IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 April 2024, accepted 24 April 2024, date of publication 29 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394657

RESEARCH ARTICLE

A Semi-Decentralized PKI Based on Blockchain With a Stake-Based Reward-Punishment Mechanism

ERHAN TURAN[1], SEVIL SEN[1], AND TAMER ERGUN[2]
[1]Department of Computer Engineering, Hacettepe University, 06800 Ankara, Turkey
[2]Imza.IO, Bostanci, 34744 Istanbul, Turkiye

Corresponding author: Erhan TURAN (erhan.turan@hacettepe.edu.tr)

ABSTRACT The conventional Public Key Infrastructure (PKI) has long been plagued by security issues stemming from its centralized and non-transparent design. In recent years, blockchain-based PKI architectures have emerged as promising solutions to overcome such issues. However, existing research has predominantly focused on SSL certificates, overlooking other certificate types used for purposes such as facilitating electronic signatures/seals, code signing, and S/MIME- all reliant on the foundational PKI infrastructure. In this study, we present a novel blockchain-based PKI architecture designed to accommodate diverse certificate types. Combining the principles of the Web of Trust with a centralized model, our SemiDec-PKI establishes a resilient, distributed infrastructure. This unique synergy minimizes reliance on a single central authority, mitigating the vulnerabilities associated with traditional PKI systems' single points of failure. With a cross-check mechanism and collective consensus of trusted entities, SemiDec-PKI provides higher fault tolerance, preventing disruptions from certificate misissuance or compromised certificate authorities. Furthermore, it introduces a stake-based reward-punishment mechanism which incentives honest behavior and penalizes malicious actions, serving as a potent deterrent against impersonation attacks.

INDEX TERMS Blockchain, certificate transparency, Ethereum, PKI, SSL, smart contract.

I. INTRODUCTION

In today's interconnected and digitalized world, ensuring secure communication and protecting sensitive information has become a critical priority. Public Key Infrastructure (PKI) serves as a fundamental framework that guarantees the integrity, confidentiality, and authenticity of electronic communications and transactions. By harnessing the power of asymmetric encryption and digital certificates, PKI establishes an infrastructure for building trust, verifying identities, and securing sensitive data across diverse networks. Certificate Authorities (CAs) play a pivotal role in the PKI ecosystem by issuing and managing digital certificates. These trusted third-party entities verify the identity of certificate subjects and digitally sign their certificates, vouching for their authenticity.

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

While CAs play a vital role in the system, they expose PKI to potential vulnerabilities and incidents where faulty certificates might be issued. A number of high-profile incidents have highlighted the risks associated with faulty certificate issuance [1]. For example, Comodo and DigiNotar were compromised, and attackers managed to issue fraudulent certificates for popular websites, including Google, Yahoo, and Skype [2], [3]. Symantec mistakenly issued certificates that included a domain name, violating CA industry policies and procedures in 2015 [4]. Furthermore, CAs (e.g., Lets Encrypt) are taken down by research in practice to show their weak security [1], [5], [6]. A rigorous vetting process and adherence to industry standards by CAs are crucial to preventing such unauthorized certificate issuances [6].

As shown in the incidents above, CAs constitute a *Single Point of Failure (SPoF)* in the certificate issuance mechanism. Therefore, CAs should provide guarantee regarding the

accuracy of the information included in the certificate or the integrity of their infrastructure. CAs offers financial compensation or remedies in situations where the CA's negligence or failure to meet industry standards has caused harm to a subject/entity. As part of the current PKI model, *punishment issues* are handled by insurance companies, and entities must prove that they have been victimized in court. Another issue is the *management of trust lists*. In PKI, a certificate chain is a list of certificates that usually starts with an end-entity certificate followed by one or more CA certificates. Root CAs are self-signed and pre-installed in the OSs, browsers, and applications. Besides the trust lists of SSL certificates, applications could have their own trust lists of CAs. For example, Adobe has a trust list for electronic signature [7], Java has a trust list for code signing certificates. Since CA needs to contact every application, browser, and OS, so it is hard to manage trust lists for each software.

In PKI, CAs have a crucial role in tasks like identity validation and application assessment. However, their involvement also introduces the risk of human errors. To enhance security and mitigate potential risks, it becomes essential to implement a cross-check mechanism. This process should involve personnel from diverse authorities to maximize its effectiveness. Currently, the conventional certificate life-cycle lacks this protective measure. Hence, impersonation attacks that impersonate legitimate entities and fraudulently obtain certificates could occur in this registration step. In such attacks, attackers may provide false documentation to the Certificate Authority (CA) in order to prove the ownership of a domain. Therefore, trust and security of the infrastructure could be compromised due to these *certificate registration issues*. The last issue called *the revocation monopoly issues* is related to certificate revocation, which is the process of declaring a previously issued digital certificate as invalid or no longer trustworthy before its expiration date. Certificate owners or CAs can initiate the revocation process for the certificate to the responsible CA. In an emergency, it can be problematic if the CA cannot be reached immediately, such as during non-working hours.

In the literature, approaches [8], [9], [10], [11], [12], [13], [14], [15] based on blockchain are proposed to solve these issues. However they particularly focus only on SSL certificates [9], [10], [11], [12], [13], [15], or they are based on identity management [8], [14]. While a PKI system should be designed to cover all certificate types, these studies do not fully provide that. Moreover, they do not encompass all the inherent challenges in the current system. For example, they usually do not include a cross-check mechanism [8], [9], [10], [11], [13], [14], or punishment mechanism [8], [9], [10], [11], [12], [13], [15], and do not prevent the single point of failure [8], [9], [10], [11], [12], [13], [14].

In order to address these issues, we propose a new semi-decentralized PKI architecture called SemiDec-PKI that supports any type of digital signature. SemiDec-PKI provides all services required for the management of certificates such as revocation, validation, dissemination, monitoring, and auditing. The proposed approach is based on blockchain due to its suitable characteristics for the problem such as decentralization, immutability, transparency and security. Our PKI architecture is built upon Smart Contracts (SCs) that effectively manages certificate issuance, revocation, validation, and fraud mechanisms. For instance, the smart contract code meticulously verifies all certificate fields and ensuring the validity of the certificate structure during the certificate issuance step, and all parties could check the status of certificates via SCs in the validation step.

The proposed approach constructs a hierarchical structure comprising a Trust List (TL), which is under the management and governance of Supervisory Bodies (SB). SBs are predefined in smart contracts (SC) and eases the *management of trust lists*. SemiDec-PKI is based on a voting scheme that combines the Web of Trust with a centralized approach in order to prevent *SPoF*. The voters are selected by using a stake reward-punishment mechanism. While SBs vote for CA certificates, CAs vote for end-user certificates. The voting system also increases robustness of the system against impersonation attacks resulting from *certification registration issues*. Moreover, the proposed approach enables users to revoke their certificates independently, eliminating the necessity for a centralized revocation authority. This empowers users with increased autonomy and effectively addresses *revocation monopoly issues*. By using SCs, we increase the security controls in each step in certificate life cycle. Last but not least, we propose a robust commercial model based on a stake reward and punishment mechanism which incentivizes certain behaviors while penalizing undesirable actions. CAs and end-users participate by locking up a specific amount of cryptocurrency tokens as collateral, referred to as their stake, which serves as assurance and is subsequently distributed among voters. Potential incidents and fraud complaints are promptly addressed by penalizing the responsible CAs through the loss of their tokens, allowing for a resolution of *punishment issues* without the need for lengthy court procedures.

To sum up, SemiDec-PKI approach presents a decentralized PKI architecture that effectively addresses the challenges inherent in the traditional PKI system. While various other proposals based on blockchain technology [8], [9], [10], [11], [12], [13], [14], [15] and log-based approaches [16], [17], [18], [19], [20], [21] are available in the literature that aim to improve certificate life-cycle management for CAs and mitigate known security threats, SemiDec-PKI distinguishes itself from other studies by providing a comprehensive solution that extends beyond SSL certificates to encompass any type of certificates. SSL certificates, due to their online and publicly accessible nature, allow for the monitoring of the number of active certificates in use. However, the

statistics about qualified certificates for e-signature or code signing, which plays a critical role in various applications, remains relatively scarce a global scale. Only a limited number of countries, such as Turkey, with its 80 million-strong population, have conducted research to compile statistics on these certificates. For example, in Turkey, there are approximately 6.9 million qualified digital certificates and 900,000 mobile signature certificates dedicated to e-signatures [22], [23]. This underscores the urgent need for secure and efficient PKI systems that can extend their benefits to various certificate types. Moreover, while certificates conform to the X509 standard and are encoded with ASN.1 in the traditional architecture and the other proposals, SemiDec-PKI redefines the certificate structure and ensures the integrity and validity of certificates by using SCs and hence prevents the issuance of malformed certificates. Of particular significance, the Semi-Dec PKI stands out by effectively resolving the persistent problem of single points of failure with its pioneering voting scheme. This approach provides robust protection against vulnerabilities linked to certificate issuance, thus fortifying the PKI system's resilience and security.

The remainder of this paper is structured as follows. Section II provides background information on Public Key Infrastructure (PKI), blockchain technology, and Smart Contracts (SCs). Section III introduces the proposed approach, SemiDec-PKI in details. Section IV analyzes the performance of SemiDec-PKI, and discusses the backward compatibility of the proposed architecture and possible threats against it. Then, we present a comprehensive overview of related studies in the literature in Section V. Section VI comprehensively evaluates SemiDec-PKI against existing PKI and blockchain-based solutions in the literature, presenting the detailed comparison in a tabular format. Lastly, Section VII concludes the study by summarizing the contributions of SemiDec-PKI and discusses potential directions for future studies.

