"LINKABLE RING SIGNATURES FOR BLOCKCHAIN PRIVACY PROTECTION"

Dr. Basavaraj G M
Professor,
Don Bosco Institute of technology, Bengaluru, IndiaPin – 560074
Email – gmb2206@gmail.com

Neethu A S
Department of Information Science and Engineering Don Bosco Institute of Technology, Bengaluru, India
Pin – 560074
Email – asneethu14@gmail.com

Pradeepkumar V
Department of Information Science and EngineeringDon Bosco Institute of Technology, Bengaluru, India
Pin – 560074
Email – vppradeep0123@gmail.com

Sparsha C S
Department of Information Science and EngineeringDon Bosco Institute of Technology, Bengaluru, India
Pin – 560074
Email – sparshashetty9@gmail.com

Prajwal V
Department of Information Science and Engineering Don Bosco Institute of Technology, Bengaluru, India
Pin – 560074
Email – prajuu2000@gmail.com

ABSTRACT: A numerous security vulnerabilities have emerged within the PK system. For example, a compromised CA can issue illegal or fake certificates for any domains, and a CA can issue unauthorized certificates without the consent of the domain owner. In addition, some high-value target domains, such as bank and government agencies may have been frequently attacked, and the adversaries can launch thetargeted attacks by making use of the disclosure of theissuing CAs. To address these pressing issues or challenges, in this work, we propose a novel blockchain-based PKI framework using linkable ring signatures, called LRS_PKI. Specially, we propose a novel certificate issuance mechanism that utilizes linkable ring signatures to hide the issuing CA, so as to reduce the risk of the PKI system being attacked. Additionally, we introduce the blockchain as a public log to record the certificate operations, and adopt the decentralized storage IPFS to store the certificates to decouple the blockchain layer and storage layer. In order to prevent the CA from issuing unauthorized certificates, we have added a condition to verify whether the issuing CA in the certificate verification.

INTRODUCTION: In today's digital landscape, privacy and security in online communication are paramount. Transport Layer Security (TLS) stands as widely-used encryption protocol, ensuring the security And integrity of network communication. TLS relies on Public Key Infrastructure (PKI) for identityauthentication and secure communication. PKI utilizesdigital certificates issued by Certificate Authorities (CAs) to verify the identities of website. Cas play a crucial role in issuing, managing, and revoking certificates, making them vital to PKI's security. Thus, the integrity of PKI hinges on the trustworthiness of the Cas issuing TLS certificates to websites.

However, such a centralized architecture of PKI has brought great challenges to secure communication in reality.

[1] CAs, or Certificate Authorities, have been targetedin cyber attacks, compromising the security of internet communication. In 2011, both Comodo and Digi Notar, prominent CAs, sell victim to hacks. Attackers were able to steal digital certificates, allowing them to impersonate legitimate websites like mail.google.com and login.yahoo.com. This breach enabled attackers toconduct man-in-the-middle attacks, putting millions ofusers at risk of having their data intercepted and manipulated.

[2] Weakness in Registration Authority (RA) auditing process have been exploited, allowing individuals without domain ownership to obtain legitimate certificates. In a notable case in September 2016, Wosign was discovered to have issued root certificatesfor domains like github.com to individuals

Who only owned subdomains on platforms like GitHub. This oversight allowed anyone with aregistered account on GitHub to claim ownership of asubdomain and obtain a certificate for the parent domain.

[3] CAs wield significant power, capable of issuing certificates without the consent of domain owners. In notable instances, Thawte, a subsidiary of Symantec, issued thousands for various domain names, including Google-owned and nonexistent domains, without proper authorization. Similarly, Trustwave distributed subordinated root certificates to users for network traffic monitoring but ended up issuing fraudulent TLScertificates to multiple domains.

These incidents highlight the inherent risks in centralized PKI systems, where a single mistake or attack on a CA can severely disrupt secure communication. To address these issues, existing solutions typically fall into two categories: CA-based trust dispersal and log-based PKI systems.

CA-based Trust Disperse: involves mitigating the power of a single CA by involving multiple CAs or entities in certificate operations. This strategy includesmethods like the PGP web of trust [1], where identities are verified through digitally signed certificates from other users. Another approach is ARPKI [2], which coordinates certificate signature and verification across multiple CAs in a sequential manner.

Log-based PKI: relies on centralized log servers to publicly record certificates issued by CAs for auditing purposes. These log servers maintain a structured log of certificate issuance using techniques like Certificate Transparency (CT) [3], AKI [4], and CIRT.CT [5], for instance, employs a unified hash tree to log certificate operations, ensuring transparency and accountability in PKI system.

Blockchain-based PKI integrates the strengths of both CA-based trust dispersal and log-based PKI, while also addressing issues of low-cost trust andpublic auditability. Leveraging the decentralized, tamper-resistant, and traceable nature of blockchain technology, it offers a solution to mitigate the risks associated with a centralized CA system, including single points of failure and CA misconduct. Projects like Certcoin [6], IKP [7], CertLedger [8], Certchain [9], and BCTRT [10] have demonstrated the benefits of blockchain-based PKI. However, challenges remain, such as:

[1] **Privacy-preserving of issuing CA :** Preserving the privacy of issuing Certificate Authorities (CAs) presents a critical challenge in PKI systems. While current privacy measures focus on protecting user identity data, the anonymity of CAs is often overlooked. However, exposing CAs through publiclylogged operations can make them vulnerable to targeted attacks, especially for high-value entities likebanks or government agencies. Attackers can exploit this information to identify active CAs and launch attacks against them, expanding the PKI system's attack surface. Therefore, developing a privacy-preserving scheme for CAs is essential to enhance PKIsecurity.

[2] **A single CA has too much power:** The concentration of power within a single Certificate Authority (CA) poses a significant risk in PKI systems.Any CA holds the authority to issue valid certificates for any domain, meaning a compromised CA can affectthe security of all domains on the network. Additionally, even a non-domain issuing CA could potentially issue a valid certificate for a domain, further exacerbating security concerns. This highlights the need to address the excessive authority granted to individual CAs to strengthen PKI system security.

