Decentralized Public Key Infrastructure with Identity Management using Hyperledger Fabric

Amisha Sinha and Debanjan Sadhya
ABV-Indian Institute of Information Technology and Management Gwalior, India

Keywords: Public Key Infrastructure, Certificate Authority, Hyperledger Fabric, Decentralized Identifier.

Abstract: Public key infrastructure (PKI) is one of the most effective ways to protect confidential electronic data on the internet. In centralized PKIs, the identity is defined by trusted third parties, specifically the Certificate Authority (CA). However, the security of the end-users becomes jeopardized if the CA gets compromised. To tackle this problem, the decentralized nature of the system can be used to eliminate a single point of failure. However, the lack of real-time support, the block complexity, and strict implementation are drawbacks that burden the practicality of these approaches. This study tries to evaluate the Decentralized Public Key Infrastructure (DPKI) framework based on a permission-less model. The model itself is constructed over the decentralized identifier to manage the identity of users. We use the Hyperledger Fabric based blockchain network to create a hierarchy Certificate Authority, where each CA is a peer in a decentralized distributed network. Hence, each peer owns a separate database validated by the blockchain. We have evaluated the model efficacy in terms of the network latency and throughput, which were all found to be acceptable.

1 INTRODUCTION

Public Key Infrastructures (PKI) rely on digital signature technology to establish trust and security association parameters between entities. It allows entities to interoperate with authentication proofs using standardized digital certificates. Even though many applications use PKI technology for their security foundations, there are several concerns about their inherent design assumptions based on its centralized trust model. There have been instances where CAs have issued rogue certificates, or the CAs have been hacked to issue malicious certificates. Traditional PKIs also encounter several versions of the Man-in-the-Middle Attacks (Dacosta et al., 2012). Due to such issues, there remains a requirement for forgery-proof identity verification techniques (Salman et al., 2019).

Unlike the traditional approach, Decentralized Public Key Infrastructure (DPKI) ensures no single third party can compromise the integrity and security of the system as a whole. In blockchain-powered DPKIs, the new third parties become miners or validators (Isirova and Potii, 2018). The trust is established and maintained based on the consensus protocols. The miners or validators follow the protocol rules that would financially reward and punish these third parties, thereby preventing misbehavior in the blockchain and limiting their roles effectively.

1.1 Contributions

This study aims to build a secure DPKI that improves the existing infrastructures for user privacy and identity management. Specifically, our framework attempts to solve usability issues like identifying malicious certificates. These certificates can be consequently used to perform MITM attacks by revoking and blacklisting CAs so that no other domain owner can request certificates from it. We utilize the blockchain technology over the Hyperledger fabric to build a web of trust model. This model allows any entity on the network to verify attributes about any other entity through a trusted network.

The proposed model guarantees security by constantly validating intermediate signing CAs through other peers. This process enables the certificates to be produced in a legitimate and unaltered manner. Furthermore, this framework provides a hierarchy Certificate Authority, in which each CA owns a separate database validated by a decentralized-distributed blockchain. The entities will have unique DIDs, attach their public keys, and write them to the public ledger. Any person or organization that can discover these DIDs will be able to access the associated pub-

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Sinha, A. and Sadhya, D.
Decentralized Public Key Infrastructure with Identity Management using Hyperledger Fabric.
DOI: 10.5220/0011273000003283
In *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT 2022)*, pages 554-559
ISBN: 978-989-758-590-6; ISSN: 2184-7711

Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

lic keys for verification purposes. Hence, this work provides an alternative to the conventional CA-based identity verification model. The efficiency of the solution is measured by the variation in the latency of requests due to the blockchain transaction flow.

2 RELATED WORK

This section briefly discusses solutions based on DPKI for identity management. Noticeably, the need for DPKIs and how it can address the usability and security challenges that plague traditional PKIs has been extensively reported.

The study by (Salman et al., 2019) compares the existing DPKI approaches from different perspectives. Even though these systems exist and are opensource, the authors noted that only a few are utilized in real-world applications. This study also gives insights into the use of security services for current applications and highlights the state-of-the-art mechanisms that currently provide these services. This study is concluded by describing the associated challenges and discussing how the blockchain technology can resolve them. A public and decentralized PKI system termed *SS-DPKI* was proposed by (Chu et al., 2020). This solution enables a user to utilize public keys of multiple devices in a single ID.

