree in surveying and mapping from Wuhan University, and the Ph.D. degree from the Queen Mary University of London, London, U.K., in 2020. He is currently working as a Postdoctoral Fellow with Shenzhen University. His research interests include geographic information systems, human activity detection, and machine learning.

Zijian Zhang (Member, IEEE) is currently an Associate Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology, China. He was a Research Fellow with the University of Auckland and a Visiting Scholar with the State University of New York at Buffalo. His research interests include authentication and identification, consensus mechanisms, and privacy computing.

Farhan Ullah received the Ph.D. degree from Shanghai University, Shanghai, China, in 2020. He is currently working as a Postdoctoral Fellow with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China. His research interests include deep learning, hyperspectral image classification, computer vision, and recommender systems.

Farhan Amin received the B.S. degree in computer science from Gomal University, Dera Ismail Khan, Pakistan, in 2007, the M.S. degree in computer science from International Islamic University Islamabad in August 2012, and the Ph.D. degree from the Department of Information and Communication Engineering, College of Engineering, Yeungnam University, Gyeongsan, South Korea, in October 2020. He had worked as an Assistant Professor with the Department of Computer Engineering, Gachon University, South Korea. He is currently working as an Assistant Professor with the Department of Information and Communication Engineering, Yeungnam University. He has over nine years of teaching and research experience. His research interests include the Internet of Things, social Internet of Things, big data, and data science. He was a recipient of a Fully-Funded Scholarship for Master's Studies and Ph.D. He is a member of ACM.

Syed Furqan Qadri received the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2019. He did a Postdoctoral Fellowship with Shenzhen University and is currently working as an Associate Researcher with the Research Center for Healthcare Data Science, Zhejiang Lab, Hangzhou, China. He has published more than 35 research papers, with Google Scholar citations more than 720 times, an H-index of 16, and one paper that was selected as an ESI highly cited paper. His research interests include computer vision, applied deep learning, image segmentation, classification, medical image processing, and pattern recognition. He has been a Reviewer of many journals, such as IEEE TRANSACTIONS ON MEDICAL IMAGING, *Artificial Intelligence in Medicine*, *Engineering Applications of Artificial Intelligence*, *Expert Systems with Applications*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, *Computer Methods and Programs in Biomedicine*, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, *Cancers*, and *Viruses*, and also many other journals from IEEE, Elsevier, Wiley, Springer, SPIE, Taylor & Francis, MDPI, Hindawi, PLOS, and Tech Science Press.

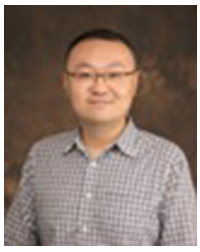
Md Belal Bin Heyat received the B.Tech. degree in electronics and instrumentation and the M.Tech. degree in electronic circuits and systems from Integral University, Lucknow, India, in 2014 and 2016, respectively, and the Ph.D. degree in electronic science and technology from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China.

He received nine awards and also worked as a Research Associate, the Vice-Chairman of the UESTC Country League, and Country Representative of India during his Ph.D. at the University of Electronic Science and Technology of China. He is currently working as a Postdoctoral Researcher with the IoT Research Center, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China. In addition, he was a Visiting Postdoctoral Researcher with CVEST, IIIT Hyderabad, India, and a Faculty Member with the Department of Science and Engineering, Novel Global Community Educational Foundation, NSW, Australia. He has published more than fifty articles in reputed international journals and conferences. He is engaged in research in the areas of detection, sleep disorders, neurological disorders, psycho-neurological disorders, cardiovascular diseases, signal processing, and medical machine learning. He has been serving as a Guest Editor/Reviewer for over 15 journals, including *Applied Sciences*, *Journal of Integrative Neuroscience*, *Electronics*, *Computers in Biology and Medicine*, IEEE ACCESS, *Sensors*, *Reviews in Cardiovascular Medicine*, *International Journal of Environmental Research and Public Health*, *Journal of Healthcare Engineering*, and *Artificial Intelligence*.

