Shahjalal University of Science and Technology

Department of Computer Science and Engineering

Self-Sovereign Identity Management System

Md. Mesbah Uddin Waheed

Reg. No.: 2016331042

4th year, 1st Semester

Tapu Das

Reg. No.: 2016331106

4th year, 1st Semester

Department of Computer Science and Engineering

Supervisor

Dr. Md Sadek Ferdous

Assistant Professor

Department of Computer Science and Engineering

15th June, 2021

Self-Sovereign Identity Management System

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

Md. Mesbah Uddin Waheed
Reg. No.: 2016331042
4th year, 1st Semester

Tapu Das
Reg. No.: 2016331106
4th year, 1st Semester

Department of Computer Science and Engineering

Supervisor

Dr. Md Sadek Ferdous

Assistant Professor

Department of Computer Science and Engineering

15th June, 2021

Recommendation Letter from Thesis Supervisor

The thesis entitled *Self-Sovereign Identity Management System* submitted by the students

1. Md. Mesbah Uddin Waheed
2. Tapu Das

is under my supervision. I, hereby, agree that the thesis can be submitted for examination.

Signature of the Supervisor:

Name of the Supervisor: Dr. Md Sadek Ferdous

Date: 15th June, 2021

Certificate of Acceptance of the Thesis/Project

The thesis entitled *Self-Sovereign Identity Management System* submitted by the students

1. Md. Mesbah Uddin Waheed
2. Tapu Das

on 15th June, 2021, hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

Head of the Dept.	Chairman, Exam. Committee	Supervisor
Mohammad Abdullah Al Mumin	Dr. Farida Chowdhury	Dr. Md Sadek Ferdous
Professor and Head	Associate Professor	Assistant Professor
Department of Computer Science and Engineering	Department of Computer Science and Engineering	Department of Computer Science and Engineering

Abstract

In this thesis, we study different types of identity management systems to improve the security of those online platforms which are often used by users or organizations to exchange their secrets and valuable digital credentials. To this end, we examine the impedance or the failings of the current identity systems. Then we propose an existing identity system utilizing secure tools for exchanging credentials where there is no trust between the user and organizations or more.

Keywords: Blockchain, Identity Management System, Self-Sovereign Identity, Hyperledger Indy, Solid.

Acknowledgements

We would like to thank the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh, for supporting this research. We are very thankful to our honorable supervisor Dr. Md Sadek Ferdous for his worthy supports and directions. Without his supports and directions, this work could not be done.

Contents

List of Figures

Chapter 1

Introduction

When making services and resources accessible across computer networks, it is common to know who the users are and what services they have access to. Throughout the initial registration phase, identity management consists of issuing credentials and unique identifiers, and during the service operation phase, the authentication and management of access to services and assets by individual users on their credentials and identifiers [1].

Self-Sovereign Identity has become one of the most commonly used concepts in the Identity Management environment in recent years. With the explosion of online services over the last fifteen years or so, user and service identity management has taken centre stage and, in many respects, has become the basis upon which created many online services [2]. Individual control, security, and complete mobility are all provided by Self-Sovereign Identity, which is independent of any individual silo. It eliminates the three preceding stages' centralized external control features. The identity belongs to the person (or organization) who owns, controls, and governs it. In this sense, the person is their identity provider and no one else can claim to be "providing" their identity since it is fundamentally theirs. Any single organization does not influence an individual's digital life. Nobody can deprive his/her self-sovereign identity [3]. In order to address the difficulties of globally scoped digital identities, decentralization is a clear trend in identity management. IP addresses were given centrally by the Internet Assigned Numbers Authority at the start of the internet. The notion of Federated Identity [1], which facilitated the use of different online services with a single account, supplanted this kind of Centralized Identity.

Other service providers created the Liberty Alliance. The emergence of User-centric Identity, promoted by projects like OpenID [4], concentrating more on the ownership of personal data by the user and highlighting that identity management should be done independently from service consumption, was the next step to decentralization. However, single, dominant service providers such as Facebook or Google continue to produce the bulk of online identities. As a consequence, in contrast to the user, the identity is still owned by a firm. The service provider, for example, may always cancel the identity, and hence any linked services, without the users' intervention. The next phase might be self-sovereign identities, in which each individual develops and administers their own digital identities [3].

The following ten commandments use to describe self-sovereign identity:

- the existence of a person's identity independent of identity administrators or providers,
- the person's control over their digital identities,
- the person's full access to their data,
- systems and algorithms are transparent,
- digital identities are persistent,
- digital identities are portable,
- digital identities are interoperable, and
- the data economy is enforced.

1.1 Research Objectives

The main research objectives that we had are presented below:

- **RO1:** First and Foremost, we must ascertain all pertinent information about identity management systems in general. Our suggested approach, "Self-Sovereign Identity," will be aimed at resolving these challenges.
- **RO2:** We must amass information on current studies on self-sovereign identity management systems, their functioning methods, and their flaws, as well as determine why these flaws exist.

- **RO3:** We wish to develop an architecture for a self-sovereign identity management system based on IMS-related technologies that addresses all of the challenges inherent in present IMS efforts. If a danger cannot be minimized, we want to explain why.
- **RO4:** We also intended to verify the users using proof or work in our system.
- **RO5:** We want to construct the system in such a manner that the user may manage his or her identity via the usage of a Personal Authentication Device (PAD).
- **RO6:** We attempt to ascertain the systems' privacy and security threats.
- **RO7:** We are attempting to ascertain the requirements for minimizing threats.

1.2 Report Structure

We divide our thesis report into seven chapters, each including sections or subsections.

