Offline Verification of SSI

by

Mashukur Rahman
21101316
Md. Zayadul Huq
21141010

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
Jun 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mashukur Rahman	Md. Zayadul Huq
21101316	21141010

Approval

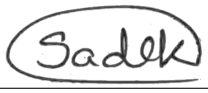
The thesis/project titled "Offline Verification of SSI" submitted by

1. Mashukur Rahman (21101316)
2. Md. Zayadul Huq (21141010)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 25, 2024.

Examining Committee:

Supervisor:
(Member)



Dr. Md Sadek Ferdous

Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

When the internet was first created no one thought about an identity layer because during the initial days, the main objective of the internet was to interchange data among people. No one was concerned about who was sharing the data with them. But as the number of users increased rapidly, suddenly the need for identity was recognised as it became apparent that anyone could impersonate someone else over the internet. To mitigate this issue, providers designed different identity models and adopted those models to provide identities to their users so that they could be independently identified. However, these models have numerous issues. For example the models do not solve how the validation will happen if both parties are offline, one model uses time-stamp but it does not solve the problem of real-time offline validation. Unfortunately, managing the identities provided by the providers was hard and users have limited control over their identity data. To mitigate these issues, the concept of Self-sovereign Identity (SSI) has been introduced. Through this approach users maintain their digital identity in a decentralised manner and have full control over it. Self sovereign identity has three parties. The user, the issuer and the verifier. The issuer issues a verifiable credential for the user and the verifier can verify it with the verifiable presentation that the user provides. To facilitate this interaction SSI utilises blockchain to store important metadata, SSI documents and public keys and it is important that the verifier and the blockchain is online during the verification process. The problem arises when the verifier and holder does not have access to the internet. In this research we will explore how SSI verification can be facilitated even when the verifier and holder is offline.

Keywords: SSI, Identity, Verifier, Holder, Issuer, Offline verification, Blockchain.

Table of Contents

Chapter 1

Introduction

1.1 Introduction

During the initial days of the internet, the creators did not think about one of its most important layers, the identity layer[1]. It was initially neglected as the main purpose of the internet was sharing information among systems. People were not very much concerned about who was sharing the information with them as the number of internet users was very less. But as time passed by, the number grew exponentially. With the increasing number of internet users, the need to identify them also increased. In order to do so, different service providers started providing identities to the users using different models. Two major models are the centralized identity model (SILO) and the federated identity model [5]. The centralized identity model is the most common identity model which is still used by various service providers. In this model the user registers an account with the service. The service provider actually lends credentials to the users which identify them and the user does not exist without the credentials. But those credentials still belong to the service provider. The users have very little control over the data which is associated with those credentials. If the users delete their accounts, they lose their access to the service but their data still remains to the service provider. This leads to data abuse by the service providers. Also remembering multiple credentials of multiple services is another hassle for the user. To solve these problems the federated identity model was introduced. This model uses an intermediary identity provider between the service provider and the user. This intermediary identity provider provides credentials to the users and these credentials can be used for multiple service providers. So the user needs to remember only one set of credentials for all the services and all of those services do not have the user data. Though this approach reduces the data abuse, the intermediary identity provider still has access to the user data. Also the identities provided by service providers and intermediary identity providers are not actually identities, rather identifiers. With these identifiers, the provider can identify users individually. This serves the organizational purpose but does not help the users. To mitigate all these issues, the idea of a decentralized identity system emerged. This decentralized identity system is called Self Sovereign Identity (SSI). In a general SSI model there are three parties. The credential issuer, the prover and the verifier. The issuer issues a verifiable credential about a prover and the verifier verifies the credential. The SSI model is not account based. Instead it works like real life identity. In real life an identity issuer issues identity for users, The

users keep this identity in their wallet and when a verifier asks for the identity for verification purpose, the users simply just shows the relevant identity and not the associated data. In the SSI model the user establishes a peer to peer connection with the verifier. No one owns this connection (unlike the credentials in centralized and federated identity models). The connection is preserved as long as both the user and the verifier wants it.
But just like all other technologies, SSI has some limitations too. The major limitations that SSI is currently facing are lack of a SSI infrastructure, decentralized key management and offline verification [4].

1.2 Motivation

This research aims to provide the solution of verifying credentials in scenarios where both the holder and the verifier are offline - they do not have any internet access but the cellular network is available. This research will solve the problem of verifying credentials in areas where there is no stable internet network like rural areas, land borders. The research aims to decrease the dependency on the internet and give users an alternate way of verifying their credentials, hence eliminating the single-point failure of the internet in terms of verifying credentials.

1.3 Problem Statement

In order to experience the advantages of SSI we need an infrastructure. When all the public and private organizations will have an SSI enabled infrastructure only then we will be able to take advantage of SSI. Many organizations have already started to build SSI enabled infrastructure so we can expect this problem to be solved very soon. Decentralized key management was another major problem of SSI during the early days. SSI fully depends on public-private key cryptography where the users hold their private keys in their digital wallet. So losing the private key means losing the identity of the user. But with the advancement of cryptocurrencies, managing the private keys are becoming easier now. But the major problem is still offline verification. SSI uses a digital system to store and access the identities of users. In order to access the digital system we need internet access. But there are scenarios where we need to verify the credentials when the holder or the verifier do not have internet access. A classic example of this is when a police officer tries to verify a driving license in a remote location where the internet is not accessible. So there is a huge demand for a system where the users can verify the crednetials without having internet connection. Unfortunately, there has not been enough research to solve this problem. Though very few researchers have proposed a few approaches, there is still no proper solution to verify credentials in offline scenarios.