[3] **Certificates Storage:** Storing certificates on existing blockchains poses challenges due to limited block sizes and generation times. To address this, a novel blockchain-based PKI framework called LRS_PKI is proposed. LRS_PKI utilizes linkable ring signatures to hide the issuing Certificate Authority (CA), enhancing privacy and protection against targeted attacks. It introduces the concept of a "RingCA" in the certificate trust chain. Additionally, LRS_PKI employs the Link algorithm to verify theconsistency of issuing CAs during certificatevalidation, limiting CA power. Certificates are storedtransparently and traceably on the blockchain, with operations recorded using InterPlanetary File System(IPFS) to decouple storage from the blockchain layer. The article is structured into several sections: Section 2 discusses related work in the field of PublicKey Infrastructures (PKIs), Section3 provides background information necessary to comprehend LRS_PKI, and Section 4 outlines the basic architectureof LRS_PKI, including its threat model and design objectives. Section 5 offers a detailed description ofthe implementation of LRS_PKI, while Sections 6 and 7 analyze the security aspects and evaluate the performance of the system. Finally, conclusions drawnfrom the study are presented in Section 8.

RELATED WORK:

Current research on PKI focuses on several categories. First, efforts aim to disperse the trust of Certificate Authorities (CAs) in PKI systems by diminishing their authority, thus reducing the risk associated with a single point of failure. Second, there's a focus on Log-based PKI and blockchain- based PKI, which aim to enhance transparency and public accessibility of certificates. Lastly, to mitigate the storage overhead of blockchain-based PKI, certificates are stored in InterPlanetary File System (IPFS), a decentralized file storage system.

2.1 CA-based trust Disperse

Certainly, The decentralized Public Key Infrastructure (PKI) systems using blockchaintechnology. CA-based Trust Disperse, employing a Web of Trust (WOT), aims to decentralize trust by involving multiple Certificate Authorities (CAs) [2] incertificate signature and verification processes, ensuring security against forgery attacks even if someCAs are compromised. However, this method may suffer from efficiency issues and susceptibility to Denial of Service (DoS) attacks [12]. Conversely, the Dynamic Blockchain-based PKI (DBPKI) modeleliminates the need for CAs entirely, with PKI nodes autonomously managing key registration and revocation, bolstered by the Practical Byzantine FaultTolerance (PBFT) protocol for consensus. Smart Contract-based PKI (SCPKI) [13] leverages smart contracts and WOT principles to enable entities to verify each other's identity attributes, sidestepping traditional CA issuance. Despite advancements, challenges remain in ensuring non-repudiation, efficient revocation handling, and addressing priortrust relationship requirements. These models reflect ongoing efforts to devise decentralized PKI solutions capable of navigating the complexities of modern digital ecosystems.

2.2 Log-based PKI

The concept of log-based PKI revolves around recording certificate operations in an immutable publiclog, ensuring transparency and accountability in certificate issuance. Certificate Transparency (CT) [3], pioneered by Google, aims to achieve this transparent by making all CA-issued certificates publiclyaccessible and auditable. ARPKI [2], AKI [4], and PoliCert [14] schemes, inspired by CT, employ log servers to record certificate operations, enhancing transparency and enabling the detection of CA misconduct. CIRT [5] extends CT by incorporating two Merkle trees to record certificate operations and achieve revocation transparency (RT). However, log- based PKI has drawbacks, including centralized storage of log records and an expanded attack surface due to the presence of log servers, making systems vulnerable to split-world attacks [15] where attackers manipulate logs to deceive users. Additionally, there's a lack of effective mechanisms to incentivize recording, monitoring of CA behaviour.

2.3 Blockchain-based PKI

Blockchain-based PKI systems leverage the decentralized and immutable nature of blockchain technology to address the limitations of traditional PKItrust systems. Examples include CertCoin [6] and BlockStack [16], use Namecoin-based decentralized PKI systems, binding public keys to user identities or linking human-readable names with public keys. However, these systems do not fully cover PKI architecture aspects. PB-PKI and Cecoin propose privacy-enhanced approaches, but they lack compatibility with transparency requirements andcertificate revocation explanations. CertChain introduces a public auditable model with a new entity bookkeeper but risks leaking user privacy [18].CertLedger aims to prevent split-world attacks by managing certificate operations within the blockchain [8], while Yakubov's [19] framework assigns dedicatedsmart contracts to each CA for certificate issuance andrevocation recording. Proofchain decentralizes the CA pool and ensures compatibility with existing X.509 standards using proof-of-issuance (PoI) [20], but all these blockchain-based PKI schemes storingcertificates directly on the blockchain face challenges such as increased data volume and complexity in nodesynchronization, potentially compromising blockchainnetwork performance.

2.4 Certificate stored in IPFS:

Certainly! Several proposals advocate for enhancing the privacy and authentication capabilities of Public Key Infrastructure (PKI) systems by storing certificates in the InterPlanetary File System (IPFS). For instance, Chen et al. [21] introduced XAuth, a cross-domain authentication scheme integrated with Certificate Transparency (CT), which utilizes IPFS tostore certificate logs. This approach facilitates blockchain-based verification while preserving user privacy. Similarly, SCPKI, proposed by Al-Bassam etal. [13], leverages IPFS to store attribute data, enabling users to manage certificates and PGP keys through IPFS and PGP interfaces. Another example isthe blockchain-based PKI architecture proposed by Zhang et al.[22] for vehicular ad hoc networks (VANETs), which stores certificates in IPFS for direct verification by verifiers. However, despite these advancements, these schemes often lack sufficientprivacy protection for issuing Certificate Authorities (CAs), leaving them vulnerable to targeted attacks. Additionally, they may fail to prevent misbehaving CAs from issuing certificates, compromising system integrity. Addressing these concerns, the proposed LRS_PKI aims to provide robust privacy protection for CAs while mitigating storage expansion by avoiding direct certificate storage on the blockchain. Moreover, LRS_PKI seeks to distribute authority more evenly among CAs, preventing the dominance of a single CA and enhancing the overall security of the PKI system. Operating on a private blockchain, LRS_PKI is particularly well-suited for domains with stringent security requirements such as finance, e- government, and banking.

BACKGROUND:

3.1 Algorithm Model

The linkable ring structure scheme of LRS_PKI contains 5 sub-algorithms setup, Key Gen, Sign, Verify and Link as the following:

Setup(λ) $\rightarrow$ $params$: This algorithm takes a security parameter λ and outputs system parameters $params$.

KeyGen(λ, $params$) $\rightarrow$ (PK, sk): Given λ and $params$, this algorithm outputs a user's public key PK and private key sk.

