Evolving Role of PKI in Facilitating Trust

Vishwas T. Patil
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
ivishwas@gmail.com

R.K. Shyamasundar
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
shyamasundar@gmail.com

2022 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA) | 978-1-6654-8883-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/PKIA56009.2022.9952249

Abstract*—A digital certificate is by far the most widely used artifact to establish secure electronic communication over the Internet. It certifies to its user that the public key encapsulated in it is associated with the subject of the certificate. A Public Key Infrastructure (PKI) is responsible to create, store, distribute, and revoke digital certificates. To establish a secure communication channel two unfamiliar entities rely on a common certificate issuer (a part of PKI) that vouches for both entities' certificates – thus authenticating each other via public keys listed in each other's certificates. Therefore, PKIs act as a trusted third party for two previously unfamiliar entities. Certificates are static data structures, their revocation status must be checked before usage; this step inadvertently involves a PKI for every secure channel establishment – leading to privacy violations of relying parties. As PKIs act as trust anchors for their subjects, any inadvertent event or malfeasance in PKI setup breaches the trust relationship leading to identity theft. Alternative PKI trust models, like PGP and SPKI, have been proposed but with limited deployment. With several retrofitting amendments to the prevalent X.509 standard, the standard has been serving its core objective of entity authentication but with modern requirements of contextual authentication, it is falling short to accommodate the evolving requirements. With the advent of blockchain as a trust management protocol, the time has come to rethink flexible alternatives to PKI core functionality; keeping in mind the modern-day requirements of contextual authentication-cum-authorization, weighted trust anchors, privacy-preservation, usability, and cost-efficient key management. In this paper, we assess this technology's complementary role in modern-day evolving security requirements. We discuss the feasibility of re-engineering PKIs with the help of blockchains, and identity networks.

***Index Terms*—security, authentication, authorization, privacy.**

I. Introduction

Upon the invention of RSA public-key cryptography in 1977 the problem of communicating a secret (e.g., a symmetric key) over an untrusted communication channel was effectively addressed [1]. However, such secure communication is possible only when the recipient's public key is known a priori. Therefore, public-key cryptography on its own was not scalable. The challenge was to reliably deliver a public key (which does not require confidentiality but requires authenticity) to any previously unfamiliar entity. In 1988, ITU-T addressed this scalability issue by releasing a series of X.500 standards as electronic directory services; using which one can request the public key of an entity whose name-key binding is available in the X.500 directory. This name-key binding is called a digital certificate and its issuer is called a certificate authority (CA) who vouches for all such bindings. Two such CAs may cross-certify each other by issuing a name-key binding to each other's public keys. The procedural tasks of a CA and associated entities are provided in X.509 standard. X.509 is an ITU-T standard that outlines the set of roles, policies, hardware, software, and procedures needed to manage certificates; collectively called PKI. Several organizations implement this standard as a PKI service and offer key management by issuing digital certificates either for a fee or for free. Based on the business model of a PKI service provider, it is imperative to imagine the level of due diligence done by the certificate issuer before signing a name-key association as a digital certificate of a customer. For example, the X.509 extended validation certificate requires a customer to furnish physical certificates issued by local governments in order to prove to the CA that the customer is indeed a legal entity that exists in the real world. Through extended validation, the CA takes additional validation steps to ascertain that the customer is true as described in its certificate signing request. Thus, the issuance of a digital certificate by a CA is a summary representation of the CA's assessment of the entity from the real world without specifying the assessment trail in the issued certificate. The user who relies on a digital certificate of an entity explicitly trusts the issuing CA of that certificate. When the relying user comes across a certificate issued by a previously unfamiliar CA who has issued that certificate, the PKI provides methods to navigate through cross-certificates if there exists a certificate link between the unfamiliar CA and the CA the relying party trusts – known as certificate path discovery. PKI as a whole is a representation of who vouches for whom and who trusts whom – a web of trust. Trust is a subjective matter [2]. From the relying party's viewpoint, the trust assertions made by PKIs are independent of relying party's assessment criteria for an entity of interest. The prevalent X.509 standard-based PKIs provide a binary assessment of the trustworthiness of an entity in question. Though the efforts like extended validation and other extensions incorporated into the standard provide a little more information about the entity; it is still rigid and the relying party has to find alternative ways to achieve higher levels of trust. PKI is a conglomeration of several service providers coming together to help end users retrieve and validate public keys (identity). A rouge component in the ecosystem has the potential to undermine the whole ecosystem [3]–[7].