- In Chapter 1, we introduce our research topic and what the research objectives are.
- In Chapter 2, we discuss the background of our research topic. We also try to give a gentle introduction about related topics like identity management system, blockchain, hyperldger Indy, solid.
- In Chapter 3, we review related works regarding self-sovereign identity management system.
- In Chapter 4, we briefly discuss our research proposal regarding our thesis.This includes threat modelling, requirements analysis and architectural design.
- Chapter 5 discusses our intended implementation procedure regarding our thesis proposal and what we have done so far.
- In Chapter 6, we present future plans for our research.
- Chapter 7 incorporates the concluding summarization of our thesis.

Chapter 2

Background

This section discusses some essential concepts of our research. Details are described below:

2.1 Identity

In real life, identity refers to a person or a thing. Every real-life entity has its own unique identity that sets it apart from others. When used for identification purposes, an identity is composed of a collection of characteristics referred to as identifiers. Depending on the identity field, these characteristics may or may not be unique. They can have a variety of characteristics, including being transient or permanent, self-selected or issued by an authority, illustratable by humans or only by computers. In real life, an entity can use their unique identity to verify that they are who they say they are. Figure 2.1 represents the relationship between entities, their identities, and their characteristics.

2.2 Identity Management System

A system that manages an entity's unique identification and allows them to communicate with other people or organizations is known as an identity management system. With the rise of online communication and identities, identity management system has become critical for identifying a person's identity. Identification management systems have established certain identity models for better user experience in the online environment.

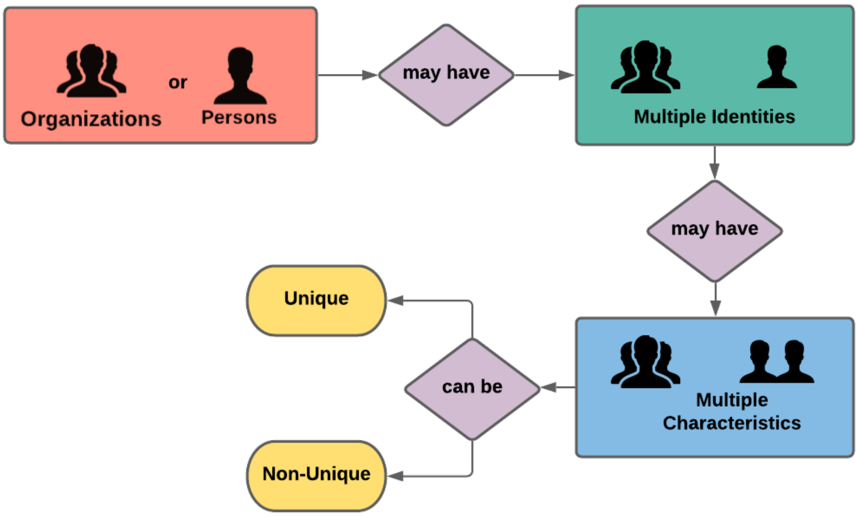

Figure 2.1: Entity, Identity, and Characteristic/Identifier Relationship

2.2.1 Isolated User Identity Model

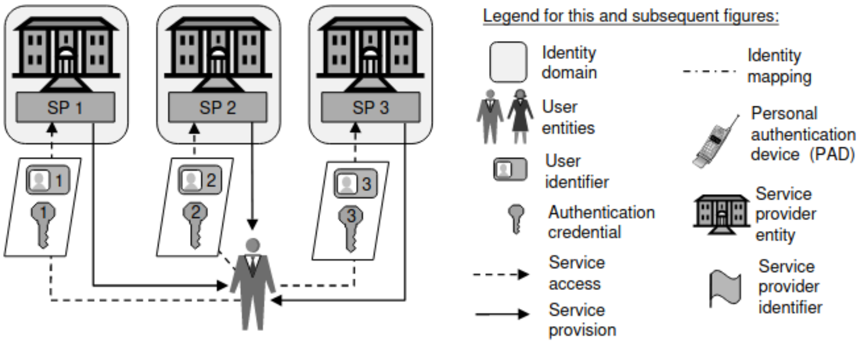

Figure 2.2: Isolated User Identity Model

Figure 2.2 depicts the isolated user identity model [1], in which the service provider assigns a unique identity to each user. Users must handle several unique IDs for multiple service providers. Although it is simple to manage for service providers, users must manage these unique identities across multiple service providers. It frequently causes users to forget their passwords or credentials.