Sign($M, n, S, sk\pi$) $\rightarrow$ (σ, $Q\pi$): Given a message M, aset of n member public keys S, and the signer's private key $sk\pi$, this algorithm outputs the signature
σ and the link tag $Q\pi$.

Verify($params$, M, S, σ, Q) $\rightarrow$ $accept$ $reject$: Given system parameters $params$, a message M, a set of member public keys S, a signature σ, and a link tag Q,this algorithm determines if the signature is valid.

Link($S, M1, M2, \sigma1, Q1, \sigma2, Q2$) $\rightarrow$ $linked$ $unlinked$:Given a set of user public keys S, two different messages $M1$ and $M2$, and their corresponding ring signatures ($\sigma1$, $Q1$) and ($\sigma2$, $Q2$), this algorithm determines if the signatures are valid and linkable.

3.2 Security Model

The Linkable Ring Signature (LRS) scheme ofLRS_PKI must satisfy three fundamental properties: unforgeability, anonymity, and link ability, as defined by the game involving a simulator and an adversary, along with various oracles. Unforgeability is defined interms of a game between the simulator and the adversary, where the adversary attempts to produce forged signatures. The adversary wins if it can create valid ring signatures on a given message with a set of public keys, without querying the corresponding private keys and without obtaining the signature through the signing oracle. This definition ensures thatthe LRS scheme is resistant to forging attacks, thus preserving the integrity of the signatures within the system.

Definition 1 (Unforgeability). If there is no ppt adversary that can win the following game with a non-negligible advantage, then the LRS scheme is considered unforgeable.

1. generate the system parameters params and send to them to □.

2. □ makes the adaptive queries to the □ □, □□, □□, andthe random oracle □.

3. □ outputs a message M* , a set of n public keys S*,and two torged signatures σ 0 ,σ 1.

□ Wins the game if the following four conditions hold:(1)

σ* , σ* are valid ring signatures on message M* .

(2) All the public keys in S* are obtaind by quering the □ □

(3) □ has not queried the corresponding private key inS*.

(4) σ* is not obtained by quering the □ □.

The advantage of in the above game is defined as:
$Adv for = Pr[wins]$.
An LRS scheme is considered anonymous if no probabilistic polynomial-time (PPT) adversary can achieve a significant advantage in winning a particular game.
An LRS scheme is deemed anonymous if no probabilistic polynomial-time (PPT) adversary, denotedas \(\mathcal{A} \), can gain a significant advantage inwinning the following game:

1. The challenger C generates system parameters paragrams and sends them to A.

2. A conducts adaptive queries to the oracle.

3. A sends a message M* along with a set of n user public keys S* = {PK_1, PK_2,…….., PKn }to C, where all public keys are obtained by querying the oracle. C randomly selects $\pi$from {1, 2,….., n}andcomputes the signature σπ= Sign (M*, n, S*, skπ), where skπ is the private key corresponding to PKπ, andthen sends σπ to A.

4.A outputs a guess ‘π’ from {1, 2,…., n}. If $'\pi' = \pi$, the adversary A wins the game.

The advantage of in the above game is defined as:

$$Advantage = |||Pr[\pi' = \pi] - 1\, n|||.$$

Definition 3 (Linkability): In the context of Linkability,if there exists no probabilistic polynomial-time (PPT) adversary π that can secure a non-negligible advantagein winning the described game, then the LRS scheme isdeemed linkable.

1. Party A generates the system parameters paramsand transmits them to Party B.

2. Party B dynamically makes inquiries to the functions f1,f2, and f3.
3. 3. Party B produces two sets of linkable ring signature values \((S, M_1, \sigma_1, Q_1)\) and \((S, M_2, \sigma_2, Q_2)\).
4. Party B produces two sets of linkable ring signature values S, M1, sigma1, Q1) and (S, M2,sigma2, Q2).

wins the game if the following four conditions arehold

Party B wins the game if the following four conditionsare met:
1. All the public keys in Party A are acquired by queryingthe function \(f_1 \).

2. Party B obtains at most one user's private key in PartyA by querying the function \(f_2 \).

3. The sets S, M1,sigma1, Q1 and S, M_2, sigma2, Q_2)\) constitute valid linkable ring signatures, and \(sigma_1 \) and \(\sigma_2 \) are not obtained by querying PartyA.

4. The link function \\text{Link}(S, M_1, M_2, \sigma_1, \sigma_2) \right arrow \text{unlinked} \) holdstrue.

The advantage of Party B in the above game is definedas:
\[\text{Advantage}_B = \text{Pr}[B \text{ wins}] \]

3.3. Linkable ring signatures algorithm

To safeguard the issuing Certificate Authority's (CA) privacy during certificate issuance, the LRS_PKI system employs a linkable ring signature algorithm. Thisalgorithm ensures that users can trust the certificate's origin within a group without disclosing the specificissuer. To address the storage and communication requirements of certificates in LRS_PKI, our linkable ring signature scheme prioritizes both rapid certificate responses and the privacy of issuing CAs. Hence, we adopt the linkable ring signature proposed in reference [23]. Notations utilized in this context are outlined in Table 1. Below delineates the linkable ring signature algorithm:

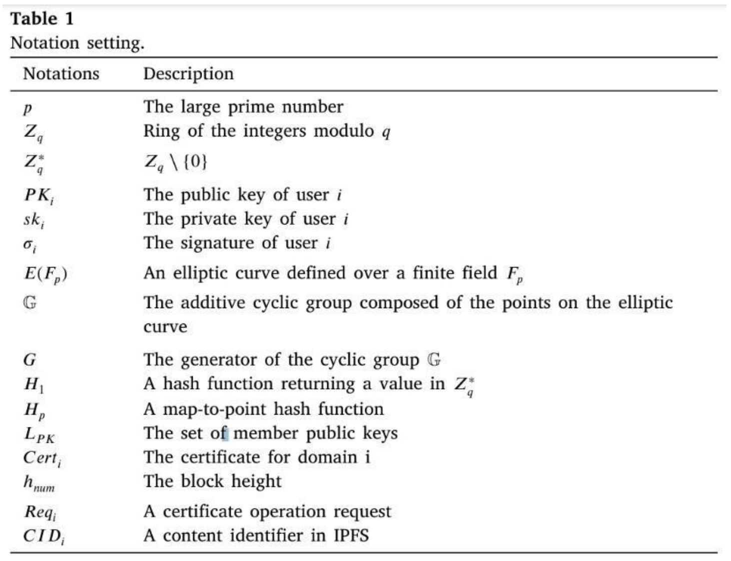


Table 1
Notation setting.