1.4 Research Objectives

The objectives of this research are presented below:

RO1. Assessing the existing methods: Our first and foremost objective is to assess the existing offline verification methods and identify gaps that can be addressed to

improve the process.

RO2. Developing an SSI infrastructure: We will propose a novel approach for the offline verification of Self-Sovereign Identity (SSI). To implement this idea, we will develop an SSI infrastructure that allows users to verify credentials offline.

RO3. Analyzing threats: We will develop a threat model to analyze potential threats to our infrastructure.

RO4. Analyzing functional requirements: We will analyze the functional requirements of the infrastructure to ensure a smooth and user-friendly experience.

RO5. Analyzing security requirements: Based on the threat model, we will derive and specify security requirements to enhance the security of the infrastructure.

RO6. Evaluation: We will evaluate the overall performance of our infrastructure to demonstrate that the proposed concept is both feasible and practical.

1.5 Report structure

The structure of this report is as follows: In chapter 2 we explore the background of SSI and offline verification of SSI. We explore the history as well as some terminologies required to understand the paper well. In chapter 3 we reviewed some research works that are related to our research. We assessed the papers in order to find a concrete problem and a possible solution. In chapter 4 we present our project proposal. We also show the steps needed to complete the research. In chapter 5 We show a threat model, functional requirements, non-functional requirements and security requirements needed to build the infrastructure. In chapter 6 we have included the architecture and the protocol flow. In chapter 7 we have thoroughly discussed the implementation of the whole system. In chapter 8 we discussed about which research objectives, functional requirements and security requirements were able to achieve. Chapter 9 concludes our entire research.

Chapter 2

Background

With the evaluation of the internet, digital identity has become a very crucial and important part of our life. A person communicates with others on the internet and holds responsibility through their digital identity. Also , with the advancement of technologies, many services including both virtual and physical are linked with our digital identity. To avail those services and access to much of our important data , we must have the control of our digital identity. So , the importance of having the control of one's digital identity into their own hands for the users is very clear. Now let us emphasize how the idea and need of self-sovereign-identity (SSI) developed. Currently users get their digital identity from Identity Providers (IdP) who are mostly private corporations and government organizations. This centralized system of digital identity began to raise concern as the central authority can disable users access to their digital identity any time. Therefore, a decentralized system of digital identity where users will be in control of their own identity and will not need to depend on a centralized identity provider came to motion. Hence, the concept of self-sovereign identity began to circulate. One of the first hints of self-sovereign identity was offered by Pretty Good Privacy (PGP) in 1991 [2]. "Web of Trust" where the peers could act as both introducer and validators of their digital identity . The advancement of digital identity can be divided into four phases: Centralized identity, Federated identity , user-centric identity and self-sovereign identity [2].
Centralized authorities like Internet Assigned Numbers Authority (IANA), Internet Corporation for Assigned Names and Numbers (ICANN) used to play both roles of issuer and authenticator in the early days of the internet [2]. Here the user had no power over their digital identity. Also , as the internet grows , so did the number of websites users need to create different accounts for different websites that caused balkanization of identity of the same user.
To tackle the issue of balkanization of identity, private organizations came up with a solution called federated identity where the users can use the same identity in multiple websites to authenticate themselves. The first initiative of this federated identity was the Microsoft Passport in 1999 [2]. Nonetheless, it made Microsoft the centralized authority same as the traditional centralized authority like IANA , ICANN.
With focus on consent and interoperability, user centric identity allows users to share identity among different services. For example, users can select a trusted website and use their OpenID to login to other websites. Here users could experience some of the advantages of self-sovereign identity. But again, the provider could take away

OpenID anytime. So, again users identity is in the hands of a central authority. Though the user-centric identity model failed to provide users a self-sovereign identity, it was a step toward self-sovereign identity as it respected some level of user concept and interoperability, turning centralized identity to interoperable federated identity. The concept of self-sovereign identity began to gain popularity in 2012 after Patric Deegan started Open Mustard Seed. Open Mustard Seed is an open source framework that makes users' data decentralized and lets users control their identity [2].

2.1 Blockchain

A blockchain is a decentralize distributed ledger that records transaction or data. Unlike traditional centralized databases blockchain uses a peer-to-peer network where the participant nodes collectively maintain the ledger. Blockchain is immutable, means transaction or data once recorded can not be altered without the consensus of other participating nodes. Blockchain also eliminates the single point of failure limitation as multiple nodes participate in the network, and if some nodes go out of service, other participating node can serve the network. Blockchain uses consensus mechanism for the participating nodes to come to an agreement. Consensus mechanisms are predefined protocol that facilitates nodes in a decentralized network to come to an agreement.

2.2 Identity Model

In digital world to access services we need digital identity. There are different types of identity models. The identity models vary in terms of how they are issued to the user, control of central authority, user control, data privacy, security.

2.2.1 Centralized Identity Model

In centralized identity model, a central authority remains in full control of user identities. The central authority can revoke a user's identity any moment they like. The users identities are stored in the central authorities server, which raises a big concern of users data privacy and security. Also in this model identity provided by one organization can not be used in other organizations system.

2.2.2 Federated Identity Model

In this model, identity provided by one organization can be use across different organizations system. For example, google account can be used to access other organizations system. Here other organizations relies on a trusted identity provider to authenticate the user.