Our entire digital economy relies on PKIs. PKIs form the building block of the security protocols responsible to provide security properties to digital information. Therefore, it is paramount to ensure that it is catering well to the evolving needs of our digital world [8]. The primary task of a PKI is to act as a trusted third party (TTP) between two unfamiliar parties who want to communicate securely. It does so by acting as a trusted intermediary for them – a *conduit of trust*. There are tens of private organizations that provide PKI services over the Internet. These private entities are prone to the fallibility of malfeasance motivated by financial gain, state-sponsored espionage, or simply lack of oversight. Thus, these custodians of our trust must be studied for their trust propagation models so that risks can be identified and appropriate mitigation measures are put in place. Bruce Schneier's analogy that *an organization's overall security is only as strong as its weakest link in the security chain* can be viewed as *the strength of a security chain is derived from the strength of its constituent links; and, the underlying PKIs characterizes their fragility.*

In this paper, we are going to put forward our perspective on this technology's evolution in the past four decades and how it has been relied upon to achieve security. We shall also provide a survey on research works that achieve authentication, authorization, and access control policy specification using PKIs. This paper shall provide a broad overview of this technology's usage in current practices to achieve important security properties. It will help practitioners to make judicious decisions while deploying their systems that need to be concerned with their system's overall efficiency, user privacy, key management costs, and the ease of policy specification.

II. PKI Components, Functionality & Limitations

The PKI that we rely upon in our present-day online interactions is an implementation of the X.509 standard. It has served well in e-commerce type of applications, especially through the SSL/TLS protocol. Though SSL/TLS allows mutual authentication of entities present at either end of a communication, in most cases, only one entity authenticates the other by looking at the digital certificate provided at the time of TLS connection establishment request. This is a common scenario in e-commerce and banking portals. The end-user is authenticated using the traditional method of user login and password; the reason is that digital certificates are difficult to manage by laymen. As we usher into an ever-expanding digital lifestyle, the demand for secure & authenticated interaction has increased. The rigidity of digital certificates comes as a bottleneck. The static data structure falls short to encapsulate user attributes required to enforce intelligent, contextual interactions. PKI will continue to offer its authentication service but the evolving technological landscape has requirements beyond the static authentication mechanism offered by traditional PKI [9]–[12]. In the following, we explain the components of a PKI, their functional role [13], [14], and their limitations that manifest in the overall security and privacy of online communications.

A. Components of a PKI

1) **Root/CA (Certification Authority):**CA is responsible for the issuance of a digital certificate to an entity that has provided the required proof. The proof could be a real-world government-issued certificate or a digital capability like ownership of an email address or a domain name. A root CA is the top trust anchor in the trust hierarchy of a PKI. The root CA issues a certificate to itself; i.e., the subject and issuer of this certificate are the same. In browsers, all root CA certificates act as a trust store, against which the top-level CA certificates are validated, and likewise the CAs at the next level. A virus altering the trust store compromises the security of that user's communication. When a browser is downloaded by a user, the trust store comes preinstalled in it – thus browser providers define the trust universe of their users. For the sake of speed, some browsers cache certificates of CAs that are not part of the trust store. This unintentionally influences the end user's ability to authenticate websites. End users seldom check their trust store. The List of Trusted Lists (ETSI TS 119 612) is an initiative by member states of the European Economic Area to publish trusted lists of qualified trust service providers in accordance with the eIDAS Regulation.
2) **RA (Registration Authority):** RA is responsible to verify the credentials supplied by a subject for certificate issuance. Upon verification of the proof of identity either in the form of real-world credentials or in digital proof like email or domain ownership, the RA forwards the subject's certificate signing request to the CA and a digital certificate gets issued to the subject. One important observation about the prevalent practice during the registration process is that government-issued credentials across the world are not necessarily of equivalent credibility. However, RAs accept such real-world credentials with some discretion. RAs have business pressure to increase their market share and at times no alternative option either. Once two users from two different geographies and legal jurisdictions obtain their digital certificates, the entity relying on these two certificates has no means to distinguish their underlying credibility inputs. PKI is a global system built on top of validations done on proofs that can be of questionable credibility in some jurisdictions. We are using a global authentication framework (PKI) built on top of legally unequal jurisdictions in absence of a uniform global representation of real-world identity practice. European Union's eIDAS regulation is a step in this direction to evoke reliable digital identity assurance.
3) **CRLs (Certificate Revocation Lists):** CRLs are databases of certificates that are revoked by their issuers from time to time [15]. The relying party is expected to consult a CRL issued by a CA who has issued the certificate that is being validated. There exists a time window between which a certificate gets revoked