Notations	Description
p	The large prime number
Z_q	Ring of the integers modulo q
Z_q^*	$Z_q \setminus \{0\}$
PK_i	The public key of user i
sk_i	The private key of user i
σ_i	The signature of user i
$E(F_p)$	An elliptic curve defined over a finite field F_p
$\mathbb{G}$	The additive cyclic group composed of the points on the elliptic curve
G	The generator of the cyclic group $\mathbb{G}$
H_1	A hash function returning a value in Z_q^*
H_p	A map-to-point hash function
L_{PK}	The set of member public keys
$Cert_i$	The certificate for domain i
h_{num}	The block height
Req_i	A certificate operation request
CID_i	A content identifier in IPFS

1. System initialization algorithm: The algorithm inputs the se curity parameter λ and outputs the system parameters $params = \{p,Fp,E(Fp),\mathbb{G},G,q,H1,Hp\}$, where p is a large prime number, $E(Fp)$ is an elliptic curve defined over a finite field Fp, G represents the additive cyclic group composed of the points on the elliptic curve, G is the generator of the cyclic group G, the order is prime q, $H1 : \{0,1\}* \to Z* q$, Hp :$\{0,1\}*$
$\to$ G are two safe hash functions. 2. Key generation algorithm: The private key of user A is $skA \in R\ Z* q$,and the public key is $PKA = skA \cdot G$. 3. Linkable ring signature generation: The signer arbitrarily se lects $n - 1$user public keys and adds his own public key to form a group $L = \{PK1,PK2,\dots,PKn\}$, and the signer is the π- th $(1 \le \pi \le n)$ user who generates a linkable ring signature for the message m using the private key $sk\pi$ and the ring member public key L. The following algorithm generates a linkable ring signature.
System initialization is a crucial process in computing where a system is prepared for use. It involves a series of steps to bring the system from a powered-down or inactive state to an operational state where it can execute tasks or run applications.

(1) Compute the link tag $Q\pi = sk\pi \cdot Hp(PK\pi)$.
(2) Randomly select $k\pi \in Z* q$ and compute $c\pi{+}1 = H1(L,Q\pi,m,k\pi \cdot G,k\pi \cdot Hp(PK\pi))$ (1)
(3) For $i = \pi + 1,\dots,n,1,\dots,\pi - 1$, randomly generate $si \in Z* q$, and compute $Vi = si \cdot G{+}ci \cdot PKi$ $Wi = si \cdot Hp(PKi){+}ci \cdot Q\pi$ $ci{+}1 = H1(L,Q\pi,m,Vi,Wi)$ where $c1 = cn{+}1$.
(4) Compute $s\pi = (k\pi - c\pi \cdot sk\pi) \bmod q$ (5) Output the linkable ring signatures $\sigma L(m) = (Q\pi,c1,s1,\dots,sn)$ (2) (3) (4) (5) (6)
4. Linkable ring signature verification: To verify the received message m' and its linkable ring signature σ'
$L(m) = (Q'\pi,c'1,s'1,\dots,s'n)$, the
verifier performs the following
steps: (1) Check whether $c1'$, $s'i$ $(1 \le i \le n) \in Z* q$. Ifnot, the verification fails, otherwise go to the next
step. (2) For $i = 1,2,\dots,n$, calculate the followings in turn:
$V'i = s'i \cdot G + c'i \cdot PKi$ 4

linkable ring signature:
Linkable ring signatures are cryptographic constructs that allow a group of users to collectively sign a messagewhile preserving the anonymity of the signer within the group. Verification of such signatures involvesconfirming that the signature was generated by a memberof the specified group without revealing the identity of the actual signer. This is achieved by utilizing a mathematical structure where each signer's public key ismixed with others in the group, creating a "ring" of keys.To verify a signature, the verifier checks that the signature is valid with respect to the message and the public keys in the ring. However, ensuring non- plagiarism in the context of linkable ring signatures typically involves additional measures beyond standard verification, such as ensuring the uniqueness of the signed messages or employing external mechanisms to detect duplicate signatures.

5.3 Certificate revocation with linkability

In the certificate revocation process depicted in Figure 5, several key steps are followed. When changes occur in the personal information of the domain owner or if the private key of the certificate becomes compromised, the domain owner initiates a certificate revocation request directed towards the Ring CA. Similar to the certificate update procedure, the issuing CA's consistency is verified as a prerequisite.

Once the verification is successfully completed, the Ring CA employs linkable ring signatures to authenticate the certificate revocation operation. Subsequently, a miner integrates this transaction into a block, which is then broadcasted and added to the blockchain.

This process ensures that when critical changes or compromises occur, the necessary steps are taken to revoke the affected certificate while maintaining the integrity and security of the overall system.
In the blockchain context, certificates cannot be directly deleted due to the immutable nature of blockchain data. So, when a domain owner requests certificate revocation, the certificate's status is updated to "revoked," rather than being deleted outright. The process of generating a revoked certificate follows the same steps as certificate registration and updates.

During this process, the block height serves as a reference point for locating the block containing the certificate operation, aiding in the verification of its authenticity. Despite revocation, the certificate is retained in IPFS to maintain the integrity of the certificate audit process.

Subsequently, the Ring CA transmits the certificate to the Certificate Revocation List (CRL), where the serial number of the revoked certificate along with its revocation date are stored. This ensures that the revoked certificate is duly recorded and can be referenced when needed, despite no longer being valid for use.

Algorithm 1 Certificate Validation

```
Input: $CertA, CRLRingCA, h_{lastOpeNum}$.
Output: $b \in \{1, 0\}$. (The certificate is valid or not.)
procedure CErTVEr($CertA, CRLRingCA, h_{lastOpeNum}$)
$T \leftarrow Extract\ Cert$ ;
$TCurrent \leftarrow Extract\{Time\}$;
$\sigma RingCA \leftarrow Extract\{\ CertA\}$ ;
$LPK \leftarrow Extract\ Cert\{A$ ;

if $TCurrent > TNotAfter$ then return 0;
return 1;

    procedure REvChEcK($CertA, CRLRingCA$)
        $Serial \leftarrow Extract\ Cert$ ;
        if $SerialA \notin CRLRingCA$ then
            if $certRevRequest[SerialA] == 1$ then return 0;
            else return 1;
        else return 0;
```

5.4 Certificate validation with linkability

When a client initiates a TLS connection request to a target domain and receives a certificate from the domain, it needs to perform a series of checks to ensure the validity and authenticity of the certificate. The certificate validation process, outlined in Figure 6, is detailed below, along with the algorithm for certificate verification (Algorithm 1):

1. **Extract Ring CA Public Keys**: The client extracts the public keys of the Ring CA members from the certificate to verify the validity of the ring signature associated with the certificate.