2.3 Self-Sovereign Identity

Self-Sovereign Identity (SSI) is a decentralized identity model where users remain in full control of their digital identities. Here users can decide which data they want

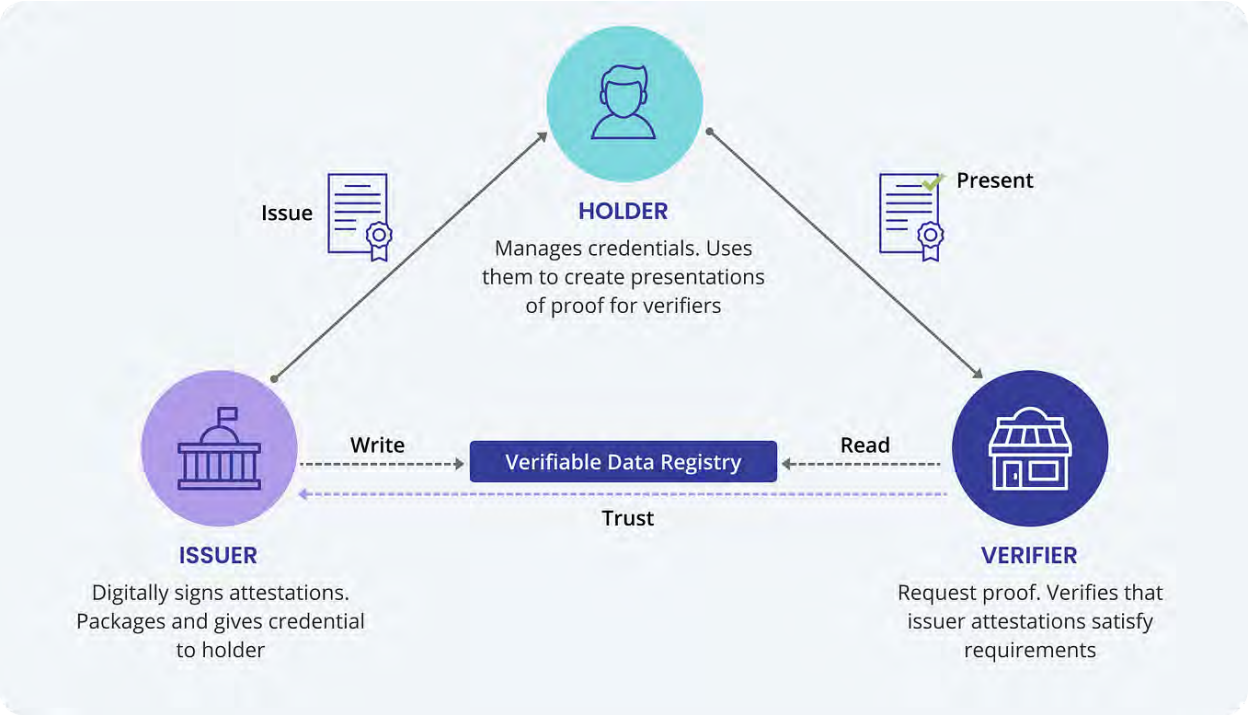


Figure 2.1: The SSI Trust Triangle

to share. The identity holder can decide with whom h/she wants to share their identity, credentials.
There are 3 main actors of this model. Issuer, holder, and verifier. Issuer is responsible for issuing verifiable credentials to the holder. Holder, the owner of the credentials manages the credentials in a digital wallet. The verifier is the entity that validates a credential claimed by the holder. The SSI model works by establishing trust among the 3 actors, called the trust-triangle.

2.4 Advantages of SSI

SSI is crucial for finding the missing identity layer of the internet because it brings a shift in the control. In the centralized and federated identity model, the control over the data associated with an user was on the hand of the issuer or verifier. But in the SSI model it shifts to the user. In SSI, the users have full control over the data that is associated with their identity because when they try to avail any service they only share their identity from their digital wallet. They can choose what information they want to share with the service provider. That way service providers can not abuse the user data as they do not have access to it

2.5 Terminologies

Now let us define some terminologies that we will be using throughout this paper.

Self-sovereign-identity: The concept of self-sovereign identity (SSI) is that the

user will be in complete control of their digital identity , and do not need to rely on any third-party identity provider. The identity will be interportable. Users can communicate with each other using that identity. Government and other trusted entities can issue verifiable credentials and Identity holders can use those credentials to verify themself. Also, the identity holder can decide how much information they will disclose [6].

Identifier: An attribute that is unique and can identify an entity separately [10].

Credential: Credential serves like a proof of a person's claim , information , identity , qualification. In our context of self-sovereign-identity by credential, we mean the digital version of this proof [5].

Issuer: Issuer is the source from where the credential is being provided. Issuer is a trusted entity on which everyone can rely for information and proof. Normally government organizations, NGO's , educational institutions, Financial institutions, international organizations etc. act as an issuer.

Holder: Holder refers to the individual to whom the digital identity belongs. The verifiable credentials of that digital identity describes the holder. Holder can use those credentials to verify himself and as proof of his claims.

Verifier: Any entity that seeks proof of claimed attributes , credentials from the holder to establish a trusted relation with the holder is a verifier. Government organizations, private organizations , and individuals can act as a verifier. Verifier seeks verifiable credentials from the holder , though holders can decide the level of information they want to share with the verifier [6].

Decentralized Identity: Decentralized Identity (DID) is implementing decentralized technologies like blockchain for identity management. The aim of decentralized identity is to eliminate the need of central authority and let users have full ownership of their identity [10].