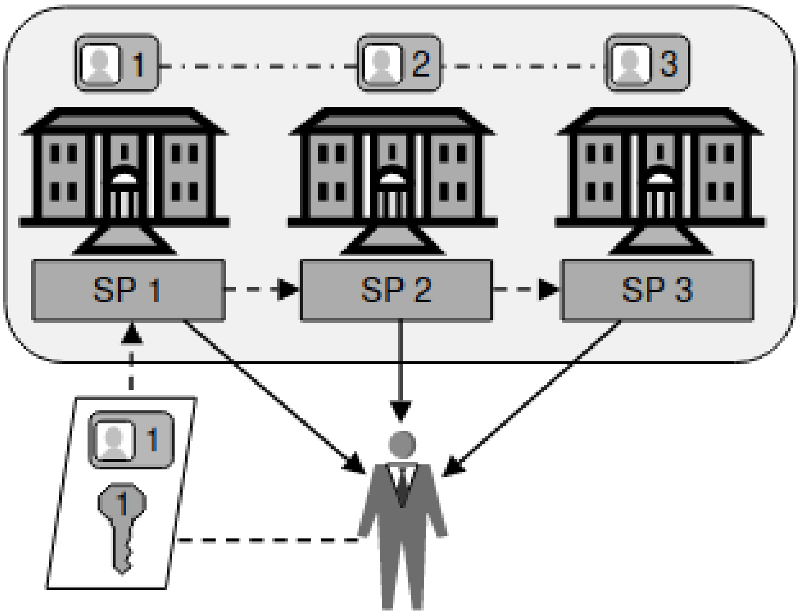

Figure 2.3: Federated User Identity Model

2.2.2 Federated User Identity Model

Identity federation is a method of connecting a user's identity across several security domains, each with its own identity management system. When two domains are federated, the user will be assigned to one of them and will be able to access resources within the other domain without having to utilize a separate login mechanism. Enterprises and their network users benefit from identity federation in terms of cost savings. Multiple firms, for example, will share a single application, resulting in cost savings and resource consolidation. Figure 2.3 [1] illustrates federated user identity model.

2.2.3 Self-Sovereign User Identity Model

In self-sovereign identity model in which the user acts as his own identity provider. The user has complete control over how his or her information is shared with others and how it is shared securely.

According to Christopher Allen, co-chair of the W3C's Credentials Community Group engaged with developing standards for decentralized identity and one of the first to coin the term "self-sovereign identity," the 10 principles of self-sovereign identity [5] are as follows:

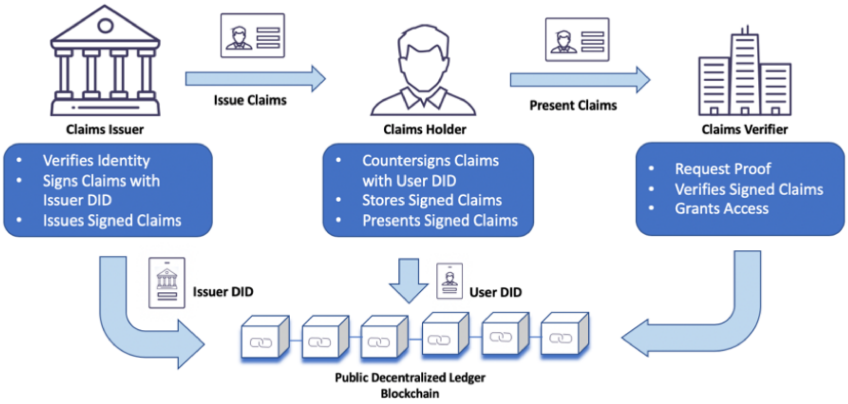

Figure 2.4: Self-Sovereign User Identity Model

1. Users must exist independently.
2. Users must have full control over their identities.
3. Users must be able to access their own data.
4. Transparency is required for systems and algorithms.
5. Identity must be long-lasting.
6. Identity-related information and services must be transportable.
7. Interoperability of identities is required.
8. Users must give their permission for their identity to be used.
9. Claims disclosure should be kept to a minimum.
10. Users' rights must be safeguarded.

The workflow for the self-sovereign identity model is depicted in Figure 2.4 [6]. Initially, users present their identity information to a trusted third party, such as the government or a reputable organization, for credential verification. Then, a trusted third party issues the user with a verified credential and assigns them a decentralized identifier (DID) [7]. DIDs are stored in a publicly accessible blockchain. Users then present that verified credential to service providers via DID in order to access their services. Service providers verifies the credential and grant users access to their services.

2.3 Blockchain

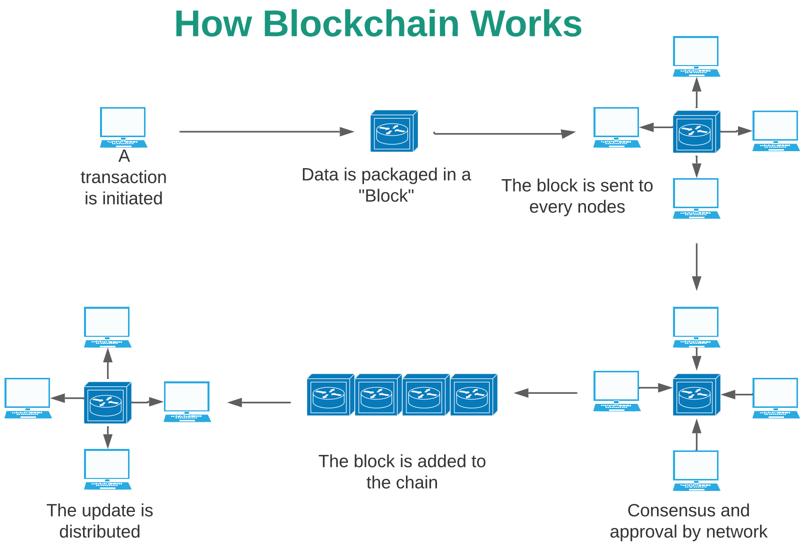
How Blockchain Works
A transaction is initiated
Data is packaged in a "Block"
The block is sent to every nodes
Consensus and approval by network
The block is added to the chain
The update is distributed

Figure 2.5: Blockchain

A blockchain is a growing list of records, called blocks, that are linked together using cryptography [8]. Blocks contain transactions, hash of the previous block and timestamp of the transaction. The integrity of a block's transactions is protected by this hash value. The integrity of the block is jeopardized by a simple modification in this hash value. The properties of blockchain are as follows.

Decentralised: Blockchain focuses on decentralisation, unlike client-server networks, which rely on a single server for authority. It means that the data on a blockchain is owned by no single party. It eliminates the possibility of a single point of failure.