2. **Check Validity Period**: The client verifies whether the certificate is within its validity period.

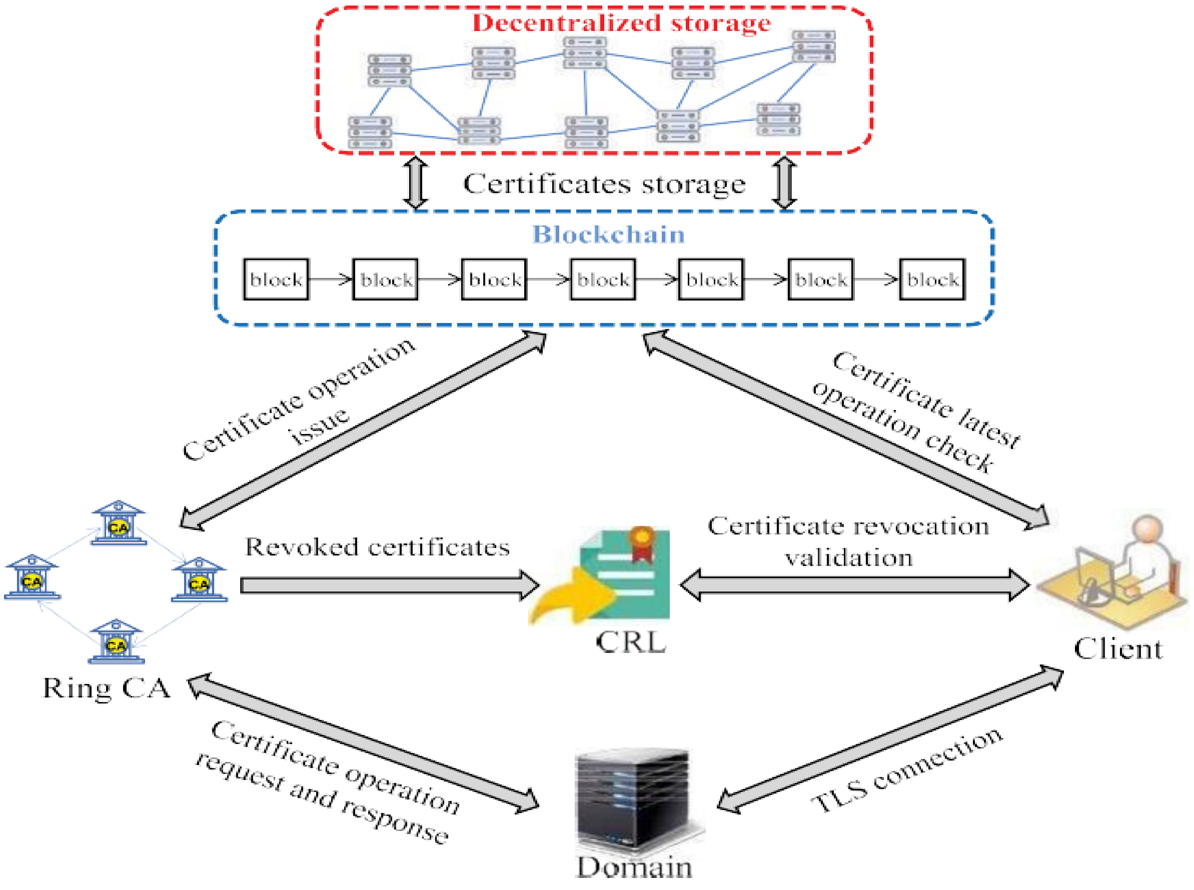
3. **Certificate Operation Existence Check**: This step involves two sub-steps:
   a. **Locate Certificate Operation Transaction**: The client retrieves the certificate operation transaction using the block height obtained from the certificate. It verifies whether the registration or update operation of the certificate exists in the blockchain.
   b. **Consistency of Issuing CA**: The client locates the linkable ring signature signed by the last Ring CA using the operation height field value in the certificate. It then compares the two ring signatures to determine whether they belong to the same issuing CA. These two sub-steps can be processed concurrently for efficiency.

4. **Request CRL from Ring CA**: The client requests a Certificate Revocation List (CRL) from the Ring CA.
After verifying the consistency of the issuing CA, the

```
if $RingVer(\sigma RingCA, Q\pi, CertA, LPK) \rightarrow 0$ then return 0;
if $BlockCheck(CertA, hnum) \rightarrow 0$ then return 0;
if $LinkVerify(CertA, LastCertA) \rightarrow 0$ then return 0;
if $RevCheck(CRLRingCA, CertA) \rightarrow 0$ then return 0;
```

Ring CA sends the corresponding CRL to the client. Theclient then verifies the authenticity and integrity of the CRL.

By following these steps, the client can ensure the validity and authenticity of the certificate received from the domain, thereby establishing a secure TLS connection.


Decentralized storage
Certificates storage
Blockchain
block
Certificate operation issue
Certificate latest operation check
Revoked certificates
Certificate revocation validation
CA
Ring CA
CRL
Client
Certificate operation request and response
TLS connection
Domain

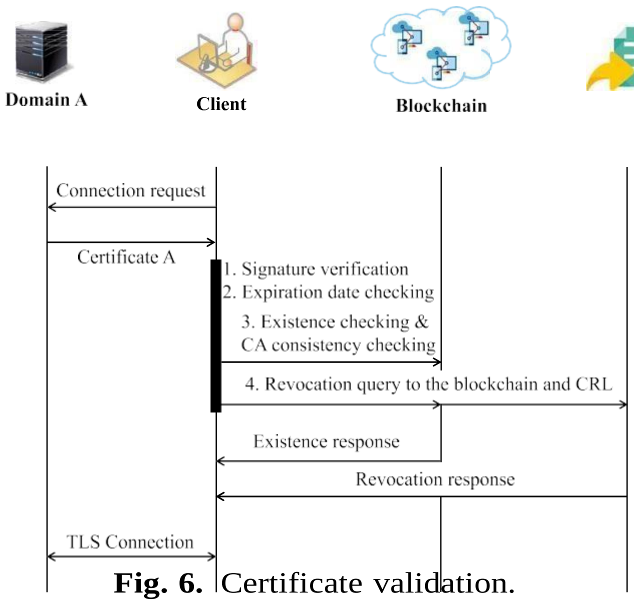


Fig. 6. Certificate validation.