Identity Provider: Identity Provider (IdP) provides users online identity and manages those identities. Identity Providers authenticates users based on their identity. Identity provider acts as a central authority of the online identity and owns the control of those identities .

Decentralized System: An architecture where the control is distributed among the participating nodes/entities. Decentralized systems democratize the power of decision making where the majority of the participating nodes/entities need to come to a consensus to perform an action.

Service Provider: An entity that is responsible for online services [5].

Identity: An accumulation of all the attributes , characteristics that can uniquely define an individual or an entity. The attributes by which we define an individual's identity may vary from person to person [12].

Identity Management: The process of management of online identities. Technologies with different policies facilitate users to enjoy online services through their online identity [5].

Blockchain: Blockchain is a record of a transaction that is distributed and the records are stored in a specific structure. The participating entities (nodes) in a blockchain follow a certain specific set of rules. The nodes form a chain as they are linked with each other following some specific rules.

Bluetooth Low Energy: Bluetooth Low Energy (BLE) is a version of Bluetooth technology with focus to have low energy consumption . This wireless technology is mainly used for peer-to-peer short ranged communication without the need of any external infrastructure (e.g: internet) [13].

Distributed Ledger Technology: Distributed ledger technology (DLT) is a concept to store records , transactions across a decentralized network. The records, transaction of the ledger will be immutable [7].

Chapter 3

Literature Review

Abraham et al. [6] came up with a possible solution to revoke and verify identities while being offline using a distributed ledger (DL). According to their generic architecture the user (i.e the prover) will first create a key pair (public-private key) and a Decentralised Identity (DID). The DID is generated using the public key. Then the user registers the DID in the SSI system. Users can verify their DID to the nodes of the SSI system using their private key. Upon registering and verifying the DID, the user requests for verifiable credentials from the service provider or issuer. The issuer also verifies the DID of the user and generates a verifiable credential (VC) with the attributes of the user, DID of both the user and the issuer and signs the credential. The VC remains on the SSI system. The issuer or the provider can request for revocation of the credential and then the credential is put in the revocation list. If the user needs to use the VC offline, then S/he can request an attestation in the SSI system. The system first checks if the VC is in the revocation list, if not, then it builds a validity attestation statement with the current timestamp and interacts with other nodes to create a multisignature. After signing the statement it issues the attestation to the prover. When a verifier gets this attestation s/he can decide the validity of the credential by checking the timestamp.
Their implementation uses mobile devices as the devices of prover and verifier. In order to establish a communication channel between the devices of the prover and the verifier, they have used bluetooth instead of wifi direct or NFC as bluetooth is more widely available across mobile devices. They have used DIDcomm as the communication protocol. The verifier invites the prover to establish a communication through a qr code.They have used a trust store to hold the DIDs and the public keys of the SSI nodes. When a new node joins or leaves the SSI system this trust store is updated. When the prover shows the VC to the verifier, the verifier can check the multi signatures of the VC and cross check it with the public keys from his/her trust store. The multi signature may have more or less nodes than the trust store (because new nodes may have joined or left the SSI system and the verifier's trust store may not have been updated) but a significant number of public keys must match in order to verify. In the distributed ledger of the SSI system the nodes also maintain a revocation list. A VC might be revoked by the user or the issuer, thus may not be attested. When the user requests for attestation the nodes check this revocation list to ensure that the VC can still be attested. Their system relies on multi signatures. They have used the BLS signature scheme for the multi signature and for the proof system they have used Bulletproof which does not rely

on a trusted setup in comparison to SNARK.
Though this approach tries to solve the offline verification problem using the SSI system, it still lacks the real time offline verification opportunity. If the prover is offline for a longer time then the attested statement will have an older timestamp which is not very convenient for the verifier to verify.

Abraham et al. [8] proposed the derivation of eIDs into the SSI system in a privacy preserving way, revocation of the identity attributes and verification of the identity attributes in an offline scenario.In real life identities or related attributes are issued by trusted third parties i.e the governments or any organization that the verifier trusts. That is why these identities are called qualified identities. But in a SSI system, anyone can issue identities for anyone. So the identities are not truly qualified identities. One approach to mitigate this issue is importing the existing qualified identities into the SSI system. One such ID is the eID issued by the government. By importing the eID we can take advantage of the SSI as well as use the qualified identity. That is what the authors tried to establish through the paper.
The process starts with the prover importing the eID to his/hr digital identity wallet. Prover sends a request to the identity provider asking for the attributes. The attribute provider or issuer asks for identity verification from the prover. After successful authentication the issuer issues an assertion about the prover's identity attribute. This assertion is then imported to the prover's digital identity wallet. Upon importing the identity to the wallet, the attestation process starts automatically. First, a proof is created stating the knowledge of secret attributes signed by the IdP and that there is a commitment to these attributes and also that there is a valid signature on the attributes. Next, the attestation is requested at the SSI network by providing the proof as well as the commitment. The SSI network, which runs an adapted version of the Byzantine-fault-tolerance (BFT) protocol, receives the request including the proof and distributes it to the network. Furthermore, the nodes in the SSI network verify the proof and sign the commitment. If the verification was successful, the nodes add the revocation ID to the revocation accumulator. Then the nodes exchange their signed result, verify it again and create a multi-signature by aggregating all signatures. The resulting attestation is issued to the prover and stored in the wallet. For revocation, the proposed approach utilizes an accumulator-based revocation mechanism to revoke credentials without revealing the identity of the user. Then the user can selectively disclose attributes and prove the validity status of their credentials to an offline verifier. The proposed approach employs non-interactive-zero-knowledge (NIZK) proof systems, and in particular succinct non-interactive arguments of knowledge (SNARKs), to prove the authenticity of the original credential and that both credentials contain the same attributes. This approach provides efficient proving and verification algorithms as well as constant-size proofs. During the offline verification process, the user provides the SSI credential and a (non-)revocation witness to the verifier. The verifier then checks the validity of the credential by verifying an attestation provided by the SSI network. The attestation is a zero-knowledge proof that the credential was derived correctly from the original eID and that the credential is still valid according to the revocation list.
The verifier uses the trust store to verify the attestation provided by the SSI network. The trust store contains all public keys of the SSI network, which are used to verify