Rukhsana Ruby received the master's degree from the University of Victoria, Canada, in 2009, and the Ph.D. degree from The University of British Columbia, Canada, in 2015. She has authored nearly 75 technical papers of well-recognized journals and conferences. From the broader aspect, her resource interests include the management and optimization of next generation wireless networks. She is the recipient of several awards or honors, notable among which are Canadian NSERC Postdoctoral Fellowship, IEEE Exemplary Certificate (*IEEE Communications Letters* in 2018 and IEEE WIRELESS COMMUNICATIONS LETTERS in 2018) and an Outstanding Reviewer Certificate (*Elsevier Computer Communications* in 2017). She has served as a Lead Guest Editor for the special issue on NOMA techniques under *EURASIP Journal on Wireless Communications and Networking* in 2017, and currently is serving as an Associate Editor *EURASIP Journal on Wireless Communications and Networking*. Besides, she has also been serving as a technical program committee member in various reputed conferences.

Lu Wang received the B.S. degree in communication engineering from Nankai University, Tianjin, China, in 2009, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2013. She is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include wireless communications and mobile computing.

Shamsher Ullah received the Ph.D. degree from the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China. He received the Postdoctoral Certificate from the School of Software, Northwestern Polytechnical University, Xian, China. He is currently working as a Postdoctoral Research Fellow with the National Engineering Laboratory for Big Data System Computing Technology, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research work has been published in reputed journals and conferences. His research interest includes cryptography, information security, privacy, data trading, e-commerce, and e-health.



Meng Li (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from the School of Computer Science and Technology, Beijing Institute of Technology, China, in 2019. He is an Associate Professor and the Dean Assistant with the School of Computer Science and Information Engineering, Hefei University of Technology, China. He is also a Postdoctoral Researcher with the Department of Mathematics and HIT Center, University of Padua, Italy, where he is with the Security and PRivacy Through Zeal Research Group led by Prof. M. Conti (IEEE Fellow). He was sponsored by the ERCIM Alain Bensoussan Fellowship Program (from 1 October 2020 to 31 March 2021) to conduct postdoctoral research supervised by Prof. F. Martinelli with CNR, Italy. He was sponsored by the China Scholarship Council (from 1 September 2017 to 31 August 2018) for a joint Ph.D. study supervised by Prof. X. Lin (IEEE Fellow) in the Broadband Communications Research Lab, University of Waterloo and Wilfrid Laurier University, Canada. His research interests include security, privacy, fairness, applied cryptography, cloud computing, edge computing, blockchain, and vehicular networks. In this area, he has published more than 60 papers in international peer-reviewed transactions, journals, and conferences, including IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Database Systems*, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON SMART GRID, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, MobiCom, ICICS, SecureComm, TrustCom, and IPCCC. He is an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT.

Victor C. M. Leung (Life Fellow, IEEE) is a Distinguished Professor of Computer Science and Software Engineering with Shenzhen University, China. He is also an Emeritus Professor of Electrical and Computer Engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, University of British Columbia, Canada. His published works have together attracted more than 50,000 citations. His research is in the broad areas of wireless networks and mobile systems, and he has published widely in these areas. He received the 1977 APEBC Gold Medal, the NSERC Postgraduate Scholarships from 1977 to 1981, the IEEE Vancouver Section Centennial Award, the UBC Killam Research Prize in 2011, the Canadian Award for Telecommunications Research in 2017, the IEEE TCGCC Distinguished Technical Achievement Recognition Award in 2018, and the ACM MSWiM Reginald Fessenden Award in 2018. He coauthored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the IEEE SYSTEMS JOURNAL Best Paper Award in 2017, the IEEE CSIM Best Journal Paper Award in 2018, and the IEEE TCGCC Best Journal Paper Award in 2019. He is named in the current Clarivate Analytics list of "Highly Cited Researchers." He is serving on the editorial boards of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE ACCESS, and several other journals. He is a Fellow of the Royal Society of Canada (Academy of Science), the Canadian Academy of Engineering, and the Engineering Institute of Canada.

Kaishun Wu (Fellow, IEEE) received the Ph.D. degree in computer science and engineering from HKUST in 2011. After that, he worked as a Research Assistant Professor with HKUST. In 2013, he joined SZU as a Distinguished Professor. He is currently working as the Vice President of Research with HKUST (Guanzhou), China, and also with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen. He has coauthored two books and published more than 100 high-quality research papers in international leading journals and primer conferences, like IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, ACM MobiCom, and IEEE INFOCOM. He is an inventor of six U.S. and over 100 Chinese pending patents. He received the 2012 Hong Kong Young Scientist Award, the 2014 Hong Kong ICT Awards: Best Innovation, and the 2014 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award.