A ChainPKI system was proposed by (Chiu et al., 2021), which still had some issues. For instance, the system uses an IP address as the ID for the relay nodes (intermediates). Though a user can change their IP address, it might still reveal some information relating to the user, like the geographic location. The system developed by (Garba et al., 2021) records a set of trusted CAs, each associated with a specific domain in the blockchain. A limitation of blockchain-based PKIs is storing a vast amount of data in the form of certificates, as well as large handshake message sizes.

The listed works have played an important role in developing meaningful solutions for replacing centralized PKIs. However, they do not claim satisfactory results for identity maintenance. Although few studies have reported promising results (Salman et al., 2019), they are quite complex and costly to be integrated into industrial domains. In this regard, our framework provides an efficient DPKI solution based on the Hyperledger fabric.

3 BACKGROUND

Now we detail some of the technological concepts related to our study. These preliminaries would facilitate in better understanding of our work.

3.1 Public Key Infrastructure

Centralized PKI relies on a trusted third party called Certificate Authority in the current standard X.509 (Burr et al., 1996). PKI uses a public and private key combination to encrypt and decrypt data. The X.509 certificate has a tree-like structure; it has a root node and is connected with various branches. The standard fields in X.509 certificates include the version number, algorithmic information, validity period, name of the CA and name of the subject to whom the certificate is issued to.

3.2 Decentralized Public Key Infrastructure

A decentralized Public-key Infrastructure is an approach that removes and separates the power of the CA. DPKI is a protocol for securely accessing decentralized consensus systems. DPKI changes the web's security model from single points of failure to decentralized consensus groups that create namespaces. DPKIs can provide a way to manage attributes for the web of trust. However, it does not specify what kind of identifiers should be used. Alternatively, it accepts different approaches that vary in terms of ease-of-use, permanence, uniqueness, security, and other properties. There are four types of identifiers:

- **Centralized**: Controlled by single entity
- **Federated**: Controlled by multiple entity
- **User-centric**: Individual/administrative control
- **Self-sovereign**: Individual control

The Self Sovereign Identity (SSI)[1] movement uses a blockchain to address several solution requirements. The most basic one is for the secure and authentic exchange of keys, which was not possible using PKIs.

3.3 Decentralized Identifiers

Decentralized identifiers (DID) are used to globally identify people and things over the internet. Previous global unique identifiers mostly include UUIDs and URNs. However, UUIDs are not globally resolvable and URNs require a centralized registration authority. Furthermore, none of these schemes is able to verify the ownership of the identifier in a cryptographic manner. In this regard, DIDs are the core component

[1] https://decentralized-id.com/literature/self-sovereign-identity/

of an entirely new layer of decentralized digital identity. The format of DID is presented in Figure 1[2].

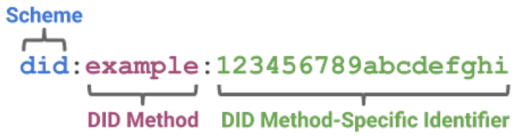
Scheme
did:example:123456789abcdefghi
DID Method
DID Method-Specific Identifier

Figure 1: The format of DID.

In Figure 1, DID Method is the namespace component identifier, and DID Method-Specific Identifier is the format of the method-specific identifier. In essence, DID is like a key-value database, where DID is the key and DID-document is the value. Interestingly, the DID document is a JSON-LD Data (a JSON-based serialization for Linked Data). The architecture of the DID is presented in Figure 2.

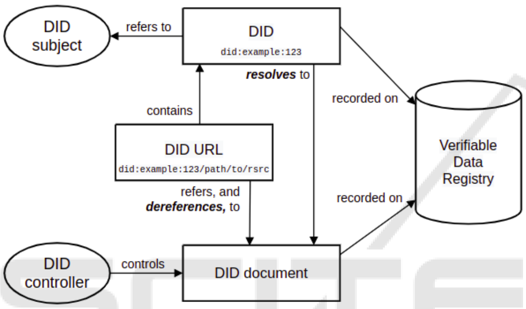

Figure 2: An overview of the DID architecture.

3.4 Blockchain

Blockchain is an emerging technology that demonstrates increased safety and confidentiality through decentralized, distributed and immutable characteristics. The blockchain is analogous to a digital ledger that stores transaction data in a distributed and peer-to-peer model. The data entered into the blockchain network is arranged in the form of blocks (Yaga et al., 2018). These blocks are immutable and hold the timestamp and the cryptographic hash of the previous block. This mechanism ensures a backward linkage between successive blocks. The hash generated by the cryptographic algorithm is foreordained. A slight modification in the data will change the hash values, which helps to resist any changes in the blockchain. Hence, this property allows having data in an open, public and immutable storage.