the digital signatures of the attestation.
In order to implement the system they have first used the NIZK proof to prove that the attributes were imported from the eID. NIZK or non-interactive zero knowledge proofs can claim the truth of a statement without knowing the details of the statement. So it does not reveal the identity attributes of the user. For transforming the eID data to SSI data schema they have used Pedersen commitments. Pedersen commitments allow one to commit an assertion without revealing the assertion itself. The approach also uses a distributed ledger to maintain the trustworthiness of the attributes.

Chotkan [9] presented an approach for revocation and offline verification mechanism of self sovereign identities using a fully distributed and gossip based propagation algorithm.
A gossip based propagation algorithm is one kind of decentralized asynchronous information exchange protocol. As the name implies, it is based on how gossip is spread in real life. It allows for a periodic exchange of data with random peers. This algorithm ensures the distribution of the revocation list across the network without relying on centralized infrastructure or strict network requirements.The steps of this approach is as follows- The issuer issues a revocation of an attestation for a credential. This issue is propagated in the entire SSI network using the gossip propagation algorithm in a decentralized manner. The nodes of the SSI network participate in the process of propagating the announcement of revocation to their peers. The algorithm ensures that the announcement reaches the verifiers and all other relevant parties. Then the revocation list is stored locally on the verifier's device. This local storage helps the verifier to verify the credentials of the prover without internet access. When a credential is presented for verification, the verifier checks the revocation status by referencing the locally stored revocation information. This allows the verifier to independently determine the validity of the credential, even in offline scenarios.
The subject presents their credentials to the verifier, along with a signature over a nonce to prevent replay attacks. The verifier then verifies the metadata signature, hashed value, and list of attestors to ensure that the credential is valid. Then based on various criterias such as if the metadata signature is correct, if the hashed value is correct, if the list of attestors contains a trusted authority, if the attestation is not revoked etc the verifier verifies the credentials. As in order to verify the credentials, the verifier only needs access to the revocation list that is stored on their device locally so there is no need for internet access while verifying the credentials.

Enge et al. [13]proposed a SSI solution with a decentralized network by using DIDComm and Bluetooth Low Energy (BLE) which can verify credentials without the need of the internet. The proposed model consists of two parts - access control and secure messaging. Out-of-band (OOB) message, BLE connection establishment, device messaging , DID connection establishment, presentation exchange these five protocols are used to achieve access control and secure messaging. Two devices establish connection using OOB. BLE session allows the flow of data over a channel between the devices. Cryptographically secure direct peer-to-peer connection is provided by the DID connection. Presentation exchange deals with proof presentation, data format of VCs and other claims. The proposed system only allows flow of

data from central to peripheral, here depending on who sends the message both end devices can act either as central or peripheral. The proof-of-concept is implemented with the react native application. Users interact with a mobile application. The authors implemented a custom library to provide the application BLE functionality. To establish connection , the device which will act as the central device advertise service UUID. This proposes a security threat as other unintended devices can discover and obtain the OOB message and connect with the central device. Also there is a possibility of MITM attack during the connection establishment. The paper suggested a secure BLE connection for data and documents exchange to prevent tampering with data. The authors claimed that the use of BLE is advantageous as most of the devices have bluetooth , it is energy efficient and does not require any other external communication support such as internet. Integrating the DIDComm messaging protocol with Bluetooth Low Energy as the transport layer enables the wireless transmission of messages while preserving the advantages of DIDComm for secure and authenticated communication at the application level, the paper claimed.

Lux et al. [7] integrated OpenId Connect Provider with Self-Sovereign Identity with the purpose of enabling users to have many options of identity provider instead of having to choose from a handful of providers (eg: google, facebook). Also , the paper proposed a ledge technology which will be decentralized by leveraging the public key infrastructure powered by verifiable credentials (VC) . Single sign on (SSO) solution provided by google , facebook is supported on many websites , though there is a serious concern for lack of privacy of users. Currently OAuth 2.0 and OpenID Connect (OIDC) are the industry standards that SSO solutions follow. Oauth 2.0 is a framework used for authorization, on top of it there is OIDC which is an authentication layer. The concept of a self sovereign version of OIDC is that users do not necessarily need to have an account in the SSI OIDC provider , but still can provide personal information when asked. Websites can authenticate with the help of certificate authority (CA) and the domain name system (DNS) and public key infrastructure (PKI) , which relies on X.509. CA can malfunction and there is a high certificate cost for PKI. It can be solved with a decentralized identifier (DID) where the DID subject can be interacted with decentralized , verifiable attributes provided by the DID. A DID document containing methods for verification, cryptographic material can be used to resolve the DID. Point to be noted , DID's objective here is to provide privacy and ensure self-sovereignty of the user (holder) , it is not designed to have users personal information which can be trusted. Verifiable information is provided by VCs. VC's can hold and express the same information as a physical credential. The use of digital signature in VC makes it more secure and credible. The authors claimed that hyper ledger indy can facilitate OIDC standard claim set when integrated with a SCHEMA and CLAIM DEF with all attributes. Here the sovrin network will allow the user to use attributes issued by any issuer to login without having an account at a particular SSI OIDC provider. The paper suggests OIDC providers powered by SSI to authorize using public key authenticated encrypted and DID-signed messages using an additional VC-based authentication factor which will contain necessary user information. This authorization process collects data directly from the user and is more trustworthy as the verifier could receive user attributes signed by government organizations. This process also protects user privacy by enabling users to send credentials signed by the users himself. So here any sovrin