5. **Check CRL for Serial Number**: The client examines whether the certificate's serial number is listed in the Certificate Revocation List (CRL). If the serial number is found in the CRL, indicating that the certificate has been revoked, the client terminates the communication.

6. **Verify Certificate Revocation Request**: If the certificate's serial number is not in the CRL, the client checks the global mapping table certRevRequest. This step aims to ensure that the certificate has not been revoked without updating the CRL. If the value returnedfrom certRevRequest is 0, indicating that there is no certificate revocation request for the certificate, the client terminates the communication.

These additional checks further enhance the security of the TLS connection by ensuring that certificates that have been revoked or flagged for revocation are not accepted for communication.

4.1. *Certificate storage*

In contrast to the traditional PKI system, which relies on centralized storage mechanisms, this work proposes employing IPFS (InterPlanetary File System) as a decentralized storage solution for certificate management. IPFS is a distributed web protocol that operates on a peer-to-peer network, facilitating the storage and retrieval of data off-chain. Each node in the

By leveraging IPFS, the certificate system mitigates therisks associated with centralized storage, such as single points of failure and susceptibility to Distributed Denial of Service (DDoS) attacks. In the IPFS architecture, datais distributed across multiple nodes, reducing reliance ona single centralized entity and enhancing systemresilience.

Moreover, IPFS incorporates an incentive mechanism where the first node to respond to a user's request for a certificate is rewarded. This incentivizes participation and contributes to the efficiency of the network.

It's worth noting that while IPFS offers advantages in decentralization and resilience, concerns regarding data integrity and privacy are addressed separately, drawing from existing technologies and methodologies. These aspects, while crucial, are beyond the scope of discussion in this paper.

5. Security analysis

asserts that within the LRS_PKI system, employing theLRS (Linkable Ring Signature) scheme to conceal the identity of the issuing Certificate Authority (CA) ensures anonymity, unforgeability, and linkability, provided that the LRS algorithm is correctly implemented.

Proof:

To demonstrate the anonymity, unforgeability, and linkability of the LRS scheme used in LRS_PKI, werely on the following arguments:

1. **Anonymity**: The anonymity property ensures that the identity of the signer remains confidential. By utilizing LRS to generate signatures, the signer's identity is concealed within a group of possible signers. This anonymity is preserved as long as the LRS algorithm is correctly implemented and the elliptic curve discrete logarithm problem (ECDLP) remains hard.

2. **Unforgeability**: Unforgeability guarantees that it is computationally infeasible for an adversary to produce a valid signature without possessing the private key. Similar to Theorem 1, the hardness of the ECDLP ensures that generating a valid signaturewithout proper authorization is infeasible, thus

3. **Linkability**: Linkability refers to the ability to link signatures generated by the same signer acrossdifferent transactions. In the context of LRS_PKI, linkability enables tracing the actions of a specific entity across multiple certificate operations.Assuming the hardness of the ECDLP and under therandom oracle model (ROM), the LRS scheme exhibits linkability, as demonstrated in Theorem 3.

By combining the properties of anonymity, unforgeability, and linkability, Theorem 4 asserts that the LRS scheme within LRS_PKI provides a robust framework for securing certificate operations whilepreserving the anonymity of the issuing CA. This holdstrue under the condition that the LRS algorithm is correctly implemented and the underlying cryptographic assumptions remain valid.

Proof:

The anonymity of the issuing CA within LRS_PKI is a direct consequence of the anonymity provided by the LRS algorithm. This anonymity ensures that any ring signature generated by a member of the Ring CA is statistically indistinguishable from others, making it improbable for an adversary to ascertain the true identity of the issuing CA. The probability of an adversary correctly identifying the issuing CAis bounded by 1/n, where n is the number of members in the Ring CA. Additionally, when an adversary attempts to impersonate an honest user torequest a certificate from the Ring CA, they are unable to deduce the true identity of the issuing CAthrough the link tag Qπ. This is because the domain owner interacts with the Ring CA rather than the actual issuing CA, thereby maintaining anonymity even from the domain owner, preventing adversaries from gaining insight into the issuing CA's identity. Thus, throughout the certificate operation process, the issuing CA remains anonymous.

The unforgeability of the linkable ring signature of the issuing CA relies on the unforgeability inherent in the LRS algorithm. Under the assumption of the hardness of the ECDLP, adversaries cannot forge avalid linkable ring signature for the Ring CA. In the first step of the certificate verification process, the Linkable Ring Signature Verification algorithm first step of the certificate verification process, the Linkable Ring Signature Verification algorithm determines the validity of the LRS. If the output is "accept," indicating a valid linkable ring signature, the verification succeeds; otherwise, it fails. Consequently, the utilization of LRS to conceal the issuing CA satisfies the criteria of unforgeability.

The linkability of the issuing CA stems from the linkability property of the LRS algorithm. Within LRS_PKI, linkability is employed for certificate revocation and verification processes, ensuring that these operations are consistently performed by the sameCA under the Ring CA. Leveraging the unforgeability of the LRS algorithm, the likelihood of adversaries forging a valid signature is negligible. By utilizing the Link algorithm, the certificate operation's consistency is determined. If the output is "linked," indicating satisfaction of linkability, the certificate operation is attributed to the same issuing CA. Otherwise, linkability is not achieved. Hence, the scheme employing LRS to obscure the issuing CA aligns with the definition of linkability.

Targeted Attack:
In traditional PKI systems, the hierarchical structure of CAs exposes them to targeted attacks. High-value targets, such as banks or government agencies, can be vulnerable because their issuing CAs are known. Attackers exploit this knowledge to target specific CAs, compromising their security. However, in LRS_PKI, the use of Ring CA protects the issuing CA's identity, ensuring anonymity as proven in Theorem 4. This anonymity makes it infeasible for attackers to deduce the issuing CA's identity with a probability greater than 1/n, where n is the number of members in the Ring CA. Consequently, LRS_PKI mitigates targeted attacks against the domain's issuing CA effectively.