II. BACKGROUND

A. AN OVERVIEW OF PKI

In PKI, digital certificates are essential for linking a public key to the entity (such as an individual or organization) that owns it as defined in [24] and [25]. These certificates are verified using digital signatures from Certificate Authorities (CAs). A digital certificate contains essential components, including the public key itself, personal information about the entity, and additional data required for validating its authenticity. CAs are responsible for signing end-user certificates. In many cases, the CA does not directly issue certificates to end-entities but uses one or more intermediate CAs. At the top of the chain of trust is the trusted root certificate. The trust list is a pre-configured set of certificates that are inherently trusted by a software application, web browser, operating system, or any other system that requires PKI functionality. They act as the starting point of trust in the PKI hierarchy, as they are self-signed certificates which are called as root certificates, representing the highest level of trust in the system. There are various types of certificates used for different purposes in (PKI) such as SSL certificates for securing web communications, electronic seal certificates for document authenticity, code-signing certificates for verifying software authenticity and electronic signature certificates. Moreover, there are emerging application domains for certification such as Multi-Dimensional Certification [26] and Certification of Internet of Things Devices [27]. Each type of certificate may have its own trust list to establish trust in the corresponding CAs. Because of the diverse range of certificates and the need to maintain their validity and trustworthiness over time, managing trust lists can be challenging.

Certificate revocation is the act of invalidating a certificate before its expiration date. A certificate should be revoked immediately when its private key is in danger of being compromised. It must also be revoked when the certificate owner lost the keys or the certificate is no longer operational. CAs are responsible for indicating the revocation status of the certificates that they issue. Revocation status information may be provided using the Online Certificate Status Protocol [28], certificate revocation lists (CRLs) [25], or other mechanisms. OCSP is a protocol used to check the real-time revocation status of a digital certificate directly from the Certificate Authority (CA) that issued the certificate. CRLs are time-stamped lists published by Certificate Authorities (CAs) that contain the serial numbers of certificates that have been revoked before their expiration dates. These lists are periodically updated, and relying parties can download the CRL from the CA's distribution point. However, a CA may choose to delegate the responsibility of issuing CRLs to a different entity known as a CRL Distribution Point (CDP) or CRL Issuer.

B. BLOCKCHAIN

Blockchain is an immutable, distributed ledger of transactions whereby transactions reside in so-called blocks, and ledgers are distributed across peer-to-peer networks. These networks commonly depend on Merkle trees. Each leaf node in the Merkle tree represents a hash of transactional data, ensuring the integrity and security of the data in the block and non-leaf nodes are labeled with the hash of all of their child nodes. Hence, this tree structure allows participants to create unique, concise, and quickly verifiable evidence. The Merkle trees grow logarithmically in relation to the number of its leaves. Each blocks consist of block headers, previous header hash, merkle root and every leaf node is a hash of transactional data as shown in Fig. 1.

New blocks, which include new transactions, are appended to the blockchain by a stochastic process called mining. The mining process commonly depends on Proof of Work (PoW) or Proof of Stake (PoS) methods. The PoW method is a consensus mechanism that requires participants to

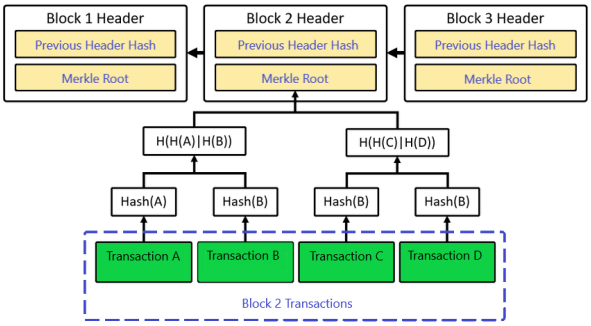

FIGURE 1. Merkle tree.

make an effort to resolve a random puzzle to ensure that nobody deceives the system. However, PoW methods require a significant amount of computational power, leading to a substantial energy consumption. As the cost of this mechanism increases proportionally with the number of transactions, the PoS system offers an alternative. In PoS, a participant is chosen to add the latest transaction stack to the blockchain and they receive a reward in the form of cryptocurrency. The selection of participants is based on a probability, with participants who hold a higher stake having an increased chance of being selected for this role [29].

A decentralized blockchain network consists of nodes that represent mining entities such as servers or computers. They are responsible for verifying and maintaining the public ledger of transactions on a blockchain network. Typically, there are three kinds of nodes in such networks, each with varying levels of responsibility: light nodes, full nodes, and archive nodes. Light nodes serve as a lightweight type of computing device or software implementation used to support a blockchain network.

In PoW systems, miners can function as either light or full nodes; whereas in PoS systems, staking wallets are the light nodes. A light node maintains the header chain, requests the remaining data, and verifies data validity against the state roots in the block headers. A full node has the responsibilities of a light node, and additionally, it stores the entire blockchain data. Archive nodes store everything kept in the full node and build a historical state archive [30].

1) STAKE REWARD-PUNISHMENT MECHANISM

A stake reward-punishment mechanism is an important concept in blockchain that involves both incentivizing and disincentivizing behaviours to ensure the protection and integrity of the network. Hence participants are encouraged to act honestly and securely within the network. In this regard, stakeholders must ''stake'' or deposit a certain amount of cryptocurrency (tokens) as a pledge to participate in specific activities or protocols. The combination of stake reward and punishment ensures that the blockchain ecosystem remains resilient and self-sufficient, with participants motivated to act in a manner that benefits the network and punished for engaging in harmful or malicious actions.

Stake reward is a positive reinforcement mechanism where participants who actively contribute to the blockchain network, satisfy their obligations, and make beneficial decisions are rewarded with additional tokens or cryptocurrency. These rewards serve as an incentive to maintain the integrity and security of the network, as participants are motivated to act in the system's best interest to earn rewards. On the other hand, stake punishment is an obstacle against malicious or dishonest behavior. If a participant fails to adhere to the network's rules, attempts fraudulent activities, or violates consensus protocols, some or all of their staked tokens may be forfeited or ''slashed''. This serves as a disincentive for bad actors and helps maintain the overall security and reliability of the blockchain network.

2) SMART CONTRACTS (SC)

Smart contracts are computer programs that can reliably and consistently perform transactions and agreements between anonymous parties [31]. They can trigger subsequent actions in a workflow when certain conditions are met without a central authority, legal system, or external enforcement mechanism. The most popular blockchain utilizing SCs is Ethereum. A SC can be deployed on a blockchain with a gas fee determined due to its size. The gas fee is a fee paid to miners to remunerate the computational power required to process and validate transactions. The immutability of blockchain secures smart contracts from tampering so that, once a SC is deployed, it cannot be modified. Users interact with a SC via transactions with the compiled code of a smart contract through the contract's address. Such transactions might change the state of the contract and receive/send coins from/to another account. Also, a SC can invoke other SCs.

Ethereum tokens are digital assets created, managed, and exchanged on the Ethereum blockchain. One of the most known Ethereum tokens is named ERC-20 [32], published as a technical standard for smart contracts on the Ethereum blockchain. In our design, we created a stake-based reward-punishment mechanism based on ERC-20.

III. SEMIDEC-PKI

This section first outlines the challenges in traditional PKI and the solutions proposed by Semidec-PKI to resolve them. Following that, the design of Semidec-PKI is presented in detail.

A. ISSUES

A well-managed Public Key Infrastructure (PKI) should adhere to various technical and legal documents [33]. The trustworthiness of a Certificate Authority (CA) is determined by its compliance with these requirements. However, the lack of a standardized, globally accepted method for managing certificate issuers can lead to specific challenges and concerns. The current issues and corresponding solutions of SemiDec-PKI are summarized as follows:

1. Management of trust lists: The European Union (EU) enacted a regulation for the management of trust lists

and, the CA/Browser Forum (CA/B Forum) [34] established industry standards and guidelines for the issuance and management of digital certificates in the SSL framework [35], [36]. Some browsers or operating system vendors run certificate inclusion programs to decide on trust lists under the supervision of CA/Browser Forum [37], hence there is no common global trust list for CAs issuing certificates. It is therefore hard to determine which trust list is valid for which certificate type from the application's point of view.

Solution: Semidec-PKI contains a chain of trust within itself. With the trust chain it contains and the ability to verify certificates via smart contracts, applications do not need to manage a separate trusted list. The system allows for expansion by welcoming participants from new countries into the SBs, and it provides a voting process where existing Supervisor Bodies can vote for new SBs.

2. Single Point of Failure (SPoF): Even if applications handle the management of trust lists, CAs which have absolute trust in the infrastructure may still make mistakes, and this may result in the *Single Point of Failures*. Attackers may still issue end-user certificates and exploit the system by compromising CAs [19].

Solution: eIDAS (electronic IDentification, Authentication and trust Services) [38] defines procedures from defining supervisory bodies to the issuance of end-user certificates. A supervisory body audits CAs through accredited conformity assessment bodies (CABs) and reviews related audit reports in order to decide if the CA is eligible to issue certificates. Therefore, one way to manage a worldwide trust list of CAs would be to specify SBs, CABs, and related technical and legislative requirements for CAs to be designated as trusted. The current study, SemiDec-PKI, proposes a system that aligns with the regulatory framework provided by eIDAS while harmonizing centralized and web of trust-based approaches. It achieves this by employing blockchain technology to deliver ease of access and transparent management of a globally trusted list. The system incorporates a voting mechanism to eliminate *Single Point of Failure* in scenarios where a Certificate Authority (CA) may be compromised or deceived.

3. Certificate registration: In conventional operation, certificates are issued only after registration authorities assess end-user application forms. Once approved and issued, an end-user certificate remains valid until revoked or expired. Registration authorities handle identity validation and application assessment, making them susceptible to human errors. Therefore, cross-checks are critical protection mechanisms, especially when involving checkers from different authorities to enhance effectiveness. However, the conventional certificate lifecycle does not currently employ such a mechanism.

Solution: SemiDec-PKI includes more than one CA in the control mechanism and prevents fraud documents and impersonation attacks related to domain ownership by providing cross check with voting dynamics.

4. Revocation monopoly: The lifecycle of a certificate spans from its creation to the end of its validity. In the conventional PKI architecture, CAs are responsible for services in a certificate lifecycle, namely registering, verifying, creating, and validating certificates. CAs can also revoke certificates in the case of an issue being reported. Although certificate owners and CAs can initiate the revocation process for the certificate, the revocation process itself can only be conducted by CAs. Therefore, if CAs are not reachable outside of normal working hours, this may cause problems in case of emergency.