issuer can be a part of the SSI OIDC provider. The exchange of proof between the holder and verifier will be a one-step process , where the public and private keys of the DID will be encrypted and decrypted. The authorization can be performed with a mobile SSI app called wallet using the Hyperledger Indy. The authors implemented the integration of SSI and OIDC with Indy Agent API (version 18) , a test OIDC client, extended Node.js OIDC provider. The paper claims to achieve more transparency and interoperability and distributed-ledger-technologies-powered PKI. The paper also states the need for more research and innovation to conclude whether distribute-ledger-based PKI can perform better than the state-of-the-art. To conclude, the paper claimed that users will be benefited in IOT, PKI and OIDC scenarios with decentralized authentication and SSI benefits.

Mohammadzadeh et al. [10] mentioned that businesses utilize invoice factoring as a financial instrument to deal with liquidity problems. The authors suggest a protocol that makes use of distributed ledger technology to improve the security and effectiveness of this procedure. The protocol is intended to address a number of issues with invoice factoring, such as the defense of private data and the avoidance of fraudulent activity. The writers stress the significance of self-sovereign identities, which give people authority over their own personal information and digital personas. Through the integration of decentralized ledger technology and SSI principles, the protocol seeks to offer invoice factoring clients a transparent and safe platform. The article describes the main elements of the suggested protocol, such as the usage of data security safeguards, dispute resolution procedures, and verifiable credentials. Credentials that can be verified are essential for proving the legitimacy of the parties taking part in the factoring process and lowering the possibility of fraud. In order to protect the integrity and confidentiality of sensitive data kept on the distributed ledger, data security procedures are put in place. The protocol also contains dispute resolution procedures in case disagreements occur during the factoring process, which improves the system's overall dependability. Utilizing distributed ledger technology—more especially, a public blockchain—to record and validate factoring agreements is one of the protocol's key components. The authors emphasize the benefits of utilizing a public ledger for this purpose and compare their approach with related works. Additionally, they highlight how smart contracts help to automate components of the factoring process like fund disbursement and agreement verification. The protocol's security analysis looks at a number of issues, such as data security, privacy, availability, verifiable credentials, and dispute resolution. To guarantee the system's resilience, the authors list various security needs and the appropriate mitigating techniques. They also go over how to keep factors safe from one another and how to stop different factors from accessing the same invoice. An assessment of the suggested protocol is included in the article's conclusion, emphasizing its possible advantages for companies and people engaged in invoice factoring. The authors emphasize the importance of their findings in the context of developing technology and financial services by acknowledging the money and assistance they got for their research. In conclusion, the paper offers a thorough methodology for decentralized factoring for identities that are self-sovereign, utilizing distributed ledger technology to improve security, openness, and effectiveness in the invoice factoring procedure. The potential for creative solutions in the financial services sector is demonstrated by the combination of blockchain technology with SSI principles

and verifiable credentials.

Bambacht & Pouwelse [12] developed an infrastructure that aims to establish a decentralized societal framework that incorporates self-sovereign identities (SSI) and facilitates private, secure, and authentic interactions between individuals. The infrastructure also supports revocation and offline verification of self sovereign identities. By enabling SSI into the system, the infrastructure allows users to have ownership and control over their identity, Users can authenticate their identities without relying on a central authority which increases privacy and security.
The infrastructure uses p2p overlay IPV8 for decentralized communication which removes the need of any intermediary service or central servers. This reduces the chances of data breach. A personalized blockchain called TrustChain is used as the distributed ledger to keep track of transactions. It incorporates distributed attestations for SSI, including offline revocation, to enhance privacy and robustness.
The authors state that SSI solutions can be implemented in mobile applications with a fraction of the cost of a centralized authentication system. The first example that they provided is of IRMA. It is an operational platform that fetches SSI and personal information of users from government servers and stores them locally in the user's device. As the identity and credentials are stored locally on the device so the verification can be done while being fully offline. Sovrin network also uses a similar methodology but they also let developers build their own SSI application on top of their blockchain ecosystem. But both IRMA and Sovrin depend on a central authority for the revocation of a credential. This is contradictory to the ideology of SSI as it must be decentralized. So the authors refer to another thesis by Chotkan (2020-2021) which incorporates decentralized attestation of credentials which also includes offline revocation and verification of SSI.