Immutability: The data saved on the blockchain cannot be changed or altered in any way. Blockchain maintains the datas' integrity in this way.

Distributed: Every transaction in a blockchain is dispersed among all nodes. It improves the efficiency of data retrieval while also maintaining data availability.

Consensus mechanism: It is a technique for all participating nodes in a blockchian network to make decisions. When a transaction block is requested to be added to the blockchain, every participating node in the network uses the consensus method to verify the transaction. The transaction is stored in blockchain when it has been verified. The consensus method is critical to the networks' trustlessness. Participating nodes can use consensus techniques to verify the transaction without having to trust each other.

Enhanced data security: Every block in the blockchain stores a cryptographic hash of its own and preceding block contents. That is how each block is linked to the next. Even a minor change in data will dramatically alter the generated hash of a block, rendering subsequent blocks invalid. An attacker must perform a large number of computations to avoid identifying tampering in a block, which is tough.

2.4 Hyperledger Indy

Hyperledger indy provides a self-sovereign, secure, and efficient ecosystem for identity software, and libindy enables users to access it. Indy does not hold individuals accountable for privacy and disclosure decisions made by organizations that typically centralize identity. Hyperledger Indy enables a slew of innovative features, including connection contracts, repudiation, new payment workflows, property and document management, creative scrows, curator reputation, and integration with other cool technologies.

Indy makes use of open-source distributed ledger technology; distributed ledgers are a type of database that is provided cooperatively by a group of participants, rather than by a centralized administrator. Data is redundantly stored in multiple locations and is accumulated through the transactions of numerous machines. It is secured using robust, industry-standard cryptography. Its design incorporates excellent capabilities for key infrastructure management and cyber-security. As a result, we have a dependable, public source of truth that is independent of any single entity, resilient to system failure, hacker-resistant, and impervious to subversion by hostile entities.

The following are some of the key characteristics of Hyperledger Indy:

- **Public-permissioned** - Hyperledger Indy is a public permissioned blockchain. Because anybody can view data from the ledger without requiring permission, it is referred to as a public blockchain. On the other hand, in order to write data into a ledger, users must first validate it using a consensus method, which is why it is also known as permissioned blockchain.

- **Secured Cryptography** - Indy makes use of the Hyperledger Ursa [9] shared crypto library, which provides high-quality cryptographic primitives and key management operations.
- **Zero-Knowledge-proof** - Hyperledger Indy employs zero-knowledge proof [10], in which one user verifies and certifies another user's authenticity and integrity without knowing any additional information.
- **Decentralized Identifier(DID)** - Individual users are assigned a unique identifier by Indy, which is referred to as Decentralized Identifiers (DID) [7]. The key corresponds to a value referred to as the DID descriptor object (DDO). They work in tandem to create a complete DID [7] record. Users communicate with one another using the public and private keys associated with the DID [7] record, which is similar to the public key cryptographic method used in Blockchain.
- **Ledger** - Ledger maintains a users' public/private DID. To add to the ledger, indy uses a consensus mechanism called Redundant Byzantine Fault Tolerant. To write something on the ledger, consensus from a certain percentage of nodes is required.
- **Correlation-resistant** - Indy is identifier correlation-resistant, which means it will never connect two DIDs or have two similar identifications in the ledger.
- **Agent** - Indy defines an agent as software that interacts with other entities and stores a users' DID [7], keys or verified credentials. Hyperledger aries is used to build the agent software.
- **Trust Over IP(ToIP)** - ToIP [11] is a collection of protocols that are being developed to enable an additional layer of trust on the Internet. These protocols are embodied in Indy, Aries, and Ursa [9]. It encompasses self-sovereign identity in the sense that it encompasses identity.

Figure 2.6 illustrates the architecture of hyperledger indy:

□ **User** - Fully in control of his/her identity.

□ **Issuer** -Issues verified credentials to entities upon request after verification.

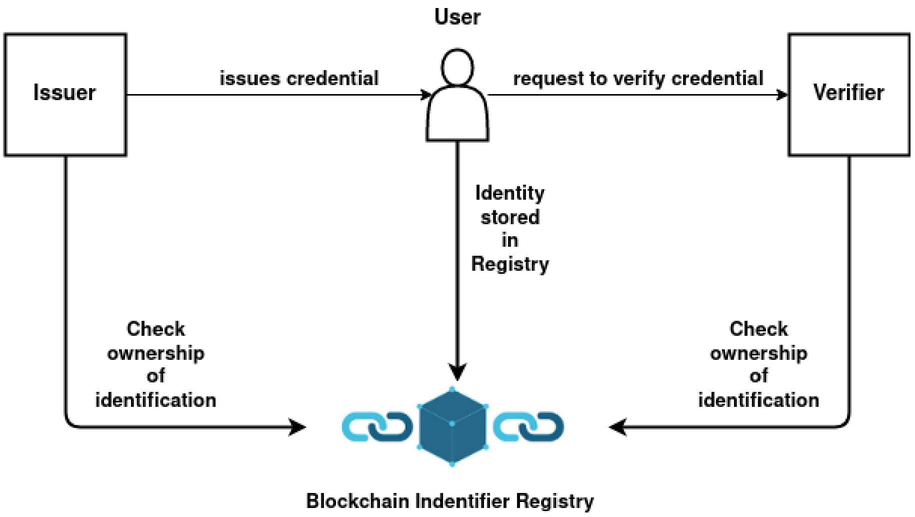
User
Issuer
issues credential
request to verify credential
Verifier
Identity stored in Registry
Check ownership of identification
Check ownership of identification
Blockchain Indentifier Registry

Figure 2.6: Hyperledger Indy Architecture

- □ **Verifier** - Examines and verifies the requestor's validity.
- □ **blockchain identifier registry** - Stores information about all users of digital identities.