4 PROPOSED MODEL

The design of the proposed system can be divided into two parts. Initially, a DPKI framework based on a permission oriented collaborative consortium model using the Hyperledger Fabric (Androulaki et al., 2018) is developed. In this framework, all different DPKI entities and chaincode functions are created. Once a basic DPKI framework is setup, we integrate it with DID. We have decided to use the Hyperledger Fabric since it has a concept of channels and is open-source.

4.1 Chaincode Creation

The Hyperledger Fabric has a smart contract named *chaincode*, in which we can define our rules and functions for revocation and management of certificates. The chaincode can be used to create custom permissioned public or private blockchain. Chaincode extended attributes are constructed on the basis of the current PKI system. Interestingly, different features can be incorporated into different chaincodes as per the requirement. The most significant property required is the signature mechanism that is used for signing certificates. Noticeably, it can be different from the one used in transaction signing, nodes in the cooperative CA and others.

It is possible to insert consensus protocols in the Hyperledger Fabric (Vukolic, 2017). It also has a novel architecture for transactions, executing order, and validating. These functions are further explained as follows:

- **Execute:** The endorser executes the transaction and verifies its correctness.
- **Order:** Ordering of transactions through the pluggable consensus protocols.
- **Validate:** Validate transactions through endorsement policies before committing them into the ledger.

4.2 Identity Verification Model

DIDs are only the base layer of the decentralized identity infrastructure. Verifiable Credentials is the next higher layer where most of the value is unlocked. DIDs can identify various entities in the Verifiable Credentials ecosystem, such as issuers, holders, subjects and verifiers. Figure 3 is a visual representation of the identity credentials verification ecosystem. The functions of the various sub-modules are:

- **Issuer:** Issues verifiable credentials about a specific subject.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

[2] https://www.w3.org/TR/did-core

- **Holder:** Stores credentials on the behalf of a subject.
- **Verifier:** Requests a profile of the subject.
- **Identifier Registry:** Issues identifiers for subjects.

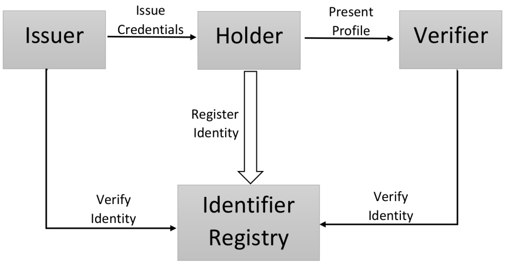

Figure 3: Flowchart of the identity verification model.

For the implementation of the DPKI, the main changes were made for the signing of DPKI content in the Endorsement System Chaincode component. Proxies act as the gateway between external clients and the DPKI network. A REST API gets exposed by the Proxy, which enables the clients in issuing and managing the certificates. The function `Cert()` that is used for creating the self signed Root certificate is presented as follows.

```
Cert() {
    var cert = forge.pki.createCertificate();
    cert.publicKey = this.keys.publicKey;
    cert.serialNumber = '01';
    cert.validity.notBefore = new Date();
    cert.validity.notAfter = new Date();
    cert.validity.notAfter.setFullYear(cert.
    validity.notBefore.getFullYear() + 1);
    cert.setSubject([{
        name: 'commonName',
        value: 'CA'
      }, {
        name: 'organizationName',
        value: "CertificateAuthority"
      }, {
        shortName: 'C',
        value: 'UK'
      }]);
      cert.setIssuer([{
        name: 'commonName',
        value: 'CA'
      }, {
        name: 'organizationName',
        value: "CertificateAuthority"
      }, {
        shortName: 'C',
        value: 'UK'
      }]);
      cert.sign(this.keys.privateKey);

      this.CA = {
          privateKey: forge.pki.
            privateKeyToPem(
                this.keys.
                privateKey
            ),
          certificate:
            forge.pki.
                certificateToPem(cert)
      };
      return this.CA;
}
```

4.3 Certificate Verification Model

In the developed system, it is possible to request signatures from the endorsing peers (seen as a cooperative CA) through a chaincode. A Certificate Signing Request (CSR) or X.509-v3 certificate is given as the input for this purpose. Figure 4 is a visual representation of this model.