Soltani et al. [11] explored the challenges of the self-sovereign identity (SSI) , the ecosystem and foundation of self-sovereign identity. In this paper the authors aim to discuss the importance as well as the lack of self-sovereign identity. The paper first addressed the challenges of management of digital identity. The paper highlighted that in the current digital identity management system , users do not have any control over their identity. Also, there is serious concern about users' privacy and security threat. Also , managing different accounts and passwords is becoming a hassle for the users. The paper claimed that SSi addressed these problems. After that the paper discussed the architecture of SSI consisting of decentralized identifiers (DID) , verifiable credentials (VC), Decentralized public key infrastructure (DPKI) , blockchain , verifiable data registry (VDR) , identity hubs and agents. DIDs , stored in a distributed structure, will have cryptographic properties and will be globally unique. Digital signature will use DPKI , DPKI will also allow users to manage , create , revoke their public-private key pairs. Users will store their personal data in VDR which is a decentralized system that will also ensure user privacy. Identity hubs and agents are softwares will provide the user with an interface to communicate in the digital domain. The authors also mentioned technologies like ring signatures , zero knowledge proof , threshold secret sharing , and trusted execution environments to ensure the privacy and security of users. Ring signature is a digital signature through which an individual on behalf of the group can sign a message anonymously. Zero knowledge proof is a protocol which allows an entity to

prove its claim to another entity without disclosing any information. For example , if I have to prove someone that I am above 18 years old , using zero knowledge proof I can prove that I am above 18 years old , but the verifiers will not know my actual age. Threshold secret sharing method distributes a secret among the participating entities in such a way that no single entity has access to the complete secret. The paper claimed that this can significantly improve security. The paper also highlighted regulations focused on protecting user privacy like General Data Protection Regulation (GDPR) , Payment Service Directive (PSD2), Electronic Identification, Authentication, and Trust Services (eIDAS), National Institute of Standards and Technology (NIST) Privacy Framework. To conclude, the paper emphasized on the importance of continued research for privacy-enhancing technologies, key management to support the evolution of SSI.

To the best of our knowledge no one proposed an approach to get the latest update of the validity of the credentials after the user or the validator goes offline. We intend to solve this issue and propose a new framework called OVoS (Offline Verification of SSI which will be discuseed in chapter 5) through which users can get their credentials validated even after going offline. This is very important because not everyone may have access to the internet everywhere and downloading credentials beforehand is not feasible. So our framework enhances the SSI ecosystem significantly.

3.1 Summary of Key Findings

The key findings from the literature review are described below -

Lack of offline verification solution: From the literature reviews, a significant research gap to provide a proper solution of offline verification of SSI is noticed. Most of the current solution depends on internet for verification of SSI. One of the research paper proposed a solution for offline verification, but the proposed solution verifies the credentials locally instead of verifying from a public ledger which is a big limitation.

Enhanced Security: Decentralization eliminates the dependency on central authority improving user privacy and security.

Effectiveness of cryptographic methods: Methods like ring signatures, zero-knowledge proof, multi-signature, timestamping are effective in ensuring credential validity, revocation and privacy.

Convenience of implementation on mobile devices: SSI solutions on mobile platforms using digital wallets demonstrates cost-effectiveness, better user experience with the scope of introducing offline verification solutions.

Regulatory Frameworks: Compliance with regulatory frameworks like GDPR, PSD2 promotes the development of user-centric, privacy-preserving identity management system.

Chapter 4

Methodology and Proposal

4.1 Methodology

This chapter contains the methodological approach we followed to research about offline verification os Self Sovereign identities. We followed a mixed method approach which included the development of an end to end system from issuing the verifiable credentials to verifying the credentials in a scenario where both the verifier and the holder is offline. For the qualitative part we observed several implementations of researchers and industry professionals. An exploratory method was pursued to test the technical feasibility of a system that can transfer verifiable credentials over bluetooth low energy, send the credentials to a sms server over cellular network and return the verification status without internet. In depth user testing was conducted in each step of the framework to ensure proper user adoptation feasibility. The methodology required us to develop three kind of systems for 3 distinct identities. First we developed a web portal for the issuer where they can invite the holder to establish connection and issuer can issue a verifiable credentials to the holders. The second step of development included developing an extended module in the Openwallet's bifold mobile wallet where the holders could advertise their bluetooth connection details through QR code. Third step of development included developing a qr scanner module inside the bifold wallet for the verifiers so that they can scan the holders' QR invitation and establish a peer-to-peer connection over bluetooth. In the fourth step we implemented a function that can fetch all the verifiable credentials from the wallt and let the holder send a verifiable credential to the verifier's wallet over bluetooth low energy. The next steps included developing functionalities so that the verifier could send a VC to a phone number over cellular network, the SMS server could receive the sms and forward the VC to a webhook, the webhook could verify the VC and reply to the verifier with the verification status through the SMS server over cellular network. At first we had used Twilio sms service for the SMS receive and reply part. But it was not consistent in receiving SMS from international operators to US domestic numbers. So we shifted to an android app which can forward a phone's SMS to a webhook and turned our own testing phone with a domestic phone number for receiving the message and calling the web hook. Then the communication was consistent and reliable.

4.2 Project Proposal

In this project we wish to propose a novel approach to validate verifiable credentials in offline scenarios. We present OVoS (Offline Verification of SSI), a novel framework for validating verifiable credentials with the motivation to facilitate an internetless and secure validation mechanism. We wish to develop an easily accessible system where users will be able to share verifiable credentials with validators. Validators will utilize their SSI to validate the status, authenticity and the ownership of the VC. An SSI wallet will be used by the users to manage their identities. A distributed ledger will be used to keep track of the decentralized identities and verifiable credentials. The system will be seamlessly accessible throughout mobile devices for the convenience of the users. We start our project by developing a threat model of our infrastructure. After that we analyze the functional and security requirements of the system. Next we develop the system and after that the evaluation of the overall performance of the system. Figure 4.1 shows the steps of our work.