2.5 Solid

Solid is a specification that lets people store their data securely in decentralized data stores called Pods [12]. Full form of solid is **SO**cial **LI**nked **D**ata. Solid has certain properties:

- **Interoperable ecosystem** - Different applications can access the same data within the interoperable Solid ecosystem, rather than requiring separate data silos for each application.
- **Data Availability** - Individuals may possess multiple solid pods. They can use either a third-party-hosted solid pod or their own self-hosted solid pod. The user's data is associated with their identity, not with the pod type.
- **Control over Data** - Users retain complete control over their data. Only they have the authority to decide what data to share and with whom. Furthermore, users may revoke access at any time.
- **Decoupled from applications** - Solid protects data by isolating it from specific applications. This enables users to share their data across multiple application platforms, thereby

facilitating interoperability.

- **Linked Data** - Solid stores structured data as linked data, which provides an interoperable format for interacting with the same data across multiple applications.

  Solid enables a more efficient management of user data. However, as a new technology, solid still has some limitations, such as the fact that it is only available for web applications.

Chapter 3

Related Works

The Identity Management System (IMS) is getting popular around the world. But there are many security and privacy-related issues in IMS like transparency, integrity, confidentiality, eligibility, robustness, coercion, end to end verifiability. There are many proposals and research work to confirm these issues. Identity management is typically viewed through the lens of service suppliers' factors and thus a service vendor's involvement in providing service data integrity. Conventional identity management systems are intended mainly for service providers to be affordable and versatile, but not for individuals. That frequently leads to a lack of accessibility. The current identity management methodology already has significant negative consequences, primarily for the consumer experience. As a result, the industry has suggested various identity management strategies to enhance the consumers. But these strategies allow users minimal respite at the expense of the comparatively high complexity of server systems. IMS introduces a Self-Sovereign Identity Management System(SSIMS) to overcome these security and privacy-related issues.

Audun Josang and Simon Pope suggested a cost-effective and user-scalable method while being compatible with standard identity management systems. A domain of identity is a domain where every identity is unique. A namespace is an environment with unique identifiers that permits a single connection between individualism and identifiers [1]. Authentication must take into account how the user is to handle IDs and credentials. They believe that it is simply impractical for consumers to maintain an inevitable expanding number of passwords and credentials via memorization or other archaic ways. One option is simply to allow individuals to save their credentials in a single tamper-proof hardware device that may be an intelligent card or another portable personal device.

The functionality of a PAD might be implemented into other devices such as mobile phones or personal digital assistants (PDA). The authors argue that this technique offers a wide range of opportunities to enhance user experience and to increase mutual authentication between users and service providers.

They propose: A PAD device may be created for many different authentication and access models that can be completely incorporated into the authentication process. This may be done by allowing the PAD to authenticate automatically on behalf of the user when connected to the customer platform. The benefits of the user-centered UI design are that the user has only to remember one credential. A master password secures a device that might be software or hardware. The PAD should be controlled by the users, not by the ID providers or credential suppliers. The latter would result in a proliferation of PADs that would defeat a single device.

Many analysts have converged on originating decentralized public key infrastructure (DPKI). Al-Bassam [13] introduced the SCPKI framework, a decentralized public key infrastructure framework that relies on Ethereum and InterPlanetary File System (IPFS) to implement identity management inclinations. On the other hand, facing the Self-Sovereign Identity Model, which depends on Hyperledger Indy and Solid, the SCPKI framework does not advocate all key management operations, including key restoration. Moreover, [13] helps the storage of user PII on Ethereum and IPFS networks. While this is proper for federal cases such as a university certificate, it is an uncertain procedure for personal identity attributes.

Faisca and Rogado [14], introduce a decentralized identity management pattern and authentication policy based on the Namecoin blockchain [15], WebID [16], and IPFS. They aim to utilize the Namecoin blockchain to authenticate publicly key and to store WebID profile addresses. Each WebID lesson connects to an IFPS stored WebID Profile Document. Unlike [14], Hyperledger Indy and Solid relies on DID and an authorized DLT created for identification usage facts.

Many projects are aimed at addressing contemporary DLT and SSI identity management provocations [17]. Among the most prominent schemes are Blockcert [18], ShoCard [19], OpenBadges [20], Civic [21], and UPort [22].

OpenBadges [20], offers the capability to get verified digital badges that indicate accomplishments and credentials. The badges are recorded in JSON-LD format and include disputes and profiles, which indicate user apps and user identification characteristics in part. OpenBadge does not directly operate on a DLT to store identification, as opposed to the Self-Sovereign Identity Model. In addition, OpenBadge use revolves upon use cases linked to the management of performance bills.

Blockcert [18], is another open design based on SSI, which accelerates the development of apps that may issue and verify authentic credentials and accomplishment certificates. Public blockchain Bitcoin holds documents. In contrast to Blockcert, the autonomous identity model depends on an identity management case particularly designed DLT with help for data reduction features like Zero-Knowledge Proof not yet exposed in Blockcert.

ShoCard [19], offers a tool to create and validate identity applications by adding a digital fingerprint of Bitcoin and other blockchains' user identification properties. ShoCard users may exchange their identification applications with application verifiers such as airport control stations using ShoCard applications. A vital aspect of the ShoCard ecosystem is ShoCard servers. Shocard is not open-source.

uPort [22], is an SSI architecture in which public keys like IPFS are stored by a decentralized system. Like Indy, it enables a technique of identifying unique identifiers for each user and enabling users to share their identification characteristics. Unlike Indy, uPort relies on Ethereum, a public blockchain without authorization. Since uPort relies on Ethereum, it has intelligent contracts and intelligent contract IDs for users. Compared with the design and implementation of Ethereum intelligent contracts, there are costs.