and the relying party consults the CRL. To speed up TLS connection formation, several browsers skip the certificate validation check with online CRLs. Even further, browsers cache certificate chains to speed up the certificate validation step. As trust is a subjective matter, it may improve or degrade over a period of time, depending upon the experience of the relying party. Note that the digital certificates are digital representations of a legal entity from the real world into the digital world and the digital certificates are issued upon furnishing proofs from the real world. However, once the digital certificate is issued, the RA does not keep track of the status of the furnished proofs in the real world. Usually, in the real world people change their roles, location, and even identities, and only in beneficial propositions, do they take an effort in reporting the change to RAs if at all legally mandatory. Therefore, CRLs play a very important role in global authentication infrastructure but are not necessarily consulted. Fig. 1 depicts the notion of trust erosion as time increases.

4) **Cross-Certification:** Trust is transitive in nature. Cross-certification is a process at the CA level where two CAs issue a special certificate to each other. This signifies that those two CAs trust each other. Transitively, the users of one CA implicitly trust the other CA and vice versa. CAs do not require the consent of their existing users in order to establish new trust relationships in the PKI ecosystem. Cross-certification can be unidirectional.
5) **Certification Paths:** When a user needs to verify the validity of a certificate issued by a CA with whom the user's CA has a trust relationship, the user needs to first verify the validity of the cross-certificate. This cross-certificate is called a node in the certification path. There are different ways to find certification paths: i) The path starts from the root (at level 1; IPRA - Internet Policy Registration Authority) and traverses downwards through level 2 (PCA - Policy Certificate Authorities) towards the certificate of the subject whose certificate the user is verifying. ii) The path starts from the verifier. In SSL/TLS protocol the subject provides a bundle of certificates, where each certificate corresponds to a node in the path. The verifier must verify the validity of each certificate by visiting the CRL URIs specified in each certificate in the path; as a result, CRL servers obtain information about who is verifying whom – a potential privacy hazard. There are instances when upstream CA is compromised then the verifier may not be able to notice the compromise and ends up communicating securely with an attacker. Certificate Transparency initiative is one such effort to help verifier to avoid such scenarios [16]–[20].

B. Types of X.509 Certificates

Having discussed a PKI's components, functionality, and shortcomings, we shall discuss the types of certificates they offer and their shortcomings. Digital certificates are static data structures and the assertions made by their issuer about their subject remain valid for a reasonably long period of time. This limits its expressiveness in dynamic environments. In the following, we shall explore the types of certificates that have been introduced over the period of time in the X.509 standard as a retrofit to accommodate evolving needs of businesses.