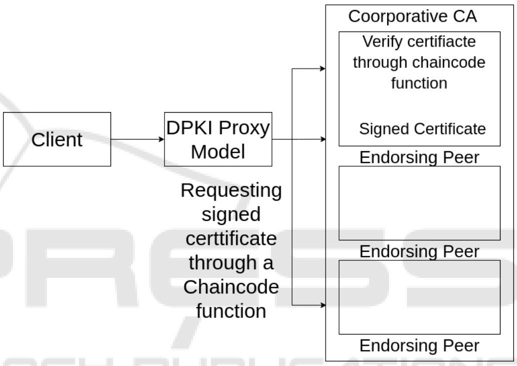
Coorporative CA
Verify certifiacte through chaincode function
Signed Certificate
Endorsing Peer
Endorsing Peer
Endorsing Peer
Client
DPKI Proxy Model
Requesting signed certtificate through a Chaincode function

Figure 4: The certificate verification signature model.

5 EXPERIMENTAL RESULTS

This section presents a set of experimental evaluations performed to support the developed framework. We specifically evaluate the framework performance over the latency and throughput of the network over various tasks. Noticeably, latency is defined as the response time per transaction, whereas throughput represents the number of successful transactions per second (Kuzlu et al., 2019).

5.1 Evaluation Environment

We used a local server to run the Hyperledger Fabric based blockchain network and Proxy. Each peer of this network is a docker container, and we have another dedicated container for its local ledger. Therefore, two docker containers are allotted to each peer. Furthermore, we have used a single proxy in the evaluations. In all assessments, four orderers were used

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

in the Byzantine Fault Tolerance (BFT) ordering service of the blockchain network (Sousa et al., 2018). The requests were made through the Proxy REST API with HTTP.

5.2 Performance Results

The primary aspect in all types of requests to the DPKI is the variation in the latency of request responses. The first test analyzes the latency of the certificate issuing process in the DPKI solution. As presented in Table 1, we can observe that the latency suffers the most when we increase the number of endorsers. The signature reconstruction process increases the latency of certificate signatures. Since we are working with a blockchain network that contains complex processes, the latency observed in the executed evaluations was expected. We can attribute these values while validating and committing transaction blocks in the Hyperledger Fabric transaction flow, mainly during ordering.

Table 1: Latency of the certificate issuing process in the DPKI solution.

No.of Endorsers	Latency (ms)
1	2440
2	2768
4	3489

Our second simulation investigates the latency in the certificate revocation process. Table 2 shows the results obtained with different numbers of endorser's peers while using the revocation process with a key size of 4096 bits. We can see that the number of endorser does not significantly increase the total latency of the requests. The latency in this case is caused due to the proxy running the chaincode algorithm that obtains the new Certificate revocation List (CRL) and signatures of the endorsers.

Table 2: Latency of the certificate revocation process the in DPKI solution.

No. of Endorsers	Latency (ms)
1	2220
2	3129
4	3564

For comparing the issuing process with the traditional issuing of X.509 certificates, we ran a docker container on the same machine where the blockchain network and Proxy were running. We did a simple configuration that enables requests for certificate revocation. In this regard, it should be noted that the time to enroll an entity is not added since it happens only in a centralized PKI. Table 3 demonstrates the results obtained while comparing our DPKI model with the traditional EJBCA PKI[3]. These results are expected since our model's characteristics and improvements have a performance trade-off. The resulting latency increases due to the ordering process, communication steps between peers in transactions, and others factors.

Table 3: Comparison of the latency in certificate issuing process with traditional PKIs.

Model Name	No. of Endorsers	Latency (ms)
EJBCA PKI	1	348
DPKI	1	2462
DPKI	4	3188

Next, we investigate the impact of the blockchain network throughput on the Hyperledger Fabric performance. We have utilized the Hyperledger Caliper Benchmark tool[4] for this purpose. Figure 5 shows the impact of the number of participants in the network on the network throughput for both Open Transactions (where one read and one write transactions were performed) and Query Transactions (where only one read transaction was performed). The results demonstrate that the query transactions have higher throughput than the open transactions since there is only one read operation in the query transaction.

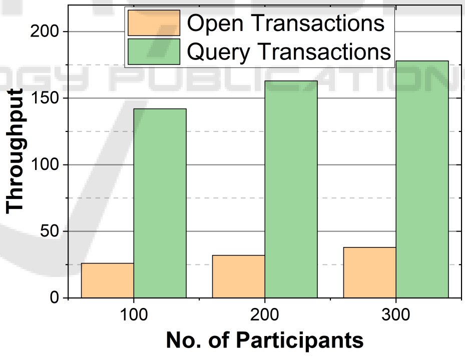

Figure 5: Impact of the number of participants on the network throughput.