Civic [21], is an identity management assistance based upon Ethereum with an integrated incentive tool. Civic supports identification information development, consumption and verification. Civic servers and Civic applications are critical Civic ecosystem features. Civic is not an open-source system, as opposed to Indy.

Depending on the use cases and needs, the answer is which platform is most suited for establishing the Self-Sovereign Identity management system. The dependence of Hyperledger Indy on an open source DLT, which is designed specifically for identity management uses, its application of new technologies such as verifiable credentials, DID and anonymous credentials, its independence from proprietary software, and its support from the community, however, make Indy a suitable choice for our self-sovereign identity management systems. We propose to utilize Solid-Pod as our server and storage to hold WebID profiles.

Chapter 4

Our Proposal

To ensure self-sovereign identity, we have built an ecosystem around hyperledger indy and solid decentralized storage. According to the concept of self-sovereign identity and our proposal, the users are the sole controllers of their data. Users can store their raw data in their own self-hosted solid pod on their device. They can contain multiple solid pods and have multiple distinct identity that is transparent to service providers. To begin, users must register using their solid pod information via a trusted third party, such as the government or a reputable organization, via the hyperledger indy interface. After a trusted third party verifies a users' data, the user is issued a verified credential that is stored on their solid pod and public blockchain Identifier registry. It is worth noting that a trusted third party can only have one public DID [7], which enables verifiers to verify their authentication and authorization capabilities for issuing verified credentials. Users, on the other hand, can generate multiple DIDs for communication with various service providers. Now, using the issued verified credentials, users can request to enroll in different service providers by providing their DIDs via the indy agent wallet. Users can communicate with service providers via multiple secure private channels for each type of communication. As a result, their communication is protected. Users' verified credentials are verified by service providers by looking up their DIDs in the blockchain registry. Each DID is associated with a DID descriptor object (DDO) that contains information about the entity's ownership of its decentralized identity. Providers validate that data using Hyperledger Ursa [9], a shared cryptography library integrated with hyperledger indy. After verification, a secure communication channel is established between the service provider and user. Thus they can communicate with each other in a secure manner.

This chapter begins with a threat modeling section and concludes with a requirements analysis for mitigating the threats. Following that, we propose an architecture and conduct a detailed analysis of it.

4.1 Threat Modeling

Threat modeling enables us to identify and mitigate threats to our designed system. We investigated STRIDE [23] as a threat modeling tool. The following details are included:

- **T1-Spoofing identity** - It includes unauthorized access to users' identity data. An attacker can put on some adversary act e.g. modifying resources, illegal access to the system with the users' identity.

- **T2-Tampering threats** - It involves an attacker modifying the data or the system(e.g. By adding or removing some fundamental elements).in addition, an attacker can intercept or even can alter the data while being transmitted. This affects the integrity of the data.

- **T3-Repudiation threats** - This is a threat where attackers refute their malicious activity using the inability or vulnerabilities (e.g. inability to track prohibited actions, no-log analysis capability) of the system.

- **T4-Information Disclosure** - Leakage of identity information to people apart from owners who are not authorized to access during a process with the verifier or issuer.

- **T5-Denial of Service** - It disables or disrupts the system or storage that is used to access or preserve the valuable identity information.

- **T6-Elevation of privilege** - It involves an attacker trying to attain the higher privilege(e.g. changing the authorization level) of the system with malicious intent and disrupt the integrity of the system.

4.2 Requirement Analysis

In this section, we proposed a set of functional, security, and privacy requirements that would be used to mitigate the system's vulnerabilities.

4.2.1 Functional Requirements

Functional requirements concern the system's functionality. They ensure that the systems' primary functions are carried out accurately. The following are the functional requirements:

- **F1** - The functionalities of self-sovereign identity must be implemented.
- **F2** - The system should have a logging method to check each transaction.
- **F3** - The system will have a robust data storage system.
- **F4** - The system should have a modular design to ensure scalability.

4.2.2 Security Requirements

Security requirements are necessary to safeguard the system and its users and data against malicious attackers. The following security requirements apply to identity management systems:

- **S1** - Data must be stored safely in secure storage.
- **S2** - The system must have a context detection method for minimal release of data to secure privacy.
- **S3** - Protected private channels must be used for secure communication between users, which can tackle T4.
- **S4** - The system must go through the encryption mechanism to secure the user attributes.
- **S5** - Cryptographic mechanisms(public key infrastructure) should be used to ensure non-repudiation, confidentiality, and integrity, which will tackle T3.
- **S6** - Secure Identification method must be used for user authentication processes. S5, along with S1 and S3, can tackle T1.

- **S7** - Appropriate endorsement policy must be used to prevent illicit activities of user credentials. S6, along with S3 and S4, can tackle T2.
- **S8** - The system must be made Byzantine Fault-tolerant even if T6 occurs in some machine, S7, and S4 (PKI) can tackle T6.

4.2.3 Privacy Requirements

The following are the privacy requirements:

- **P1** - The system will use privacy protection techniques to support user anonymity and pseudonym to ensure that users are unlikable at different SPs.
- **P2** - The system will follow the data minimization mechanism to ensure that only the required data is transferred, stored, and processed at an SP. P2, S4, and S6 will mitigate T5.
- **P3** - The system will ensure that the user must be empowered. It will allow a user to provide explicit consent before any data is released to an SP, and also, the user can choose specific attributes for a particular SP.