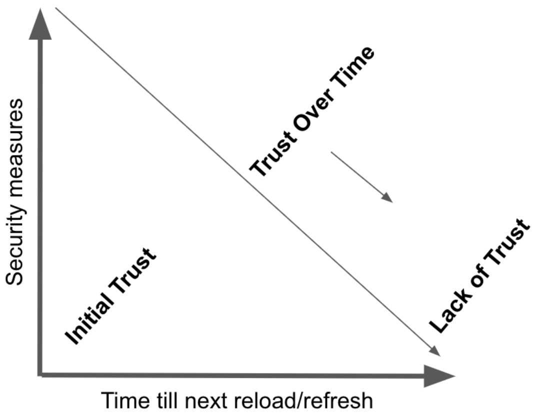

Fig. 1. Erosion of Trust over time

1) **Identity Certificates:** This is the most primitive type of certificate where the CA signs the certificate signing request of a subject. The signing request is composed of the public key of the subject and an identity string with which the subject wants to operate her public key. The CA specifies usage criteria of the subject's public key, whether the subject can issue certificates to others or just use the key for encryption of messages et al. For example, in SSL/TLS scenario, a web server in its certificate signing request to a CA specifies its FQDN (domain name) as its identity to be associated with its public key in the certificate. The web server presents such an identity certificate to its clients, who in return verify the certificate for its validity, and identity and challenge the webserver to decrypt a nonce using its private key. The web server can also force the client to authenticate themselves using their respective certificates before serving them content but it is i) costly, ii) inconvenient, iii) rigid to enforce layered access control.
2) **Attribute Certificates:** In many contexts, identity is not the criterion that is used for access control decisions; rather, the role or group membership of the accessor is the criterion used [21]. An attribute certificate of a subject is a certificate used in conjunction with the identity certificate of the subject. The issuer of the attribute certificate is called Attribute Authority and is responsible for its management. An attribute certificate is also called an authorization certificate since it is used to list out authorizations of the subject mentioned in the identity certificate. Attribute Authorities are usually organization-specific and are responsible for authorization management of that organization. This approach of access control through identity-cum-authorization is

better than identity-based-authorization as the validity of identity is relatively long-lived than the validity of authorization attributes. However, certificates being static data structures that are valid until revoked are not suitable for dynamic environments where the access decisions are dependent on external context that cannot be captured a priori.

3) **Certificate Extensions:** X.509 version 3 extensions provide methods for associating additional attributes with subjects or public keys and for managing a certification hierarchy. There are 15 standard extensions defined [13] in X.509 v3 and those can be supported by CAs out-of-the-box. The X.509 v3 certificate format also allows organizations to define private extensions [22] to carry information relevant to their applications. Each extension in a certificate is marked as either critical or non-critical. If a certificate is marked as critical and the relying party is unable to interpret the extension; it discards the certificate, which might prevent the certificate's use in a general context (identity). Non-critical extensions can be ignored by the relying party and the certificate can still be relied upon for identity verification purposes.

The fundamental objective of X.509 PKI was to deliver identity service. In the above, we have seen a retrofit to the standard to accommodate authorization information within the certificates as they assure provable delivery of identity across the Internet. A few alternative certificate formats have been specified and are used in niche environments like WAP WTLS in Enterprise WiFi, PGP for emails, DNSSEC for DNS resource records authenticity/ownership, and SPKI/SDSI for a distributed authentication-cum-authorization framework [23], [24]. The phenomenal rise of our digital lifestyle was not possible without the security guarantees to our digital information and X.509 PKI continues to serve as a robust identity verification system. However, the PKI ecosystem driven by competitive business and geopolitical interests is under constant attack. Some of the vulnerabilities are technical and others are procedural. In the following, we discuss some of the prominent technical vulnerabilities and their mitigation approaches.

III. X.509 PKI: Concerns and Mitigations

The X.509 ecosystem plays a critical role in securing communications over the Internet. The majority of the communication is secured using the SSL/TLS protocol, which is intertwined with the HTTP protocol built on top of TCP/IP. Thus increasing the attack surface of the PKI ecosystem. In the following, we enumerate some of the major mitigation strategies devised to contain structural vulnerabilities of the X.509 ecosystem.