Impersonation Attack:
In traditional PKI systems, attackers can forge certificates by impersonating legitimate web server administrators. However, LRS_PKI enhances transparency by recording all certificate operations on the blockchain. This transparency enables domain owners to detect impersonation attacks through self-auditing. By leveraging blockchain's immutability and transparency, LRS_PKI prevents attackers from impersonating legitimate web servers to obtain undetected fraudulent certificates.

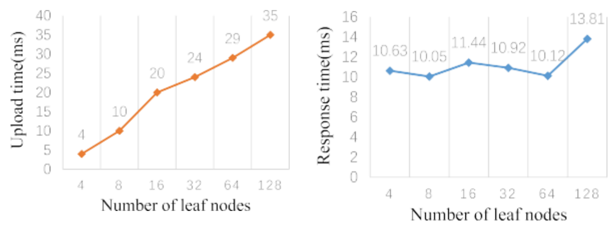
Rogue Certificates or Operations:

Table 2
Technical characteristics of the testbed's machines.

Node type	CPU architecture	CPU max speed	CPU operation mode	RAM	Operation system
Server	Core i5-10500	3.10 GHz	64 bits	16 GB	Ubuntu 20.04
CA	Core i5-11400H	2.70 GHz	64 bits	16 GB	Kali Linux 5.16
Client	Core i5-11400H	2.70 GHz	64 bits	16 GB	Kali Linux 5.16

Table 3
Average running time of every step for different ring size and signer position π.

$(size, \pi)$	(5, 2)	(10, 6)	(15, 10)	(20, 14)	(25, 18)
Sign time (s)	0.319	0.322	0.329	0.334	0.339
Verify time (s)	0.322	0.327	0.334	0.337	0.342
Link time (s)	0.327	0.330	0.334	0.338	0.344



(a) Upload time (b) Response time

Table 4
Processing time of every step in certificate validation.

CertVerify step	LRS verify	Check cert Date	Query block & CAcon_check	Check CRL & certRevRequest
Time (ms)	322	4	351	337

4. Experiment and evaluation

4.1. Implementation

We have developed a simplified LRS_PKI (Linkable Ring Signature Public Key Infrastructure) model on the Ethereum blockchain. Our prototype system utilizes JavaScript and Go (v1.18) programming languages. Specifically, we leverage Vue.js for constructing the user interface framework and implement private Ethereum using Truffle (v4.1.13). To simulate the blockchain nodes, we employ Ganache (v2.5.4), while communication between entities is facilitated through the Web3 API. IPFS integration is based on go-ipfs (v0.12.2).The core components, including the CA (Certificate Authority)entity and the linkable ring signature code, are implemented in Go. Cryptographic computations relyon the Go standard library crypto, employing the secp256k1 elliptic curve and SHA3-256 hash algorithm.In our prototype system, we deploy 5 CA nodes, 3 Client nodes, and 2 Server nodes, each running on separate machines. Table 2 provides abreakdown of the technical specifications of the machines utilized in our testbed.

4.1. Result analysis

In our certificate storage experiment, we aim to compare the size requirements between traditionalTLS certificates and those stored using the LRS_PKI model. Typically, a TLS certificate for a domain like google.com has a size of approximately 5 KB.

In the LRS_PKI model, rather than storing the entire certificate directly on the blockchain, we leverage IPFS for storage. Each certificate is stored in IPFS, andaccess is facilitated through a unique identifier. The blockchain stores only the certificate operation transactions, which are linked to the corresponding certificates stored in IPFS. These transactions are organized in the leaf nodes of a Merkle tree.For instance, the size of a certificate operation transactionin our experiment is approximately 0.5 KB, considering a ring size of 5 for the linkable ring signature. This contrasts with the size of traditional TLS certificates, providing a potentially more efficient approach to certificate management and storage in decentralized systems. In our experiments, we observed that the storage overhead in the blockchainis approximately 10 times greater than that of the LRS_PKI model. This led us to utilize IPFS for decentralized storage in LRS_PKI, addressing the increasing demand for identity authentications in the network.

Using the Wrk tool for performance testing in our IPFS experiment, we evaluated the upload andresponse time of certificate files. The average responsetime for obtaining a certificate was found to be about
11.16 ms, directly impacting user experience. Additionally, the certificate upload time was observedto increase with the increase of leaf nodes, but without significant impact on overall network transmission, making IPFS a viable option for certificate storage.

Comparatively, traditional PKI technologies often utilize the Online Certificate Status Protocol (OCSP) for certificate revocation checks. Stark et al. [25] observed that OCSP response times generally range between 100 ms and 600 ms, with an average responsetime of approximately 497.55 ms. In LRS_PKI, the average response time for checking certificate revocation status is about 337 ms, similar to OCSP.

Table 5
Comparison of different public key infrastructures based on function, storage and security.

	Basic function				Storage	Security		
	Registration	Updating	Revoking	Validation	Not stored in the blockchain	MITM attack resistance	CA consistence check	Privacy-preserving of issuing CA
IKP [7]	✓	×	×	✓	×	×	×	×
Xauth [21]	✓	✓	✓	✓	✓	✓	×	×
Cecoin [18]	✓	✓	✓	✓	×	✓	×	×
Authcoin [24]	✓	×	×	✓	×	×	×	×
Certcoin [6]	✓	✓	✓	✓	×	✓	×	×
CertLedger [8]	✓	✓	✓	✓	×	✓	×	×
ProofChain [20]	✓	✓	✓	✓	×	✓	×	×
LRS_PKI	✓	✓	✓	✓	✓	✓	✓	✓

Basic function

Storage

Security

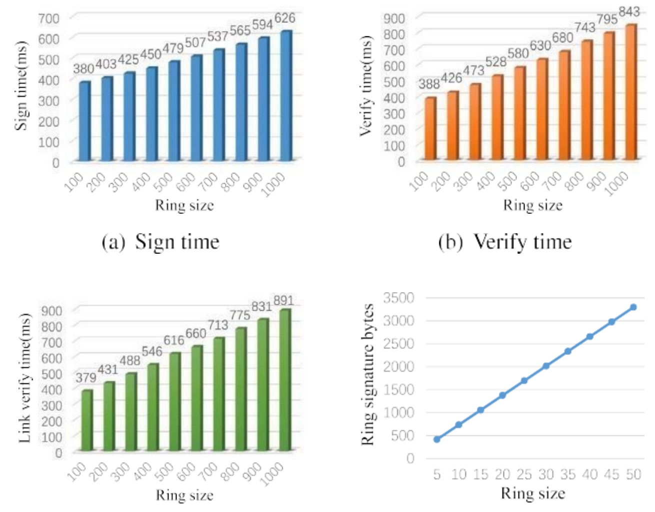


Fig. 8. Performance of linkable ring signature.