Solution: SemiDec-PKI resolves this issue by actively involving multiple CAs in the certificate revocation process. Voters in this design consist of Supervisor Bodies and CAs. The revocation of a CA is dependent on the votes of Supervisor Bodies.

5. Punishment: In the context of traditional PKI, Certificate Authorities (CA) engaged in fraudulent activities or failed to meet security standards can face a wide range of consequences. One notable consequence is the potential removal of the CA's root certificate from the trust lists maintained by browsers and operating systems. Additionally, legal actions might ensue against the errant CA, leading to fines or penalties for various infractions, including negligence, breach of contract, and involvement in fraudulent activities. However, it's important to note that these punitive measures occur external to the PKI framework and fall under the jurisdiction of legal authorities. Despite their effectiveness, these actions are not inherently integrated into the PKI system itself. As PKI develops, there is a growing desire to strengthen its capacity for identifying such breaches within its framework. This would lead to quicker detection of misconduct and more effective resolution for those impacted.

Solution: SemiDec-PKI also incorporates a punishment mechanism. In the event of an attempted manipulation by an attacker, SemiDec-PKI responds by imposing appropriate penalties, as determined by the voting results. The attacker faces the risk of losing ERC tokens due to their actions, serving as a powerful deterrent against malicious behavior.

6. Monitoring and logging: Although it is possible to track SSL certificates of end-users created by CAs, this does not apply to other types of certificates. Moreover, both in traditional PKI and the Certificate Transparency Project, *split-world attacks* [10] are an issue resulting from the difference between the validation process of SSL/TLS certificates by web browsers and their logging procedures on servers. In this attack, attackers acquire fraudulent certificates from a Certificate Authority (CA) and manipulate the logging mechanism to either incompletely log or entirely omit the fraudulent certificates from Certificate Transparency (CT) logs.

Solution: SemiDec-PKI incorporates multiple certificate authorities (CAs) in its control mechanism to prevent fraudulent documents and impersonation attacks concerning

TABLE 1. Issues and corresponding solutions proposed by SemiDec-PKI.

Issues	Corresponding Solutions in SemiDec-PKI
Management of trust lists	trust chain and smart contract-based certificate validation.
Single Point of Failure (SPoF)	voting scheme.
Certificate registration	cross check via multiple CAs'.
Revocation monopoly	revocation that can be done by users in smart contract.
Punishment	disincentive with ERC Tokens.
Monitoring and logging	audit and fraud mechanisms.
Split-world attacks	via blockchain.

domain ownership through cross-checks with voting dynamics. It employs a hierarchical structure for auditing and cross-checking the entire PKI ecosystem. To eliminate *Split-world attacks* SemiDec-PKI uses a blockchain-based approach without requiring more than one log server.

All the issues and the corresponding solution proposed by SemiDec-PKI are summarized in Table 1.

B. DESIGN OF SEMIDEC-PKI

SemiDec-PKI introduces a novel approach by combining centralized and Web-of-trust paradigms, utilizing smart contracts to establish a robust Public Key Infrastructure. SemiDec-PKI, redefines the following mechanisms based on blockchain: (1) certificate issuance, (2) certificate revocation, and (3) audit and fraud reporting. The first mechanism defines the steps of certificate issuance for supervisory bodies, CAs, and for end-users. Certificates are validated by supervisory bodies and CAs using a smart contract-based operation reinforced with a stake-based reward-punishment-based mechanism. In the second mechanism, the proposed SemiDec-PKI solution defines a revocation mechanism that can not only be triggered by issuers as expected, but also by certificate owners. Lastly, the SemiDec-PKI system is monitored by auditors who are willing participants seeking rewards, act as full nodes voluntarily, and continuously verify system activity. Whenever auditors identify an issue, they promptly report it and receive ERC-20 tokens for their efforts.

In SemiDec-PKI, a smart contract is defined with the state variables listed in Table 2. The requirements of these variables are denoted as "sh" for shall, "c" for conditional, and choice for "ch." Considering the size of each variable and assuming String is 32 bytes, the size of a certificate is approximately 256 bytes, which is half of the average X509 certificate size (512 bytes) given in CertLedger [7]. All mechanisms utilize these variables, as given in detail below.

1) CERTIFICATE ISSUANCE

An authoritative Certificate Authority (CA) is responsible for issuing certificates containing a public key and the identity of the owner. Certificate issuance signifies the CA's validation that the public key within the certificate corresponds to the entity, whether an individual, organization, server, or other entity, as specified in the certificate. In the context of traditional PKI, the issuance of inaccurate certificates can arise due to errors during the verification and certificate generation processes. These vulnerabilities expose the system to potential attacks. If attackers succeed in acquiring the CA's authority keys, they can generate additional certificates, undermining the system's security. The reliance on a single CA for generating certificates also brings about risks of Single Points of Failure (SPoF), as the entire system's reliability depends on the trust vested in that CA. In order to address these challenges, the implementation of a decentralized monitoring and auditing system becomes imperative. This system should ensure that even if an attacker gains control over a supervisory body or a CA, the overall system's availability remains unaffected. Additionally, it should prevent attackers from issuing unauthorized certificates to end-users or CAs.

The proposed approach, SemiDec-PKI presents a viable approach to meet these requirements through a voting scheme based on a stake-based reward-punishment mechanism. By incentivizing stakeholders to actively participate in the voting process and penalizing malicious actions, the system ensures a collaborative and secure environment. It significantly mitigates the risks associated with SPoF, and with participant of more CAs reduce certificate registration issues.

In the proposed approach, Certificates Authorities (CAs) and supervisory bodies (SBs) serve as authorized issuers during the certificate issuance phase. When a CA is issued, it can initiate issuing end-user certificates. Initially, a smart contract (SC) defines the set of supervisory bodies, who have the authority to issue a new CA or supervisory certificates. Certificate issuance operates on a voting mechanism, where both SBs and CAs play pivotal roles. SBs partake in voting to suggest the inclusion of new SBs or the issuance of new CAs. CAs cast their votes to endorse the issuance of end-user certificates. When the number of votes surpasses a predetermined threshold, the proposed addition or issuance moves forward.

The voting threshold is of utmost significance within this process, as it establishes the minimum number of votes required to attain consensus among stakeholders and validate the action. Each voting transaction accrues gas fees, and conducting numerous voting sessions might result in reduced overall cost-effectiveness and system efficiency. Therefore, the threshold should consider the trade-offs between augmenting security through increased cross-checks and maintaining the practicality and effectiveness of the

TABLE 2. SemiDec-PKI variables and requirements.

State Variable	Explantion	Type	Req.
certificateId	serial number of the certificate	uint64	sh
includeTransactionKey	transaction key inclusion status	bool	c
subject	identity information	string	sh
certificateType	type of the certificate	uint8	sh
publicKey	ECC256 compressed public key	bytes32+1	ch
issuerCertId	certificate Id of the issuer certificate	uint64	sh
expirationDate	expiration date of the certificate	uint32	sh
ownerAddress	Ethereum address of the owner	bytes20	ch
issuerAddress	Ethereum address of the issuer	bytes20	sh
audit	IPFS address of audit/registration documents	bytes32	c
cryptographicSignature	cryptographic signature of the certificate	2× bytes32	c
positiveVotersCount	stands for a trust measure	uint8	sh
negativeVotersCount	stands for a trust measure	uint8	sh
positiveVoters	addresses of positive voters	bytes32	sh
negativeVoters	addresses of negative voters	bytes32	sh
revocationStatus	status of the certificate	bool	sh
waitingTime	penalty for improper voting	uint32	c
X509Certificate	IPFS address of X509	bytes32	c

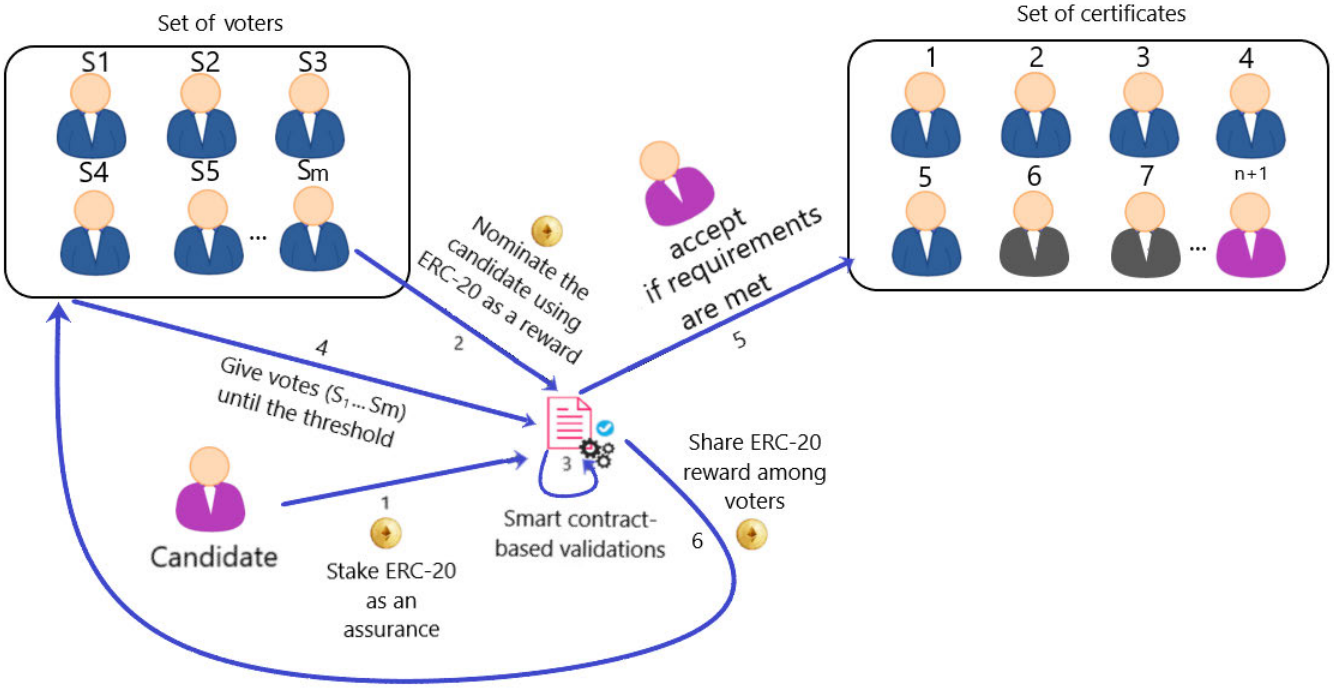

FIGURE 2. Certificate issuance mechanism.

system. Please note that the threshold value in smart contracts can be modified through majority consensus among relevant voters.