1) **DANE (DNS-based Authentication of Named Entities):** DANE allows a domain owner to specify which CA is allowed to issue certificates for a particular resource from the domain, which solves the problem of any CA being able to issue certificates for any domain [25], [26]. DANE needs the DNS records to be signed with DNSSEC for its security model to work. However, DNSSEC is not yet a widely deployed protocol.
2) **Certificate Pinning:** Pinning is the process of associating a host with their expected X509 certificate or public key [20]. Once a certificate or public key is known or seen for a host, the certificate or public key is associated or *pinned* to the host. If more than one certificate or public key is acceptable, then the program holds a *pinset* – the advertised identity must match one of the elements in the *pinset*.
3) **Perspectives:** Perspectives is a decentralized approach to securely identifying Internet servers. It automatically builds a database of server identities using lightweight probing by *network notaries* - servers located at multiple vantage points across the Internet. Each time a user connects to a secure website, Perspectives compares the site's certificate with network notary data and warns if there is a mismatch. Perspectives prevents *man-in-the-middle* (MitM) attacks and lets users use self-signed certificates and helps them trust that their connections really are secure.
4) **Convergence:** Convergence is an extension of Perspectives with additional guarantees from DNSSEC/DANE initiative. It is a method of using multi-path probing to establish a domain identity. Convergence clients verify a site's certificate by comparing it to the certificates obtained by trusted notaries that have accessed the target site via different network paths. By comparing the certificates, Convergence can probabilistically detect the presence of a MitM attack, with increasing confidence as the notary set's network path diversity increases. This procedure replaces traditional signature verification in multi-path probing systems; hence, even when a site's certificate was issued by a CA, Convergence need not trust that CA's public key in order to validate the certificate. Convergence offers the property of trust agility for the end user. Trust agility is comprised of two properties: first, the ability to re-evaluate trust decisions at any time without repercussion; second, an ability for each individual to select their own trust anchor.
5) **MECAI (Multiple Endorsing Certificate Authority Infrastructure):** MECAI doesn't introduce a new set of Authorities, it rather expects additional contributions from the existing CAs. CAs are expected to act as Web notaries, similar to what has been already proposed by other projects, such as the Perspectives Add-On for Mozilla Firefox, or as part of the Convergence project. A web notary shall be one or multiple servers that are run by a CA. A web notary is expected to make statements about facts that can be discovered on the web. A client connecting to a server will receive the server's certificate as part of the TLS handshake. A client could contact a notary, and ask the notary for the server certificate that can be obtained from the Notary's perspective, and send this information to the client. The client can compare

the perspectives, they should be identical. A MitM will need to use a different certificate that the MitM controls, in order to read and/or manipulate the data exchanged. If the connection between client and server is influenced by a MitM, then the certificates seen will be different based on the perspectives. If the information returned by the notary is different than the information seen by the client, then the client probably shouldn't trust the information presented by the server.

6) **Sovereign Keys:** The Sovereign Keys design allows clients and servers to use cryptographic protocols without having to depend on any third parties after the moment the server creates a Sovereign Key. Sovereign Keys are created by writing to a semi-centralized, verifiably append-only data structure. The main requirement for being able to do this is that the requesting party controls a CA-signed certificate for the relevant domain, or uses a DNSSEC-signed key to show that they control that domain. Master copies of the append-only data structure are kept on machines called "timeline servers". There is a small number, around 10-20, of these. The level of trust that must be placed in them is very low because the Sovereign Key protocol is able to cryptographically verify the important functions they perform. Sovereign Keys are preserved so long as at least one server has remained good.

IV. PKI as a Trust Management Protocol

Access control is a decision process where a subject is allowed to access a controlled object upon satisfying the access policy defined for that object. As the first step in this decision process, the subject needs to be correctly authenticated (identification-cum-verification). Upon authentication the subject is evaluated against the conditions specified in the access policy of the object; if all conditions are satisfied by the subject, access is granted. For example, a simple access control policy is to list the identifiers of subjects who are allowed to access the object. It can be implemented by listing the public keys of the subjects; similar to `authorized_keys` construct in SSH protocol. Authentication alone acts as an implied authorization. What if the identifiers of subjects are not known a priori? Typically, apart from OS-based access control mechanisms, web applications require more dynamic access control decisions that require attributes beyond identifiers; for example, time, role, exceptions, et al, and not all such attributes can be a priori known. Attribute certificates can be considered for this scenario but for a dynamic/volatile decision environment, there will be a large number of revocations of attribute certificates – overhead in decision making. As applications become complex and need to interact with subjects from different administrative domains, access control becomes challenging. In order to ease the management of resources with understandable access policies, approaches like delegation of authority, and placeholders like roles instead of actual identifiers are used. When a delegation of authority is allowed, the verifier is trusting the delegatee to ensure that intended subjects are being authorized. When a role is used to write an access policy, the role controller is expected to resolve the set of subjects accurately. These are the types of trust decisions made during the desing of an access control mechanism. PKI plays an important role in this trust management because it serves identity service at its core, which is the first step in doing access control. However, it falls short in dynamic environments where attributes other than identity play a role in decision-making. In the following we elaborate on how the architecture of a PKI limits/enhances the expressiveness of trust.