Our final simulation analyzes the size of the issued certificates for different key sizes. Figure 6 presents the corresponding results for the RSA 2048 bit and RSA 4096 bit certificate signatures. Since we are working with multi-signatures, the issued X.509-v3 certificates contain all the signatures in extensions. Their size considerably increases due to this additional information. This entire process facilitates the validation of the certificate and its signatures by other entities.

[3] https://doc.primekey.com/ejbca/

[4] https//hyperledger.github.io/caliper/

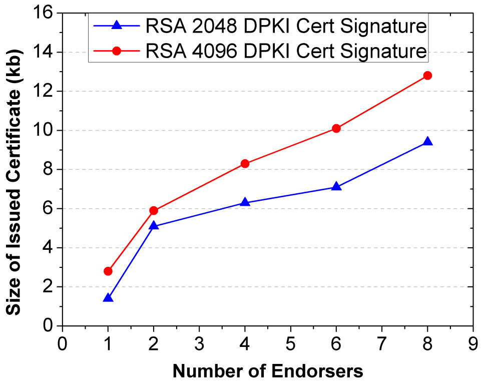

Figure 6: Variation of the size of issued certificates with the number of endorsers for different key sizes.

When the requests were made through the Proxy REST API with HTTP, a Round Trip Time of 32ms was noted with the dedicated server. In this scenario, the local-host client was used to call the DPKI functions in a local computer. In both the evaluations (i.e., latency comparison of certificate issuing and certificate revocation process), functions write data to the ledger of the blockchain. The currently stored CRL is obtained and returned so that the endorsers can sign it and insert their signatures in the response.

6 CONCLUSIONS AND FUTURE WORKS

The objective of this work is to provide an effective, robust and reliable user identity management PKI using a decentralized approach. This study merges the concepts of blockchain and decentralized storage structure with the ability of PKI to generate a certificate. Each of the peers on the network uses a unique intermediate certificate authority and validator. Any peer who joins the network will become an intermediate signing CA. The guarantee that the produced certificates are legitimate and unaltered is constantly validated by other peers in the network.

We can optimize the current implementation of our model to enhance its functioning. Specifically, we would emphasize on using a blockchain framework that uses DID and DPKI internally. In our current model, the certificates used in transactions are still generated by a utility function of the Hyperledger Fabric. Another possible improvement is to use the same generated key-pairs and certificates for both DPKI signatures and transaction signatures.

REFERENCES

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolic, M., Cocco, S. W., and Yellick, J. (2018). Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA. Association for Computing Machinery.

Burr, W. E., Nazario, N. A., and Polk, W. T. (1996). A proposed federal pki using x. 509 v3 certificates. *NIST Gaithersburg*.

Chiu, W.-Y., Meng, W., and Jensen, C. D. (2021). Chainpki-towards ethash-based decentralized pki with privacy enhancement. In *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–8. IEEE.

Chu, Y., Kim, J. M., Lee, Y., Shim, S., and Huh, J. (2020). Ss-dpki: Self-signed certificate based decentralized public key infrastructure for secure communication. In *2020 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6.

Dacosta, I., Ahamad, M., and Traynor, P. (2012). Trust no one else: Detecting mitm attacks against ssl/tls without third-parties. In *European symposium on research in computer security*, pages 199–216. Springer.

Garba, A., Chen, Z., Guan, Z., and Srivastava, G. (2021). Lightledger: A novel blockchain-based domain certificate authentication and validation scheme. *IEEE Transactions on Network Science and Engineering*.

Isirova, K. and Potii, O. (2018). Decentralized public key infrastructure development principles. In *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pages 305–310.

Kuzlu, M., Pipattanasomporn, M., Gurses, L., and Rahman, S. (2019). Performance analysis of a hyperledger fabric blockchain framework: Throughput, latency and scalability. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 536–540.

Salman, T., Zolanvari, M., Erbad, A., Jain, R., and Samaka, M. (2019). Security services using blockchains: A state of the art survey. *IEEE Communications Surveys Tutorials*, 21(1):858–880.

Sousa, J., Bessani, A., and Vukolic, M. (2018). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 51–58.

Vukolic, M. (2017). Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, page 3–7, New York, NY, USA. Association for Computing Machinery.

Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2018). Blockchain technology overview.