The voting mechanism within SemiDec-PKI integrates a stake-based reward-punishment system to uphold the network's integrity and security. All voters, including supervisory bodies (SBs), Certificate Authorities (CAs), and end-users, are involved in the voting process by staking tokens. Hence the proposed approach encourages honest and responsible participation, as individuals have a vested interest in the stability of the system. If a voter or end-user commits any malicious or erroneous act, the consensus mechanism enforces punishments through agreement of network participants.

The certificate issuance steps within the SemiDec-PKI system are depicted in Fig. 2 and can be summarized as follows:

(1) Where a candidate is an issuer, it needs to stake ERC-20 tokens as a blocked safety deposit for the purpose of assurance. (2) An issuer sends a transaction using ERC-20 tokens as a reward in order to add a new certificate candidate, which in turn triggers the smart contract-based validation. (3) To start the voting process, the following SC-based validation requirements need to be met in the SC as shown

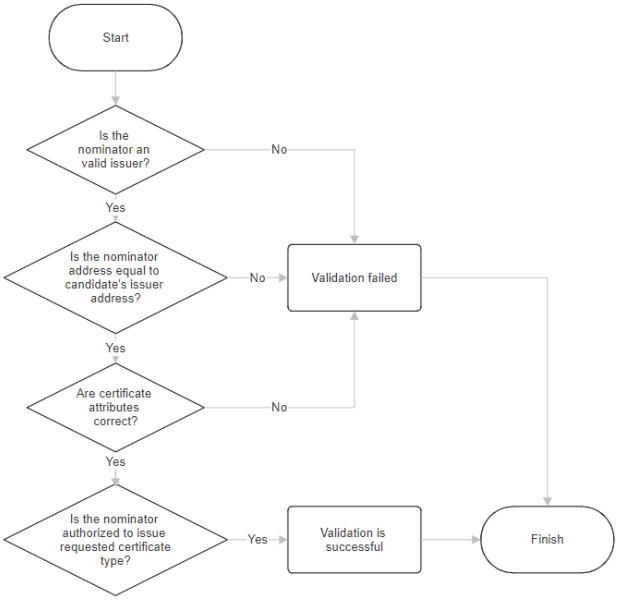

FIGURE 3. SC-based validation steps on certificate issuance.

FIGURE 4. Certificate revocation mechanism.

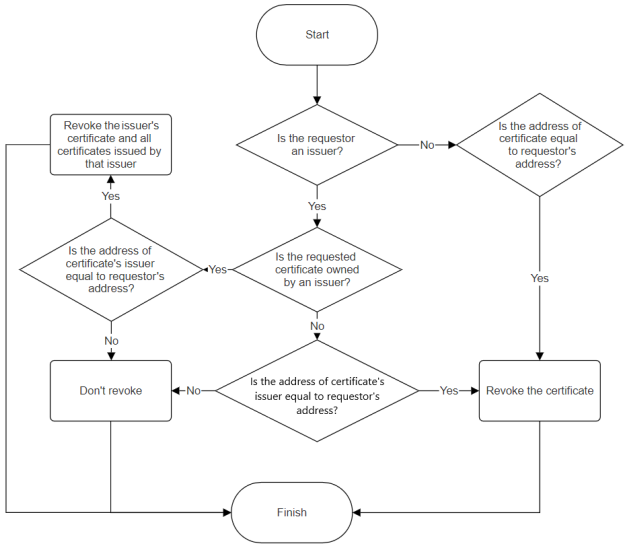

FIGURE 5. SC-based validation steps on certificate revocation.

in Fig. 3. If these requirements are not met, the SC-based validation fails, and the token is subsequently refunded. However, if the requirements are met, the voting process proceeds according to the following requirements defined in the SC. (4) If the SC-based validation is confirmed, voting for the acceptance of this new certificate begins. If the candidate is an issuer certificate then SBs can vote; however, if it is an end-user certificate then it is the CAs that can vote.

The requirements of the voting mechanism are specified as follows: Voters must have valid certificates, and have an adequate stake. The waiting time (penalty) must expire of voters who are punished for previously registering improper votes. If a *cryptographicSignature* is given in the certificate, its verification must be conducted by the participants during voting. Voters need to assess documents such as the registration documents and audit reports uploaded by the issuer. Only a single vote can be given for each certificate candidate. Each voting operation has a cost, with the *voting threshold* value used to prevent excessive cost expenditure. As such, voting is discontinued if positive or negative votes counts reach the voting threshold defined in the SC. (5) Only candidates that successfully pass the voting mechanism will be issued. Even though an attacker may compromise a supervisory body, it cannot forge a trusted supervisory body or a CA, which prevents the *Single Point of Failure*. (6) Reward tokens are shared among the winning voters.

2) CERTIFICATE REVOCATION

Certificate revocation is a critical process that invalidates a digital certificate for various reasons, including compromise of the associated private key, loss or deletion of the certificate, changes in entity information, discontinuation of use, or suspicion of misuse. Revocation of a certificate means invalidating the certificate before its expiration date. In the traditional infrastructure, end-users or CAs can initiate the revocation process. However, the process itself can only be conducted by only CAs. CAs provide certificate revocation information through mechanisms like the Certificate Revocation List (CRL) and the Online Certificate Status Protocol (OCSP).

In contrast, the proposed SemiDec-PKI introduces a novel approach to the revocation process. In this system, the responsibility for certificate revocation lies with the certificate owners or issuers, who can initiate the revocation process via smart contracts (SC) as depicted in Fig. 4. This decentralized revocation process empowers certificate owners to take prompt action and eliminate the need for a *revocation monopoly*, enhancing the efficiency and responsiveness of the PKI infrastructure.

3) REVOCATION TRIGGERED BY THE ISSUER

When the certificate issuer initiates the revocation mechanism in the proposed SemiDec-PKI, some validations take place as shown in Fig. 5.

4) REVOCATION TRIGGERED BY THE OWNER

If the certificate to be revoked includes the owner's Ethereum address, then the owner can revoke it by sending a transaction. Please note that supervisory bodies are unlikely to be revoked.

Since the initial supervisory bodies are defined in a smart contract, they do not have an issuer. However, if a supervisory body is compromised and labeled as malicious based on the voting mechanism carried out by other supervisory bodies as given in the audit and fraud reporting mechanism, then their certificates are revoked. This decentralized revocation process empowers certificate owners to take prompt action in response to security concerns and ensures the overall integrity and trustworthiness of the SemiDec-PKI system. Moreover, it adds a layer of resilience and agility to the PKI system, enhancing its adaptability to changing security requirements and potential threats by eliminating *revocation monopoly*.

5) AUDIT AND FRAUD REPORTING

Any participant who volunteers to become a full node can be an auditor. In the fraud reporting mechanism, for the inclusion of a candidate certificate, a new vote is arranged, and the previous voters are blocked from the current voting by being added to a temporary ban list. When an auditor detects a forgery regarding certificates in the system, the steps shown in Fig. 6 are applied in the following order: In step 1, the auditor uploads evidence to IPFS. In step 2, the auditor sends a transaction for fraud reporting using ERC-20 as an assurance. In step 3, all those who previously voted for the reported certificate are banned recursively. In step 4, voters assess the evidence. In step 5, eligible authorities vote for the fraud report until the voting threshold has been reached. In step 6, the certificate's status is updated according to the election result. In the final step, when the voting concludes as fraud, the *revocationStatus* of the certificate is updated, and the auditor and winner voters punish the previous voters and the issuer by getting their tokens and applying them with an incremental waiting time penalty. If the voting fails to conclude fraud, the winner voters earn ERC-20 by receiving the auditor's assurance tokens.

As previously stated, voting is concluded when either the positive or negative votes reach a certain threshold. In theory, numerous attackers can dominate the voting for a transaction because of this threshold-based approach. However, auditors who can send a fraud transaction by a stake can prevent such cases. Once the transaction is sent, a chain of events is triggered. Firstly, SC blocks voters in the previous voting and starts new voting. Validators vote, and the reporter and new voters receive the ERC-20 tokens by punishing attackers where fraud is proven. However, where fraud is not proven, the tokens staked by the auditor are shared among the validators.