A. PKI Architecure Affects Expressiveness of Trust

Based on the position of the trust anchor in the trust relationship graph, the PKIs can be categorized into two types: centralized and decentralized. X.509 type of PKI is a centralized PKI since the root CA is the trust anchor and trust flows downwards from it towards the leaf nodes – the certificates. The end users in centralized PKI must trust its CA. The CAs may form a trust relationship among themselves but the validation of the certification path starts from the root. Therefore, it is also called as hierarchical trust model in which trust flows top-down and the end users do not have much say in how the trust relations are formed at higher levels. However, in decentralized PKIs like PGP, SPKI/SDSI [23], [24], there is no root authority in which the trust of the whole ecosystem is anchored. PGP is a bottom-up trust model where individuals form trust relationships with each other at their own discretion. PGP is an identity-providing PKI. Whereas, in the SPKI/SDSI type of PKI, apart from identity certificates there are authorization certificates that assign authorization attributes to roles (in SPKI they are known as local/extended names). Each public key in SPKI is allowed to define roles to which authorizations can be asserted through separate certificates and at the time of access control decision the role value is evaluated. Thus, public keys (subjects) can be added or removed from roles as required. This way the controller of the role definition controls identity relationships by maintaining group membership, thus eliminating the need for CRLs. Also, the authorization certificates can delegate authority to another public key – as an expression of trust, which simplifies access policy management. Since SPKI is authorization centric and the identity of a resource requestor is evaluated as per need [27], [28], the access control decisions are accurate and involve the participation of actual stakeholders. This bottom-up trust relationship formation is more expressive than the top-down option of X.509. However, X.509 is the most widely deployed identity provider but poor at communicating dynamic authorization attributes. In the following, we discuss a few of the alternative trust management protocols for decentralized systems.

B. Alternative Trust Management Protocols

In a decentralized system, it is difficult to ascertain whether an access request with an access policy for a given set of credentials. PolicyMaker [29] is a tool that accepts an access request, an access policy, and a set of credentials (certificates) to determine whether the request complies with the policy. It provides an abstraction level where security policies can be written without modifying the underlying trust management system like X.509 or PGP. *flexi*-ACL [2] is another approach based on SPKI/SDSI where security assertions can be not only certificates but any other pluggable authentication or authorization module outside the PKI ecosystem. It provides a language to specify access policies based on such security assertions. It also supports a special security assertion called a token, which is a cryptographic proof generated upon successful evaluation of an access policy elsewhere. However, both of these approaches cannot deterministically guarantee compliance of a set of credentials against a security policy. Because of the obvious characteristic of a distributed system – lack of a uniform replication of *state* across all the federated administrative domains. In the recent past, blockchains have emerged as a global *state* replication system. Due to their ability to transparently and verifiably provide proofs of correct execution of transactions with the help of *smart contracts*, are adjudged as modern trust management protocols [30]. However, though all of these expressive trust management protocols are adept at performing complex access control in a distributed environment, they still require an identity service that is global and widely available like the X.509 ecosystem.

C. Alternative Identity Management Protocols

Blockchain as a global state machine has paved way for engineering Internet-wide platforms where desperate organizations/individuals congregate, commit, and then abide by their commitments. This generic promise of blockchains has allowed this technology to graduate itself from privacy-preserving financial transaction networks to large-scale identity management platforms. Recent advancements in blockchain technology now allow every public key to have its own address, which is called a decentralized identifier (DID). There are several publicly available identity networks based on blockchain technology. i) Hyperledger Indy: A distributed ledger that provides tools, libraries, and reusable components for creating and using independent digital identities. ii) Sovrin: The Sovrin Network is the first public-permissioned blockchain designed as a global public utility exclusively to support self-sovereign identity and verifiable claims. iii) W3C DID & VC: A verifiable credential can represent all of the same information that a physical credential represents. Going beyond providing identity-as-a-service, these networks are also capable of providing URIs for resources in need of dynamic, contextual access control.