4.3 Proposed Architecture

The proposed systems' architecture is depicted in Figure 4.1. We can see all of the system's components and their interactions. The following sections detail the functions of each component:

1. **Indy app:** Users or organizations interact with trusted third parties and service providers via an indy agent app.This piece of software also stores public private key pair and DIDs.
2. **Solid pod:** A decentralized storage system that stores both raw user or enterprise data and verified credentials in an interoperable format. It gives the user complete control over the data.
3. **Trusted Third Party/Issuer**: After verifying users' and organizations' identity information, issuers issue them verified credentials.

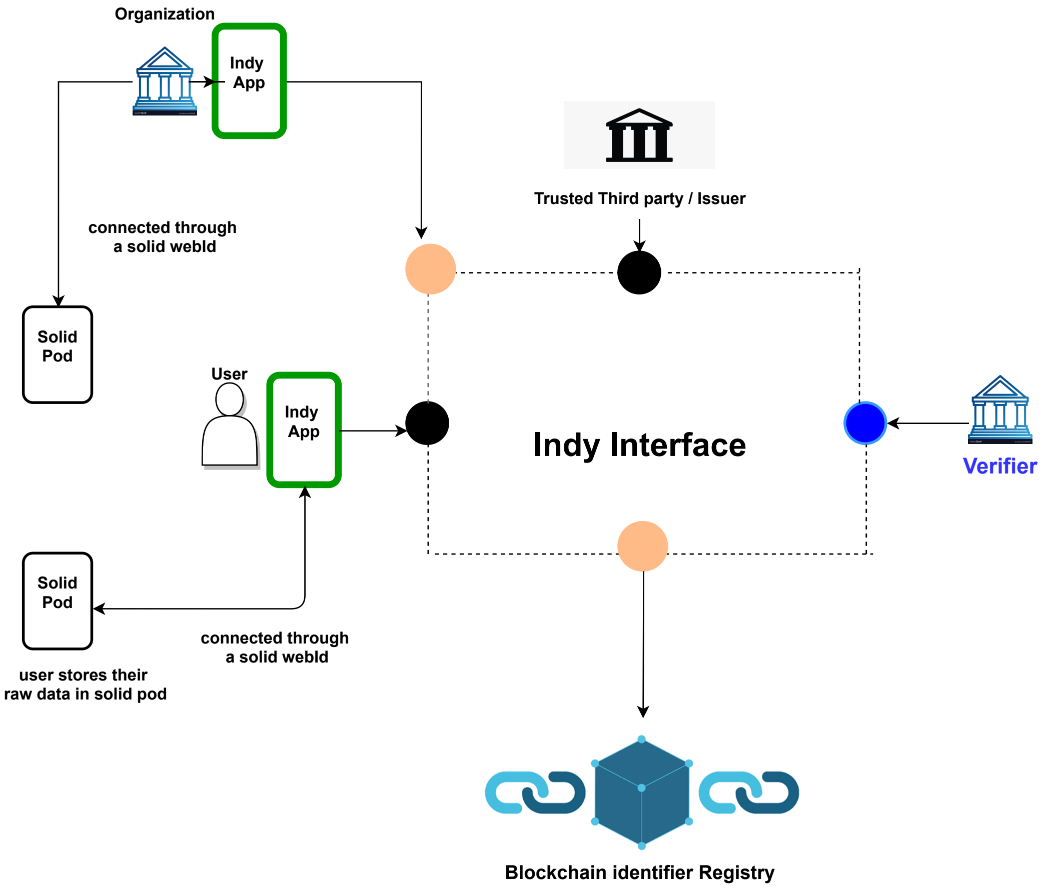
Organization
Indy App
Trusted Third party / Issuer
connected through a solid webId
Solid Pod
User
Indy App
Indy Interface
Verifier
Solid Pod
user stores their raw data in solid pod
connected through a solid webId
Blockchain identifier Registry

Figure 4.1: Proposed Architecture

4. **Verifier**: Users and organizations provide the verifier with their verified credentials in order to establish communication.

5. **Indy interface:** It is a user interface through which entities securely communicate with one another.

6. **Blockchain identifier registry**: This component stores public DIDs, schemas [24], credential definitions [24], and if necessary, a revocation registry [25].

Chapter 5

Implementation

In recent times, work on putting in place a fully functional self-sovereign identity management system is still on going. All of the associated works give a theoretical perspective on how to put self-sovereign identity into practice. Some systems are attempting to establish self-sovereignty via public blockchains such as Ethereum, but these systems still have privacy and security concerns, as well as challenges with user experience. Hyperledger indy, unlike any other decentralized platform, has built-in capability for implementing a self-sovereign identification system. Indy includes all of the essential protocols and tools for securing user identification information and communication. Furthermore, the solid pod creates an interoperable environment in which data can be shared across platforms and individuals have complete control over their data.

We have completed our research objectives RO1, RO2, RO3, RO6, RO7 so far. We conducted extensive research on the various components of hyperledger indy.

We are also working on deploying a self-hosted solid pod on users' devices. We did, however, developed a demo notepad web app[1] that communicates with the solid server to store the data of users. For testing purposes, we utilized an identity provider such as SolidCommunity [26]. This demonstration application enables users to store notes on their solid pod. Users have complete control over who has access to their personal information.