In summary, the proposed approach, SemiDec-PKI, effectively tackles various infrastructure challenges, including *trust list management*, *Single Point of Failure (SPoF) prevention*, *certificate registration*, *revocation monopoly*, *punishment*, *monitoring and logging*. Trust Lists are efficiently managed and governed by predefined Supervisory Bodies (SBs) within smart contracts (SC), streamlining the *trust list management* process. The SemiDec-PKI architecture, which combines Web of Trust and centralized approaches, ensures

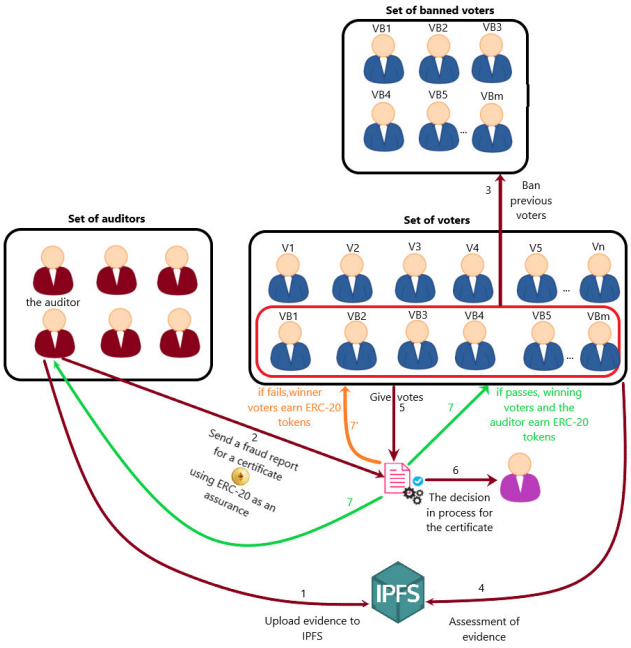

FIGURE 6. Fraud reporting mechanism.

resilience against *SPoF* by involving multiple CAs in the certificate issuance process. The introduced voting scheme significantly enhances the system's robustness, thwarting impersonation attacks arising from *certification registration* issues. Furthermore, our approach eliminates the need for centralized revocation authorities, empowering end-users to independently revoke their certificates and eradicating the *revocation monopoly* issues. A stake reward and *punishment* mechanism promotes active participation among end-users and voters by incentivizing positive contributions and disincentivizing malicious or erroneous behavior. This mechanism ensures a collaborative and secure environment within the system. Furthermore, SemiDec-PKI utilizes a hierarchical structure that simplifies auditing and cross-verification of the entire PKI ecosystem, allowing for *monitoring and logging*. To comprehensively evaluate the proposed approach, subsequent sections delve into more detailed analyses of its security, usability and performance aspects.

IV. SECURITY, USABILITY AND PERFORMANCE

In this section, we discuss the applicability of the designed system on the basis of security and performance.

A. SECURITY

Threats to the proposed design are investigated under three groups: Threats against the infrastructure; cryptographic threats; blockchain-based threats.

Threats Against the Infrastructure: A semi-decentralized structure is established in SemiDec-PKI, and the proposed voting mechanism eliminates single points of failure. Even if the most potent entities, supervisory bodies, are compromised, attackers cannot issue a CA certificate, supervisory certificate, or an end-user certificate.

In SemiDec-PKI, the validation steps are secured by the proposed stake-based reward-punishment mechanism, and a voting threshold value is defined to limit the number of transactions for each voting. Since voters are composed of supervisory bodies and CAs, they are unlikely to be compromised. However, in an extraordinary scenario in which the number of attackers (n) is bigger than or equal to the threshold (m) (i.e. $n \geq m$), attackers could potentially manipulate the first voting. Of course, an honest auditor would submit a fraud report for the transaction in this scenario, so a new voting would be triggered. Therefore, m attackers that took part in the previous voting would be blocked; hence the remaining $n - m$ attackers would still be able to participate in the new voting. In the case that the sum of $n - m$ is still may be greater than m ($n - m \geq m$), attackers could still dominate the voting, and a part of the auditor's stake would be given to the attackers. These steps are therefore repeated for n/m times, so that the attackers would lose in any voting process where dominated by benign nodes. As a result, the tokens of benign validators would be refunded, whilst the attackers would lose a considerable amount of tokens, and the attackers' certificates could be revoked due to the fraud reporting mechanism.

CA votes after validating the application of end-users. In that process, CA can be exposed to impersonation attacks. For an attacker A, it must deceive over than the threshold T number of CA or SBs. Assume that, the total number of CAs is n in the system and an attacker A can deceive CAs with probability p_r. The number of deceived CAs Z follow the probability distribution on $P(n, p)$. Then, the probability of being $Z \geq T$ can be calculated as in [12]:

$$P_r(Z \geq T) = \sum_{x=T}^{n} C_n^x p^x (1-p)^{n-x} \tag{1}$$

The correlation between P and n, T is illustrated in Fig. 7. It is given that for any n, it can be selected a proper T in order to minimize the probability. Even p_r is picked as 0.5, it is hard to achieve an impersonation attack and deceive a CA in real life. In each voting process, the participation of multiple Certificate Authorities (CAs) enhances the security of the certificate registration process by facilitating increased cross-checking. This multi-party involvement ensures a higher level of scrutiny and validation, thereby reducing the chances of fraudulent or erroneous certificate issuance. However, it is essential to consider that every voting transaction incurs gas costs within the smart contract (SC) environment. As a consequence, conducting multiple voting sessions may lead to an overall decrease in the cost-effectiveness of the system, affecting its efficiency within the SC framework. Striking a balance between the enhanced security achieved through increased cross-checking and the associated gas costs is critical to optimizing the SemiDec-PKI system's performance and ensuring its practicality and sustainability within the blockchain-based ecosystem. So we initially picking five CAs (T=5) in order

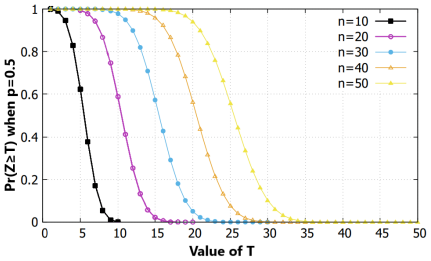

FIGURE 7. The probability of deceiving T CAs; assuming an impersonation attack success rate is 50%. $p = 0.5$.

to increase security and reduce assuming risk from 0.5 to $0.5^5(0, 03125)$. However, to maintain flexibility and optimize efficiency, the voting threshold value needs to be adjustable. Thus, we implement a smart contract feature that allows initiating a voting process for changing the threshold value. When this voting is triggered, the approval of at least $(n/2) + 1$ voters is required to approve the threshold change. By empowering the stakeholders to modify the threshold value, the SemiDec-PKI system ensures adaptability to changing security requirements while maintaining a practical balance between security and efficiency.

Cryptographic Threats: Thanks to the immutability characteristic of blockchain, all transactions and certificates are protected in SemiDec-PKI, as supervisory bodies and CAs use not only transaction keys but also private keys for signing. Thus, an attacker would need to take possession of both keys in order to render harm to the system. Please note that SemiDec-PKI uses ECC256 keys and allows rekeying.

Blockchain-Based Threats: The proposed infrastructure is built on Ethereum, which is still hypothetically vulnerable to threats in which attackers compromise 51% of the system. If a group of attackers controls more than 50% of the mining hash rate or processing power, they could block new transactions from being confirmed, thereby stopping payments among users [39]. While this threat is still proof-of-stake, attackers would need to outlay considerable amount of money in order to take control of 51% of the Ethereum (ETH). In addition, such actions would trigger a downturn in the value of Ethereum, which would directly conflict with the attackers' motivation.

B. USABILITY

In this study, we present a novel approach that does not rely on the ASN.1 format, which is essential for representing data within X.509 certificates. These certificates are easily understood when used in common systems like web browsers. However, things get more complicated when we enter the realm of blockchain-based smart contracts [40], [41]. Our approach focuses on efficiently storing specific X.509 attributes as variables within the smart contract structure. Our challenge here is to provide a smooth transition to a smart contract-friendly certificate structure without sacrificing the familiarity and reliability associated with the X.509 format. Therefore SemiDec-PKI enables the attachment of an IPFS

address for the standard X.509 certificate alongside the new blockchain-based certificates for backward compatibility. User could choose between these two formats based on their specific requirements and preferences during the transition.

C. PERFORMANCE

End-users are light nodes, and voters and auditors are defined as full nodes in the proposed infrastructure. The PoW algorithm is utilized in the current Ethereum system. However, the increased number of transactions (Txn) and the subsequent rise in Ethereum prices would lead to an increase in the total price for transactions and deploying SCs. Therefore, this consensus algorithm is planned to be changed to Proof of Stake (PoS) in Ethereum 2.0. Hence, it will be able to support more transactions and the use of SCs. Moreover, it is expected to decrease energy consumption.

Here, the storage cost of the proposed infrastructure based on blockchain is calculated as in [10]. There are about 3.64×10^8 registered domain names (as of 2021 [30]) and 46 million websites using SSL certificates [42]. Therefore, in a worst-case scenario, let us consider that half of all registered domains are assumed to use SSL certificates [43]; the storage cost is calculated as follows, based on the following assumptions:

- Estimating the number of other certificates used for purposes such as facilitating electronic signatures and code signing is challenging. This difficulty arises from the fact that these certificates are not centrally logged. Hence we have assumed that the sum of these certificates is equivalent to the sum of SSL certificates ($3.64 \times 10^8 \div 2$). Hence, the total number of all certificates is assumed to be equal to 3.64×10^8.
- Redefined certificates in this proposal would be approximately 256 bytes, or half of the average X509 certificate size (512 bytes) given in CertLedger [10].
- SSL certificate maximum lifetime period is defined as approximately 1 year (398 days) in a recent CAB Forum Ballot [34]. For the sake of simplicity, the worst case scenario is adopted for all other certificate types, and the certificate lifetime is defined as 1 year.
- PoS is used as the consensus algorithm in Ethereum. In PoS, the average block time, which is the time it takes to generate a new block, is 12-14 seconds [30].
- Adding new certificates is expected to cover the majority of transactions (Txn) in SemiDec-PKI, so the costs of all other transactions are ignored here.
- We also assume that certificates are issued homogeneously throughout the year, and the block time is 12 seconds [10].