V. Conclusions

The Internet continues to grow in scale and complexity, and the requirement for a credible identity service and a global trust management framework is underscored. X.509 PKI ecosystem has been serving its core functionality of efficient authentication. However, functionally, it is falling short while realizing an Internet-wide expressive trust management framework. Though new approaches are emerging as alternative trust management frameworks, they are not yet mainstream and have issues with shared ledgers. Today, we have a desperate set of technologies that are proficient in their own right but a lack of comprehensive trust management framework exists. The question for which the community needs a consensus is: Should there be a singular technology to manage Trust over the Internet?

Acknowledgment

This work is carried out as a part of the project RD/0120-NCSC001-001 "AI Powered Security Operation Product Suite for National Critical Information Infrastructure", funded by the NSCS, Government of India.

References

[1] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] V. Patil and R. K. Shyamasundar, "Trust management for e-transactions," *Sadhana*, vol. 30, no. 2, pp. 141–158, 2005.

[3] D. Kumar, Z. Wang, M. Hyder, J. Dickinson, G. Beck, D. Adrian, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey, "Tracking Certificate Misissuance in the Wild," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 785–798.

[4] S. B. Roosa and S. Schultze, "Trust Darknet: Control and Compromise in the Internet's Certificate Authority Model," *IEEE Internet Computing*, vol. 17, no. 3, pp. 18–25, 2013.

[5] Z. Dong, K. Kane, and L. J. Camp, "Detection of Rogue Certificates from Trusted Certificate Authorities Using Deep Neural Networks," *ACM Trans. Priv. Secur.*, vol. 19, no. 2, 2016.

[6] H. Birge-Lee, Y. Sun, A. Edmundson, J. Rexford, and P. Mittal, "Bamboozling Certificate Authorities with BGP," in *27th USENIX Security Symposium*. USENIX Association, 2018, pp. 833–849.

[7] T. Dai, H. Shulman, and M. Waidner, "Off-Path Attacks Against PKI," in *CCS '18: Proceedings of the Conference on Computer and Communications Security*. ACM, 2018, pp. 2213–2215.

[8] A. Delignat-Lavaud, M. Abadi, A. Birrell, I. Mironov, T. Wobber, and Y. Xie, "Web PKI: Closing the Gap between Guidelines and Practices," in *21st Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2014.

[9] C. Ellison and B. Schneier, "Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure," *Computer Security Journal*, vol. 16, no. 1, pp. 1–7, 2000.

[10] R. Forno and W. Feinbloom, "Inside Risks: PKI: A Question of Trust and Value," *Communications of the ACM*, vol. 44, no. 6, p. 120, 2001.

[11] P. Gutmann, "PKI: It's not Dead, Just Resting," *Computer*, vol. 35, no. 8, pp. 41–49, 2002.

[12] A. Slagell, R. Bonilla, and W. Yurcik, "A Survey of PKI Components and Scalability Issues," in *2006 IEEE International Performance Computing and Communications Conference*, 2006, pp. 10 pp.–484.

[13] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, May 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5280

[14] C. Adams, S. Farrell, T. Kause, and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)," RFC 4210, Sep. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4210

[15] R. Housley, W. Polk, W. Ford, and D. Solo, "RFC3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," 2002.

[16] S. Khan, Z. Zhang, L. Zhu, M. Li, Q. G. Khan Safi, X. Chen, and S. Zeadally, "Accountable and Transparent TLS Certificate Management: An Alternate Public-Key Infrastructure with Verifiable Trusted Parties," *Sec. and Commun. Netw.*, vol. 2018, jan 2018.

[17] Z. Wang, J. Lin, Q. Cai, Q. Wang, D. Zha, and J. Jing, "Blockchain-Based Certificate Transparency and Revocation Transparency," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 681–697, 2022.