3

5
Ring Size

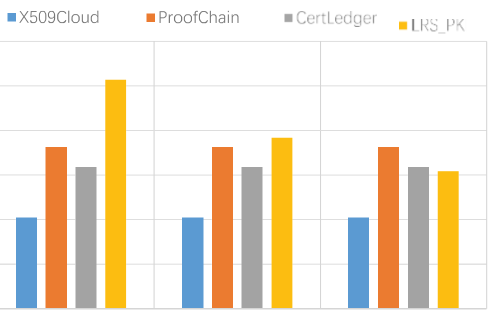
7

consistency of issuing CAs, LRS_PKI limits the authority of CAs to issue certificates arbitrarily, preventing misbehaving CAs from doing so.In comparing LRS_PKI with other PKI schemes (seeTable 5), it excels in both storage scalability and security. By storing certificates in IPFS and only recording certificate operations on the blockchain, LRS_PKI minimizes transaction size, thereby enhancing blockchain network transaction processing capability. Throughput analysis using the BlockSim toolkit indicates that LRS_PKI achieves acceptable performance, with a verification process taking approximately 1.014 s, suitable for high-security domain applications.

In the verification process of LRS_PKI, there are several steps involved. The client firstly verifies the

ring signatures in the certificate, a transaction that takes approximately 351 ms. Subsequently, the client checks the revocation status of the certificate by inspecting whether the Ring CA's Certificate Revocation List (CRL) contains the certificate's serial number. This step, which includes consistency verification of the issuing CA, takes around 337 ms. Incases where the certificate has been revoked but not

yet updated in the CRL, the client searches the global status of the certRevRequest table, adding another 337ms to the



verification process.

Fig. 9. Comparison of the number of transactions that can be accommodated in a block of the proposed system and the previously developed systems.

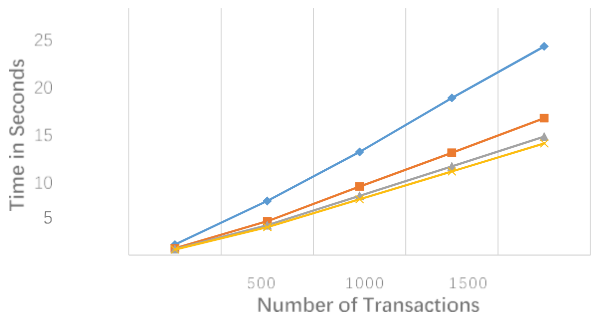


Fig. 10. Comparison of transactions speed of the proposed system and the previously developed systems.

The toolkit we utilized provides a controlled environment for simulating blockchain networks and executing tests or experiments to evaluate various performance parameters, suchas processing time, throughput, and computation time. To ensure fair andcomparable testing across all schemes, we standardized certain parameters. Specifically, we set the block size to 1 MB, the average block generation time to 12 s, and the block delay to 0.5 s.With these settings in place, we conducted tests comparing different schemes by inputting their respective certificate transaction sizes (with a ring size of 5). The results, illustrated in Fig. 10, demonstrate that LRS_PKI outperforms other schemes by effectively processing more transactions simultaneously.By optimizing transaction sizesand leveraging IPFS for certificate storage, LRS_PKI minimizes the burden on the blockchain network, allowing for efficient processing of transactions. This efficiency is crucial in scenarios where large volumes of certificate transactions need to be handled swiftly and securely.

5. Conclusion

To address the challenges encountered by Certificate Authorities (CAs) within the PKI system, we propose an innovative solution calledLinkable Ring Signature Public Key Infrastructure (LRS_PKI). This blockchain-based PKI leverages linkable ring signatures to enhance security significantly. Our approach involves recording certificate operations on the blockchain while utilizing IPFS for certificate storage, effectively separating the blockchain layer from the storage layer.

To mitigate targeted attacks on issuing CAs associated with specific certificates, we introducethe concept of a Ring CA, which safeguards the privacy of issuing CAs. Additionally, to counteract potential misbehavior by CAs, we implement checks for the consistency of issuing CAs during the certificate validation process. Security analysis and experimental results affirm the effectiveness of LRS_PKI, highlighting itssecurity, efficiency, and practicality. Moving forward, we plan to conduct large-scale applicability experiments to further evaluate LRS_PKI's performance when deployed across multiple nodes in various regions. This will provide a more accurate assessment of its capabilities and scalability in real-world scenarios.

References

[1] A. Abdul-Rahman, S. Hailes, A distributed trust model, in: Proceedings of the 1997 Workshop on New Security Paradigms, 1998, pp. 48–60, http://dx.doi.org/ 10.1145/283699.283739.

[2] D. Basin, C. Cremers, T.H.-J. Kim, A. Perrig, R. Sasse, P. Szalachowski, ARPKI: Attack resilient public-key infrastructure, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 382–393, http://dx.doi.org/10.1145/2660267.2660298.

[3] B. Laurie, Certificate transparency, Commun.ACM 57 (10) (2014) 40–46, http: //dx.doi.org/10.1145/2659897.

[4] T.H.-J. Kim, L.-S. Huang, A. Perrig, C. Jackson, V. Gligor, Accountable key infrastructure (AKI) a proposal for a public-key validation infrastructure, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 679–690, http://dx.doi.org/10.1145/2488388.2488448.

[5] M.D. Ryan, Enhanced certificate transparency and end-to-end encrypted mail, Cryptology ePrint Archive, Paper 2013/595, 2013, https://eprint.iacr.org/2013/ 595.

[6] C. Fromknecht, D. Velicanu, S. Yakoubov, A Decentralized Public Key Infrastruc- ture with Identity Retention, Cryptology ePrint Archive, Paper 2014/803, 2014, https://eprint.iacr.org/2014/803.

[7] S. Matsumoto, R.M. Reischuk, IKP: turning a PKI around with decentralized automated incentives, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 410–426,