Hence, five blocks are generated in 60 seconds. Then, the total number of blocks generated in 1 year would be 26,280,000 ($365 \times 24 \times 60 \times 5$). The number of transactions (Txn) is equal to the total number of certificates (3.64×10^8) / generated blocks in a year, as in CertLedger [10] shown in the following:

$$Number\ of\ Txn := \frac{Number\ of\ Certificates}{Annually\ Generated\ Blocks} \quad (2)$$

The size of a block (BS) is given in Equation 3.

$$BS := Txn\ Size \times Number\ of\ Transactions + Header \quad (3)$$

$$Txn\ Size := Message + Signature \quad (4)$$

$$Message := PKSender + Receiver + Data \quad (5)$$

where PKSender, Receiver, Data, and Signature correspond to the size of the sender's public key (64 B), the receiver's address (20 B), the certificate (256 B), and the size of the signature (64 B) in a transaction. Hence, the transaction size becomes 404 B. The header size of a block is fixed at 508 B [44]. Hence, the average size of a block becomes 6,103 B. The total size of the blockchain are calculated as in CertLedger [10]:

$$Blockchain\ Size := Annually\ Generated\ Blocks \times BS \quad (6)$$

Hence, the blockchain size of a full node is approximately 150 GB, as given in Equation 5. Since the cost of 1 GB of disk storage is about 0.02 USD [45], the combined cost for all certificate transactions would be approximately 3 USD per annum. On the other hand, light nodes do not store the entire block, just the header (508 bytes per block). As such, the total blockchain size of a light node per year would be approximately 640 MB ($26,280,00 \times 508$).

In the literature, certificates are logged to the blockchain and sent to the client in X509 format. In this study, the log in the blockchain was used without sending a separate certificate in the X509 standard. While the X509 certificate size is 512 B on average [10], our certificates are approximately 256 B.

Certificate issuance time is not measurable because registration authorities checks the appliance documents and gives votes according to validation result. But in the certificate transparency which in usage on SSL when a certificate is submitted to a log successfully, the server sends a Signed Certificate Timestamp (SCT) as proof and promise to add the certificate in the Merkle Tree within a fixed amount of time known as the Maximum Merge Delay (MMD) [25]. MMD is usually 24 hours [46]. While the voting mechanism ensures cross-checking, CA cross-check durations are expected not to cause performance issues if they are reasonable.

V. RELATED WORK

This section categorizes related studies into two groups. Firstly, it will delve into blockchain-based proposals. Secondly, it will cover log-based studies, as the Certificate Transparency project [47], still in use, falls under this category.

A. BLOCKCHAIN-BASED STUDIES

Blockchain has been widely used in various fields recently, and obtained remarkable results [31], [48], [49], [50], [51]. However, PKI studies on blockchain are few in the literature, and new studies are needed in this field. These studies and the important findings obtained from these studies are summarized in this section.

In Al-Bassam [8], a smart contract-based PKI and identity system (SCPKI) based on a Web of Trust model is proposed. It utilizes SCs in order to detect fraudulent attempts. In their trust model, participants can add their attributes to the blockchain, whilst other participants can acknowledge trust in them by sending new transactions. In SCs, executing transactions and changing states have a cost, so an incentive mechanism needs to be introduced in order to urge participants to contribute to the system. Moreover, SCPKI is susceptible to attacks due to the lack of a punishment mechanism.

Yakubov et al. [9] designed a blockchain-based PKI management structure for issuing, validating, and revoking X.509 certificates. According to their design, a smart contract (SC) is created for each CA, including a CA certificate, a digest of each certificate issued by that CA, and their revocation status. They defined a new X.509 extension and added the SC address to the certificate in order to create a link between certificates and SCs. However, the study did not define a supervisory system for SCs; hence, attackers can deploy SCs to the blockchain to issue root certificates and, thus, generate fake end-user certificates.

Kubilay et al. [10] proposed a PKI architecture called CertLedger to validate, store, and revoke SSL certificates, and to manage trusted CA certificates within an open blockchain. CertLedger prevents MITM attacks by making the certificate issuance and revocation lifecycle more transparent. While CertLedger stores the whole certificate within a transaction, storing and decoding an entire X.509 certificate on a smart contract can prove to be a costly process due to the information being encoded in ASN.1 [52]. Moreover, the CertLedger mechanism supports only SSL certificates and website URLs, and therefore it is not applicable to other certificate types or adaptable to new projects.

Kubilay et al. [11] proposed KORGAN, which is based on a permissioned blockchain with a Practical Byzantine Fault Tolerance (PBFT) consensus mechanism. The main contribution of the study is that end-users are not required to join the blockchain network; only public keys are needed to validate the dynamic threshold signatures of blocks. The study mainly focused on optimizing certificate validation during TLS negotiation.

Garba et al. [12] proposed a blockchain-based PKI (BB-PKI) model in order to prevent impersonation attacks resulting from compromised Registration Authorities (RA). They provide a transparent registration process by assigning RAs as the intermediary nodes between users and CAs for reviewing and approving certificate requests. BB-PKI introduces a Blockchain Maintenance Manager (BMM) in order to manage the activities of RAs and trusted CAs; however, it lacks a mechanism to manage trust lists. Moreover, it does not define the inclusion and exclusion of CAs.

Hwang et al. [13] proposed a semi-decentralized PKI system based on public blockchains that can easily prevent the *Single Point of Failure*. The proposed architecture defines four actors: web owner, CA, web user, and SC, but failed to introduce a supervisory mechanism. While it is a practical approach that optimally uses TP-Merkle trees, it assumes that all CAs will deploy SCs themselves. Hence, an attacker could create a fake CA smart contract and thereby issue a fake end-user certificate which, in the case of a man-in-the-middle attack, would not be detectable. Furthermore, the proposed design handles only SSL certificates and does not provide a solution for other certificates types.

Koa et al. [14] also proposed a mechanism based on a Web of Trust model called Ethereum-based PKI Identity management with a reward-punishment mechanism (ETHERST). It is mainly built on SCPKI with a new stake-based reward-punishment mechanism. However, threats against the validity of the design are not sufficiently discussed for scenarios where attackers may dominate the voting. Moreover, it does not have a fraud reporting mechanism for recovery.

Liang et al. [15] proposed a system called LRS_PKI, which is based on linkable ring signatures. In this approach, certificates are signed by a ring CA consisting of multiple CAs rather than only one CA, thereby enabling the hiding of the issuing CA in the PKI system. Additionally, it records certificate operations using certificate storage facilitated by the InterPlanetary File System (IPFS), leveraging the blockchain. Their methodology incorporates a verification step during certificate validation to ensure the integrity of the Certification Authority (CA) and mitigate malicious activities. While LRS_PKI focuses on SSL certificates, it lacks a punishment mechanism, commercial model, assurance mechanism, or fraud detection mechanism. Furthermore, it relies on X509 Certificates with new extensions.

SemiDec-PKI distinguishes itself from earlier work by integrating both Web of Trust and a centralized approach. This integration incorporates a voting scheme and a cross-check mechanism to mitigate SPoF vulnerabilities. In addition to other studies Semidec-PKI provides a trust mechanism not only for end-entity certificates but also CA certificates and supervisor bodies. Furthermore, the proposed blockchain-based certificate model is capable of supporting all types of digital certificates while ensuring monitorability. Last but not least, the study proposes an automated assurance mechanism to safeguard users from potential losses.

B. LOG-BASED STUDIES

Kim et al. proposes a novel Public Key Infrastructure (PKI) architecture called AKI [16] (Accountability in Key

Infrastructure) to reduce the level of trust placed in Certificate Authorities (CAs). Unlike conventional PKIs where CAs hold absolute authority, AKI involves multiple entities in all defined operations to distribute accountability. The new entities introduced by AKI include Certification Agency (CA), Integrity Log Server (ILS) Operators (ILSO), and Validators.

AKI makes a strong assumption that trusted entities (CAs, ILSs, Validators) do not collude, which may be unlikely in the face of a determined adversary. There are vulnerabilities in AKI such as a compromised CA and ILS are enough to generate a fake certificate, and a strong adversary could use this to conduct a split-world attack. Detection of this attack is not possible in AKI. Moreover, an adversary with a compromised domain private key can request certificate revocation without additional verification.

Basin et al. proposes ARPKI [17] (Advanced Accountability in Public Key Infrastructure) is an enhancement of AKI, providing a security guarantee against adversaries capable of compromising even less than or equal to $(n-1)$ trusted entities. In the context of ARPKI, the generation of an ARPKI certificate, referred to as ARCert, necessitates the involvement of at least two CAs and one ILS. It's important to note that ARPKI is still susceptible to a split-world attack if the entities required to generate an ARPKI certificate collude. Similar to AKI, ARPKI lacks a detection mechanism for this type of attack. Additionally, the designation of an ILS for synchronization with other ILSs introduces a potential single point of failure in the ARPKI system.

Szalachowski et al. [18] proposes Policert which operates as a public log-based program that facilitates the management, issuance, and enforcement of certificate policies. Multisignature certificates and subject certificate policies are logged on a public log server. However, this approach lacks established mechanisms to detect and control errant log behavior. Khan et al. [19] proposes Accountable and Transparent TLS Certificate Management which explore two different attacks for Policert and eliminates these attacks by introducing an improved revocation system and monitoring mechanism. Khan et al. [20] also proposes a secure and accountable TLS certificate management (SCM). In SCM, CA-signed domain certificates are stored in log servers which is conducted on the blockchain platform. Moreover SCM decreases the storage cost of blockchain dramatically. Khan et al. [21] also propose a log-based PKI called as Attack-Resilient TLS Certificate Transparency. ARCT eliminates impersonation attacks on registration process of certificate-issuance by collaborative certificate-issuance mechanism. In addition, it provides an revocation mechanism. However they don't provide a audit or issuance mechanisms for CA certificates.

VI. COMPARISON WITH RELATED STUDIES

This section will delve into a detailed discussion regarding the comparison with related works. The comparison criteria are selected according to cover the most critical shortcomings of PKI and mainly taken from Certledger [10]. In addition to them, new criteria regarding security and operation are added as shown in Table 3: *Punishment Mechanism*, *Commercial Model*, *Enables Cross-check*, *Support Any Certificate*.

Certificate Validation: SemiDec-PKI does not rely on third-party certificate validation as in the traditional PKI, since the certificate itself contains information about its expiration date and revocation status. As in Certledger [10], KORGAN [11], LRS_PKI [15] and Hwang et al. [13], clients only need to verify proofs. Yakubov et al. [9] utilizes smart contracts or web services. SCPKI [8] and ETHERST [14] depend on the Web of Trust. SCPKI [8] redefines the certificate as *Attribute* and validates its *Signature* and *Revocation* as a validation step. ETHERST [14] contributes SCPKI and adds *trustorCount* attribute to *Signature*.