[1] https://github.com/Tapu106/Solid-Notepad

Chapter 6

Future Work

The initial purpose of our research was to find out the traditional identity management system in different domains and improve existing standards with cutting-edge technologies. We choose the self-sovereign identity management system as our domain of interest. There is still no current standard identity management system that offers perfect user identity security and privacy. Most IMS models only target some of the threats faced by consumers. We wanted to develop a system that addresses all the issues as perfectly as possible while mitigating some of the shortcomings of traditional IMS models. But our self-sovereign identity management system is not completed yet. We will implement the Hyperledger Indy to secure user identification information and communication and use the Solid pod server to control his identity.

Once the development phase is completed, we will run performance testing and do extensive testing on the whole system to find any vulnerability to which sensitive information may be leaked. We will also run a comparative analysis on the platform to evaluate its weakness and strength against other IMS models.

Currently, there is no privacy-preserving framework that can be used or modified quickly depending on use cases. Therefore, the main goal of our research is to create a general-purpose Blockchain Identity Register-based Hyperledger Indy interface integrated with solid pod that anyone can use to develop privacy-preserving applications and protocols very quickly. We intend to make such a prototype by the end of this year.

Chapter 7

Conclusion

In the era of digital identities, it is difficult for users to manage this volume of identity information. Additionally, there are concerns about the data's security and privacy. The self-sovereign identity management system's primary concern is the security and privacy of users' identity information. As a result, we propose a self-sovereign identity management system in which users can communicate securely with other entities via Hyperledger Indy and have their identity information stored on the Solid pod server. Solid gives users complete control over their data, ensuring data security and privacy.

References

[1] A. Jøsang and S. Pope, "User centric identity management," in *AusCERT Asia Pacific information technology security conference*. Citeseer, 2005, p. 77, [Online; accessed 10-March-2020].

[2] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103 059–103 079, 2019, [Online; accessed 25-April-2020].

[3] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, vol. 29, no. 2016, 2016, [Online; accessed 18-July-2020].

[4] M. Ferdous, "User-controlled identity management systems using mobile devices," Ph.D. dissertation, University of Glasgow, 2015, [Online; accessed 13-April-2020].

[5] C. Allen, "The path to self-sovereign identity," April 2016, [Online; accessed 15-April-2020]. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html

[6] "Self-sovereign identity," September 2020, [Online; accessed 12-June-2021]. [Online]. Available: https://www.cyberark.com/wp-content/uploads/2020/09/Rajesh-Self-Sovereign-image1.png

[7] Wikipedia contributors, "Decentralized identifiers — Wikipedia, the free encyclopedia," 2021, [Online; accessed 12-June-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Decentralized_identifiers&oldid=1026311010

[8] Wikipedia Contributors, “Blockchain Wikipedia, the free encyclopedia,” 2021, [Online; accessed 13-June-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Blockchain&oldid=1027822655

[9] “Hyperledger ursa-shared crypto library,” November 2018, [Online; accessed 13-June-2020]. [Online]. Available: https://www.hyperledger.org/use/ursa

[10] Wikipedia contributors, “Zero-knowledge proof — Wikipedia, the free encyclopedia,” 2021, [Online; accessed 15-June-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Zero-knowledge_proof&oldid=1027338920

[11] “Trust over ip,” 2020, [Online; accessed 15-January-2020]. [Online]. Available: https://trustoverip.org/

[12] “Solid,” 2017, [Online; accessed 12-September-2020]. [Online]. Available: https://solidproject.org/about

[13] M. Al-Bassam, “Scpki: A smart contract-based pki and identity system,” in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, 2017, pp. 35–40, [Online; accessed 15-July-2020].

[14] J. G. Faísca and J. Q. Rogado, “Decentralized semantic identity,” in *Proceedings of the 12th International Conference on Semantic Systems*, 2016, pp. 177–180, [Online; accessed 20-July-2020].

[15] “Namecoin.” April 2001, [Online; accessed 12-Apr-2021]. [Online]. Available: https://namecoin.org/

[16] “Webid specifications.” 2010, [Online; accessed 12-Apr-2018]. [Online]. Available: https://www.w3.org/2005/Incubator/webid/spec/

[17] P. Dunphy and F. A. Petitcolas, “A first look at identity management schemes on the blockchain,” *IEEE Security & Privacy*, vol. 16, no. 4, pp. 20–29, 2018, [Online; accessed 12-Apr-2018].

[18] Blockcerts, “Blockchain credentials. blockcerts,” [Online; accessed 12-Apr-2020]. [Online]. Available: http://blockcerts.org/

[19] “Secure enterprise identity authentication | shocard.” 2015, [Online; accessed 12-Apr-2020]. [Online]. Available: https://shocard.com/

[20] “Open badges homepage.” 2010, [Online; accessed 12-Apr-2020]. [Online]. Available: https://openbadges.org/

[21] “Civic identity verification | secure & protect identities,civic,” [Online; accessed 12-Apr-2020]. [Online]. Available: https://www.civic.com/

[22] “uPort.me,” 2016, [Online; accessed 12-Apr-2020]. [Online]. Available: https://www.uport.me/

[23] A. Shostack, *Threat Modeling: Designing for Security*, 1st ed. Wiley Publishing, 2014, [Online; accessed 12-November-2020].

[24] “Credential schema, hyperledger indy,” March 2019, [Online; accessed 20-November-2020]. [Online]. Available: https://w3c-ccg.github.io/vc-json-schemas/

[25] “Hyperledger indy,” March 2019, [Online; accessed 20-July-2020]. [Online]. Available: https://hyperledger-indy.readthedocs.io/projects/hipe/en/latest/text/0011-cred-revocation/README.html

[26] “prototype implementation of a solid server,” 2020, [Online; accessed 15-September-2020]. [Online]. Available: https://solidcommunity.net/