[18] S. Eskandarian, E. Messeri, J. Bonneau, and D. Boneh, "Certificate Transparency with Privacy," *CoRR*, vol. abs/1703.02209, 2017.

[19] D. Kales, O. Omolola, and S. Ramacher, "Revisiting User Privacy for Certificate Transparency," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2019, pp. 432–447.

[20] C. Evans, C. Palmer, and R. Sleevi, "Public Key Pinning Extension for HTTP," RFC 7469, Apr. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7469

[21] S. Farrell, R. Housley, and S. Turner, "An Internet Attribute Certificate Profile for Authorization," RFC 5755, Jan. 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5755

[22] V. Patil, P. Gasti, L. Mancini, and G. Chiola, "Resource Management with X.509 Inter-domain Authorization Certificates (InterAC)," in *EuroPKI '10: Public Key Infrastructures, Services and Applications*, vol. 6391. Springer, 2010, pp. 34–50.

[23] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI Certificate Theory," Internet RFC 2693, 1999, Internet RFC 2693.

[24] J.-E. Elien, "Certificate Discovery Using SPKI/SDSI 2.0 Certificates," Masters Thesis, MIT LCS, Tech. Rep., 1998.

[25] P. Hoffman and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA," RFC 6698, Aug. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6698

[26] P. Hallam-Baker and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record," RFC 6844, Jan. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6844

[27] N. Li, "Local names in SPKI/SDSI," in *CSFW-13: Proceedings 13th IEEE Computer Security Foundations Workshop*, 2000, pp. 2–15.

[28] N. V. N. Kumar and R. K. Shyamasundar, "Specification and Realization of Access Control in SPKI/SDSI," in *ICISS '06: Proceedings of International Conference on Information Systems Security*, vol. 4332. Springer, 2006, pp. 177–193.

[29] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management," in *SP '96: Proceedings of the Symposium on Security and Privacy*. IEEE Computer Society, 1996, p. 164.

[30] R. K. Shyamasundar and V. T. Patil, "Blockchain: The revolution in trust management," in *Proceedings of the Indian National Science Academy*, vol. 84 (2). Springer, 2018, pp. 385–407.

[31] C. Ellison, "Establishing Identity Without Certification Authorities," in *6th USENIX Security Symposium*. USENIX Association, 1996.

[32] P. Gutmann, "PKI Design for the Real World," in *NSPW '06: Proceedings of the Workshop on New Security Paradigms*. ACM, 2006, pp. 109–116.

[33] R. Perlman and C. Kaufman, "User-Centric PKI," in *IDtrust '08: Proceedings of the 7th Symposium on Identity and Trust on the Internet*. ACM, 2008, pp. 59–71.

[34] M. Brandt, H. Shulman, and M. Waidner, "Evaluating Resilience of Domains in PKI," in *CCS '21: Proceedings of the Conference on Computer and Communications Security*. ACM, 2021, pp. 2444–2446.

[35] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of Trust and Distrust," in *WWW '04: Proceedings of the 13th International Conference on World Wide Web*. ACM, 2004, pp. 403–412.

[36] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS Certificate Ecosystem," in *IMC '13: Proceedings of the Conference on Internet Measurement*. ACM, 2013, pp. 291–304.

[37] C. Adams, M. Burmester, Y. Desmedt, M. Reiter, and P. Zimmermann, "Which PKI (Public Key Infrastructure) is the Right One? (Panel Session)," in *CCS '00: Proceedings of the 7th ACM Conference on Computer and Communications Security*. ACM, 2000, pp. 98–101.

[38] M. Abadi, A. Birrell, I. Mironov, T. Wobber, and Y. Xie, "Global Authentication in an Untrustworthy World," in *14th Workshop on Hot Topics in Operating Systems (HotOS XIV)*. USENIX Association, 2013.

[39] C. Brubaker, S. Jana, B. Ray, S. Khurshid, and V. Shmatikov, "Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations," in *2014 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2014, pp. 114–129.

[40] A. Bates, J. Pletcher, T. Nichols, B. Hollembaek, and K. R. Butler, "Forced Perspectives: Evaluating an SSL Trust Enhancement at Scale," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. ACM, 2014, pp. 503–510.