Log Proofs: In Certificate Transparency, end-users or CAs make certificate-related logs into servers managed by different centers. There may be differences and synchronization issues in these log servers that tried to be corrected with the gossip protocol [47]. In SemiDec-PKI, all data, including the certificate itself, is kept on the blockchain. Therefore, a single log is copied multiple times and distributed in the system. Thus, inconsistent logs related to a certificate are prevented from being found on different servers. SemiDec-PKI provides proof of existence and revocation status for all certificates as in Certledger [10], KORGAN [11], Yakubov et al. [9], BBPKI [12], Hwang et al. [13], SCPKI [8], and ETHERST [14], LRS_PKI [15].

Auditing - Monitoring: The Certificate Transparency project reveals the concept of monitoring and auditing certificates. Third-parties audit certificate logs on different, partially independent log servers. This way, it is ensured that the logs on different log servers are consistent. Although Certledger [10], Yakubov et al. [9], and BBPKI [12] do not need an audit for consistency, they do not offer an additional audit mechanism before the certificates (CA and end-user) are produced and valid. A CA may generate a certificate that should not have been generated in the first place, and this generation cannot be revealed until a complaint. In this situation, end-users are expected to notice that CAs have been attacked or their certificates are issued by mistake. Likewise, these logs must be audited by third parties. In Hwang et al. [13], the end-user is also included in the certificate generation in order to solve this issue. On the other hand, Yakubov et al. [9] introduces an audit mechanism, and KORGAN [11] further builds upon this concept by incorporating a threshold signature mechanism, making audits verifiable for anyone possessing the public key of block signers. These solutions, however, necessitate external monitoring for their effectiveness. While, Yakubov et al. [9] and KORGAN [11] requires an external monitoring, SemiDec-PKI does not due to its own audit and monitoring mechanism. LRS_PKI [15] provides an strong audit process due to the use of ring PKI.

TABLE 3. Comparison of security and certificate management.

	SCPKI Al-Bassam 2017 [8]	Yakubov et al. 2018 [9]	Certledger Kubilay et al. 2019 [10]	KORGAN Kubilay et al. 2020 [11]	BBPKI Garba et al. 2020 [12]	Hwang et al. 2020 [13]	ETHERST Koa et al. 2021 [14]	LRS_PKI Liang et al. 2023 [15]	SemiDec-PKI
External Dependency During Certificate Validation	No	Yes	No	No	Yes	No	No	No	No
Existence of Logs with Different Content	No	No	No	No	No	No	No	No	No
Necessity of External Audit	Yes	No	No	No	No	Yes	Yes	No	No
Necessity of External Monitor	Yes	Yes	No	No	No	No	Yes	No	No
Assurance Mechanism	No	No	No	No	No	No	No	No	Yes
Punishment Mechanism	No	No	No	No	No	No	Yes	No	Yes
Commercial Model	No	No	Yes	No	No	No	Yes	No	Yes
Prevent Single Point of Failure	No	No	Partly[a]	Partly[a]	Partly[b]	No	Partly[c]	Yes	Yes
Enables Cross-check for Certificates	No[d]	No	No	No	Yes	No	Partly[c]	Yes	Yes
Trust List Management	No[d]	Yes[e]	Partly[g]	Partly[h]	Partly[g]	Partly[g]	No[d]	Yes	Yes
Storing Certificate	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes
Support Any Certificate	Partly[i]	No[f]	No[f]	No[f]	No[f]	No[f]	Partly[i]	No[f]	Yes
Require X509 Certificate	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Users' self revocation	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Architecture	Web of Trust	Centralized	Semi-decentralized	Semi-decentralized	Semi-decentralized	Semi-decentralized	Web of Trust	Semi-decentralized	Semi-decentralized

[a] If CA is compromised the attacker can issue a certificate
[b] Not for Supervisory
[c] Supports Voting Scheme on WoT
[d] Because of WoT Characteristics
[e] Only SSL Certificates but attackers can upload their smart contracts.
[f] Supports only SSL
[g] Adding supervisory bodies is not supported
[h] End-users need to store the validation keys of block signers
[i] Focus on identity management rather than certificates.

Assurance- Punishment - Commercial Model: CAs require to take out insurance in conventional PKI for the damages resulted from faulty certificates. In such cases, end-users and CAs must compromise or the matter goes to court. Certledger [10] proposes a fraud reporting mechanism that only suggests that penalties such as financial or total prohibition may be imposed. ETHERST [14] utilizes a reward-punishment mechanism to encourage commercial adaptation of blockchain-based PKI. In the web of trust mechanism, untrusted nodes are punished with PKITokens. Yakubov et al. [9], SCPKI [8], KORGAN [11], BBPKI [12], Hwang et al. [13], and LRS_PKI [15] do not provide a punishment mechanism. In SemiDec-PKI, this insurance is guaranteed by initial assurance tokens. Complaints are created, and the result of the complaint is concluded with a reward-punishment system. Certledger [10], ETHERST [14], and SCPKI [8] also provides a commercial model for incentive shareholders. While SCPKI [8] and ETHERST [14] utilizes ERC tokens, SemiDec-PKI provides a commercial model that includes an assurance mechanism in addition to the punishment-reward mechanism.

Single Point of Failure - Cross-Check: In traditional PKI, Yakubov et al. [9], Certledger [10], KORGAN [11], BBPKI [12], and Hwang et al. [13], the certificate is activated after the certificate is issued, and faulty situations are handled only afterwards. In addition, Hwang et al. [13] includes end-users in the system in order to increase controls on certificate issuance process. However, in cases where end-users could be less conscious, it is possible to generate problematic certificates. SemiDec-PKI provides a cross-check mechanism for each certificate that includes other CAs to the certificate issuance. Thanks to this mechanism, a certificate is not accepted and will only be activated with the review of at least two CAs. Therefore, even if an attacker captures a CA, he cannot generate end-user certificates without the permission of other CAs due to the voting mechanism. Even if an attacker compromises the supervisory body, the most powerful actor in the system, it can not issue a certificate without going through the voting mechanism. Thus, single-point-of-failure is prevented. When a CA's certificate is issued, Certledger [10] asks for the approval of more than one of the management board

members. However, when one of CAs is compromised, it cannot prevent attackers from issuing certificates. LRS_PKI [15] provides a cross check mechanism with ring based signature model. Although Ring CA offers many innovations, the status of adding new members to Ring CA is not fully explained.

Certificate Management: Certificates conform to the X509 standard and are encoded with ASN.1, which is a formal notation used for describing data transmitted by telecommunications protocols. In existing PKI models, certificates are produced by the CA and sent to end-users. Although Certledger [10], KORGAN [11], BBPKI [12], Hwang et al. [13], and Yakubov et al. [9], LRS_PKI [15] proposed new approaches, they still use the same X509 certificate standard. However Certledger [10] stated that encoding and decoding the certificate based on ASN.1 is problematic for smart contracts and decoding is not implemented. ASN.1 decoding is a challenging process for SCs and SemiDec-PKI suggest a new solution. SemiDec-PKI supports logging all certificate-related fields to the blockchain structure instead of using the certificate's X509 and ASN.1 notations. It utilizes these logs instead of X509 certificates. Thus, CAs can generate certificates only after filling in all certificate-related fields that are checked and approved by other CAs. Therefore, SemiDec-PKI does not need to follow the X509 standard. This approach is similar to the definition of *attribute* in SCPKI [8] and ETHERST [14], which replaces a new certificate.

Trust List Management: In conventional PKI, certificates are signed by the issuer while issued. While the certificate is being validated, a validation process occurs, starting from the end-user certificate to the root certificates of the issuers. For the verification, all issuers must have root certificates on the side that validates the certificate. The store where the issuers have root certificates is called a trust list. Clients do not have to store trusted keys or certificate logs during certificate verification on the client side in Certledger [10], BBPKI [12], and Yakubov et al. [9]. This way, certificate validation can be performed without a client-side trusted root store. KORGAN [11] also replaces and eliminates the client-side conventional trust list, but end-users must store the blockchain signing keys' public key. LRS [15] introduces a novel Ring Based PKI involving multiple CAs for certificate issuance. However, it does not offer a solution for expanding the ring of CAs with new members. Additionally, the concept of a management board that responsible for adding root certificates is discussed in Certledger [10]. However it may require to increase the number of management board members for its world-wide application. SemiDec-PKI provides an internal trust list and extendable supervisory mechanism to increase supervisor bodies. Thus, it will be possible to dynamically include supervisory bodies of countries that join the system. SCPKI [8] and ETHERST [14] adopts a Web of Trust-based approach and are therefore exempt from trusted root or supervisory body concepts.

Users' self revocation: Revocation of the certificate is an essential issue in PKI. In the current PKI scheme, end-users contact CAs and perform certificate revocation via CAs. Certledger [10], KORGAN [11], BBPKI [12], and LRS_PKI [15] support user self-certificate revocation. In Hwang et al. [13], users can send a change status request to Certificate Authority and perform the cancellation over the CA. Since there is no authority in SCPKI [8] and ETHERST [14], revocation can be done by only end-users. SemiDec-PKI also supports user self-certificate revocation.

VII. CONCLUSION AND DISCUSSION

Our study introduces a novel and robust Public Key Infrastructure (PKI) solution named as SemiDec-PKI, which leverages blockchain technology and smart contract-based mechanisms to address long-standing security challenges in the conventional PKI model. By combining elements of Web of Trust and centralized approaches, SemiDec-PKI significantly reduces the risks associated with single points of failure and impersonation attacks, thus enhancing the overall security of certificate issuance. One of the key innovations of SemiDec-PKI is the implementation of a stake-based reward-punishment mechanism, which ensures that stakeholders have a vested interest in maintaining the system's integrity. This mechanism not only incentivizes honest participation but also penalizes malicious actors, deterring them from attempting to manipulate the system. Moreover, the system's voting scheme provides a collaborative approach to certificate issuance and validation, involving multiple Certificate Authorities (CAs) and Supervisor Bodies (SBs). This approach not only increases security through cross-checking but also ensures that the PKI system can operate independently, eliminating the need for centralized control. Another noteworthy feature is the decentralized certificate revocation mechanism, which empowers certificate owners to initiate revocation independently through smart contracts. This decentralization eliminates the revocation monopoly that exists in traditional PKI systems, enhancing system responsiveness and trustworthiness. Moreover, SemiDec-PKI is designed to be highly adaptable, allowing for the adjustment of the voting threshold value to strike a balance between security and efficiency. This flexibility ensures that the system can evolve to meet changing security requirements.

To sum up, SemiDec-PKI presents a unique approach to the PKI architecture, leveraging blockchain and smart contracts to create a secure, adaptable, and decentralized system. By addressing critical issues such as single points of failure, impersonation attacks, and revocation monopolies, SemiDec-PKI contributes to the advancement of secure digital communication and transaction environments. This study underscores the suitability of Ethereum's smart contract capabilities for implementing such a system and paves the way for future research and development in the field of blockchain-based PKI solutions.

REFERENCES

[1] S. Khan, F. Luo, Z. Zhang, F. Ullah, F. Amin, S. F. Qadri, M. B. B. Heyat, R. Ruby, L. Wang, S. Ullah, M. Li, V. C. M. Leung, and K. Wu, "A survey on X.509 public-key infrastructure, certificate revocation, and their modern implementation on blockchain and ledger technologies," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2529–2568, 4th Quart., 2023.

[2] N. van der Meulen, "DigiNotar: Dissecting the first Dutch digital disaster," *J. Strategic Secur.*, vol. 6, no. 2, pp. 46–58, Jun. 2013. [Online]. Available: http://www.jstor.org/stable/26466760

[3] Comodo. *Comodo Incident Report*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html

[4] Rob Wright. *23,000 Symantec Certificates Revoked Following Leak of Private Keys*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.techtarget.com/searchsecurity/news/252436120/23000-Symantec-certificates-revoked-following-leak-of-private-keys

[5] H. Birge-Lee, Y. Sun, A. Edmundson, J. Rexford, and P. Mittal, "Bamboozling certificate authorities with BGP," in *Proc. 27th USENIX Secur. Symp. (USENIX Secur.)*. Baltimore, MD, USA: USENIX Association, Aug. 2018, pp. 833–849. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee

[6] 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, doi: 10.1109/COMST.2022.3178081.

[7] Adobe. *Adobe Approved Trusted List Website*. Accessed: Nov. 10, 2022. [Online]. Available: https://helpx.adobe.com/acrobat/kb/approved-trust-list2.html

[8] M. Al-Bassam, "SCPKI: A smart contract-based PKI and identity system," in *Proc. ACM Workshop Blockchain, Cryptocurrencies Contracts (BCC)*, Abu Dhabi, United Arab Emirates. New York, NY, USA: Association for Computing Machinery, 2017, pp. 35–40, doi: 10.1145/3055518.3055530.

[9] A. Yakubov, W. M. Shbair, A. Wallbom, D. Sanda, and R. State, "A blockchain-based PKI management framework," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp.*, Apr. 2018, pp. 1–6.

[10] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "CertLedger: A new PKI model with certificate transparency based on blockchain," *Comput. Secur.*, vol. 85, pp. 333–352, Aug. 2019.

[11] 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. 1, pp. 564–574, Nov. 2019.

[12] 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)*, Dec. 2020, pp. 824–829.

[13] G.-H. Hwang, T.-K. Chang, and H.-W. Chiang, "A semidecentralized PKI system based on public blockchains with automatic indemnification mechanism," *Secur. Commun. Netw.*, vol. 2021, pp. 1–15, Oct. 2021.

[14] C.-G. Koa, S.-H. Heng, and J.-J. Chin, "ETHERST: Ethereum-based public key infrastructure identity management with a reward-and-punishment mechanism," *Symmetry*, vol. 13, no. 9, p. 1640, Sep. 2021.

[15] W. Liang, L. You, and G. Hu, "LRS_PKI: A novel blockchain-based PKI framework using linkable ring signatures," *Comput. Netw.*, vol. 237, Dec. 2023, Art. no. 110043. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623004887

[16] 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*, May 2013, pp. 679–690.

[17] 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. Secur.* New York, NY, USA: Association for Computing Machinery, 2014, pp. 382–393.

[18] P. Szalachowski, S. Matsumoto, and A. Perrig, "PoliCert: Secure and flexible TLS certificate management," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 406–417, doi: 10.1145/2660267.2660355.

[19] 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," *Secur. Commun. Netw.*, vol. 2018, pp. 1–16, Jul. 2018.

[20] 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, p. e4503, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4503, doi: 10.1002/dac.4503.

[21] 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.

[22] Transport Infrastructure. *News*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.uab.gov.tr/haberler/2023-yili-2-nci-ceyregine-iliskin-turkiye-elektronik-haberlesme-sektoru-3-aylik-pazar-verileri-raporu-aciklandi

[23] Indian Institutes of Technology. *Market Data in Turkey*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.btk.gov.tr/pazar-verileri

[24] *The Directory: Public-Key and Attribute Certificate Frameworks*, ITU Standard X.509, Oct. 2019.

[25] *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP*, Standard RFC 6960, Jun. 2013.

[26] M. Anisetti, C. A. Ardagna, and N. Bena, "Multi-dimensional certification of modern distributed systems," *IEEE Trans. Services Comput.*, vol. 16, no. 3, pp. 1999–2012, May 2023, doi: 10.1109/TSC.2022.3195071.

[27] R. Neisse, J. L. Hernández-Ramos, S. N. Matheu, G. Baldini, and A. Skarmeta, "Toward a blockchain-based platform to manage cybersecurity certification of IoT devices," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2019, pp. 1–6.

[28] *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Standard RFC 5280, May 2008.

[29] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85727–85745, 2019.

[30] *Ethereum Website: Nodes*. Accessed: Nov. 10, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/nodes-and-clients/

[31] P. Chinnasamy, A. Albakri, M. Khan, A. A. Raja, A. Kiran, and J. C. Babu, "Smart contract-enabled secure sharing of health data for a mobile cloud-based e-health system," *Appl. Sci.*, vol. 13, no. 6, p. 3970, Mar. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/6/3970

[32] Ethereum Request for Comments. *Ethereum Request for Comments (ERC-20) Homepage*. Accessed: Nov. 10, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[33] C-Forum. *Cab Forum Ballot 185*. Accessed: Nov. 10, 2022. [Online]. Available: https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-v2.0.1.pdf

[34] C-Forum. *Cab Forum Ballot 185*. Accessed: Nov. 10, 2022. [Online]. Available: https://cabforum.org/2017/02/24/ballot-185-limiting-lifetime-certificates

[35] *The EU Cyber Security Agency: Guidelines on Supervision of Qualified Trust Services—Technical Guidelines on Trust Services*, ENISA, Dec. 2017.

[36] *Commission Implementing Desicion*. Accessed: Nov. 10, 2022. [Online]. Available: https://ec.europa.eu/futurium/en/system/files/ged/celex_32015d1505_en_txt.pdf

[37] *CA/Browser Forum*. Accessed: Jul. 21, 2023. [Online]. Available: https://cabforum.org/

[38] *Regulation (EU), no. 910/2014 of the European Parliament and of the Council of 23 July 2014 on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market and Repealing Directive 1999/93/EC*, document 32014R0910, Eur. Commission, 2014.

[39] Investopedia. *Diginotar 531 Fraudulent Certificates*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.investopedia.com/terms/1/51-attack.asp

[40] J. Groendal. *ASN1 Decode Project*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.btk.gov.tr/pazar-verileri

[41] Code4Rena. *Bytes Transform Codes for ASN.1 Decoding*. Accessed: Nov. 10, 2022. [Online]. Available: https://github.com/code-423n4/2023-04-ens/tree/main/contracts/dnssec-oracle

[42] Serpwatch. *Sepwatch SSL Statistics*. Accessed: Nov. 10, 2022. [Online]. Available: https://serpwatch.io/blog/ssl-stats

[43] *Wired Homepage*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.wired.com/2017/01/half-web-now-encrypted-makes-everyone-safer/

[44] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–34, Aug. 2014.

[45] GC Website. *HDD Prices*. Accessed: Nov. 10, 2022. [Online]. Available: https://cloud.google.com/storage/pricing
[46] Certificate Transparency DEV Website. Accessed: Nov. 10, 2022. *How CT Works*. https://certificate.transparency.dev/howctworks/
[47] *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*, Standard RFC 3161, RFC Editor, Aug. 2001.
[48] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[49] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology? A systematic review," *PLoS One*, vol. 11, no. 10, Oct. 2016, Art. no. e0163477.
[50] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2017, pp. 557–564.
[51] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.
[52] *OSI Networking and System Aspects—Abstract Syntax Notation One (ASN.1)*, ITU Standard X.680, Feb. 2021.

ERHAN TURAN is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Hacettepe University. He is a member of the Wireless Networks and Intelligent Secure Systems (WISE) Laboratory. He has been involved on PKI and the electronic signature technologies for more than ten years.

SEVIL SEN is currently a Professor with the Department of Computer Engineering, Hacettepe University, and leads the Wireless Networks and Intelligent Secure Systems (WISE) Laboratory. Her research interests include computer networks and network and systems security. Her focus is mainly in the area of mobile systems and wireless networks. She is also serving as an Area Editor for *Ad Hoc Networks* and *Genetic Programming and Evolvable Machines*.

TAMER ERGUN has been a Researcher, studying cryptography, PKI, and electronic signature technologies, and professionally working on these topics, since 2005. He started his professional career as a Cryptography Researcher with TÜBİTAK, and after 2012, he was with Turkish National Certification Authority (Kamu SM), as the Head of e-Signature Technologies. He has continued his studies with Imza.io, since 2023. Throughout his career, he has involved in the aforementioned fields not only as a Researcher but also as a Software Developer, a Consultant, an Auditor, and a Trainer.