Figure 4.1: Steps of this research

Chapter 5

Requirement Analysis

5.1 Threat Model

We will model the security threat using the STRIDE model of Microsoft [3].

T1. Spoofing Identity: this threat indicates unauthorized access by pretending to be another user. In our model, we have considered identity spoof as a security threat considering an attacker uses another user's identity when verifying herself offline.

T2. Tampering with data: This threat is concerned about the modification of data performed by unauthorized personnel. In our case, an attacker can attempt to change the user attribute store in the distributed nodes. Also, a user can fake his attributes and gain a fake identity.

T3. Repudiation: It indicates denying of actions done by a user . Here , users can deny their attempt to use fake attributes in their identity. Also the issuer can deny that it issued certain attributes of a user's identity.

T4. Information disclosure: This threat indicates the leak of users' private data. Here, an attacker can act as a verifier and leak the user's private data once the user presents that data for verification. Also , as the data will be stored in a distributed system, an attacker can access and leak user data if the attacker can hack more than half of the nodes of the distributed system.

T5. Denial of service: Attackers can flood the system with fake requests resulting in the unavailability of resources of the system to respond to its intended users. In our scenario attacker can flood our SMS server (SMS server will be discussed in chapter 5) through fake request.

T6. Elevation of privilege: This happens when an attacker can perform actions beyond normal capabilities by gaining access to the system's resources. We consider this as a security threat as the attacker can perform this operation by corrupting more than half of the nodes of our system.

5.2 Security Requirements

To mitigate the identified security threats , following requirements are needed to be fulfilled for our proposed model.

Spoofing identity: User verifiable credentials will be signed by digital signature that can be unlocked only by a private key known to the user. By this the system can prevent identity theft (T1).

Tampering with data: All the data will be stored in a distributed system. When verifying , data stored in the verifying node will be compared to other participating nodes of the system. The nodes will verify a certain credential only after the majority of nodes can reach a consensus. By doing so , any temperament of data will be identified by the system preventing data tampering (T2).)

Repudiation: Blockchain will store every transaction/action that occurs inside the system. Any action performed by a user will be traceable. It will mitigate repudiation threat (T3).

Information disclosure: User information will be encrypted. This information can only be decrypted by the private key of the user. So, users' private data will be secured as even if the attacker gets his hands on user data , he can not decrypt the data. It will mitigate information disclosure threat (T4).

Denial of service: Number of request will be limited to prevent denial of service (T5).

Elevation of privilege: Fault tolerance model will be applied to deal with the corrupted nodes to stop elevation of privilege (T6).

5.3 Functional Requirements

The functional requirements are presented below:

F1. The system is required to incorporate SSI functionalities into its online services, enabling users to access these services through SSI processes

F2. The system must be accompanied by an SSI wallet, allowing users to effectively manage their identities using this wallet.

F3. Users should have the capability to utilize the system across different devices without any disruption, ensuring a seamless experience.

F4. The system needs to support the involvement of three distinct parties: the issuer, the prover and the verifier.

F5. Users should have the ability to request verifiable credentials from the issuer through the system.

F6. Issuers must be able to issue verifiable credentials to users using the system.

F7. The holder must be able to share the credential with the verifier from the wallet via Bluetooth.

F8. The verifier must be able to send the credential to the SMS server via SMS.

F9. The verifier must be able to receive the verification result to the verifier's device via SMS.

5.4 Non-Functional Requirements

The non-functional requirements are presented below:

F1. The system should be adaptable across all types of smart phones.

F2. The system should be able to verify different types of credential formats.

F3. The exchange of credential via Bluetooth should be completed in 50 seconds.

F4. The SMS server should respond with the verification result in 40 seconds.

Chapter 6

Architecture and Protocol Flow

A novel framework called SSI4Web which is used for accessing restricted web services using SSI with the motivation to facilitate a passwordless secure authentication mechanism for the web (Ferdous et al. [14]). That framework includes the issuance and verification of verifiable credentials. Our approach extends that framework and proposes a new framework called OVoS(Offline verification of SSI) where a validator can validate the verifiable credentials of an user in an offline scenario.

6.1 Architecture

The core motivation of the OVoS framework is to facilitate an internetless offline validation mechanism for verifiable credentials. SSI4Web framework proposes the detailed protocol and PoC for VC issuance and verification and as we extend that framework we assume that our users already have verifiable credentials issued by a service provider. Our framework only facilitates the validation of those verifiable credentials by a validator in an offline scenario. Within our setting, OVoS must accommodate a few crucial functionalities. Those are:

1. Establishing a connection: A connection needs to be established between the validator and the user over bluetooth.

2. Sharing VC: For validation, the user has to share the VC with the validator through the established bluetooth connection.

3. Requesting validation: The validator requests for validation to a SMS server through SMS which uses the cellular network thus removing the need for the validator or the user to be online.

The figure 5.1 shows the high level architecture of OVoS framework. It has been designed in a way to facilitate all the functionalities mentioned for offline validation of a verifiable credential. The architecture has 5 major entities. The user, the validator, the SMS server, the SSI agent and the blockchain. The users store the Verifiable Credentials in their mobile wallet. They also use the wallet to scan the QR code generated by the Validator's wallet in order to establish connection. After the connection is established the validator requests for the credentials and the user shares that. Validator forwards the credentials to a SMS server using the SMS channel and cellular network. The responsibility of the SMS server is to forward the VC to the Controller which then forwards that to the SSI agent. SSI agent fetches

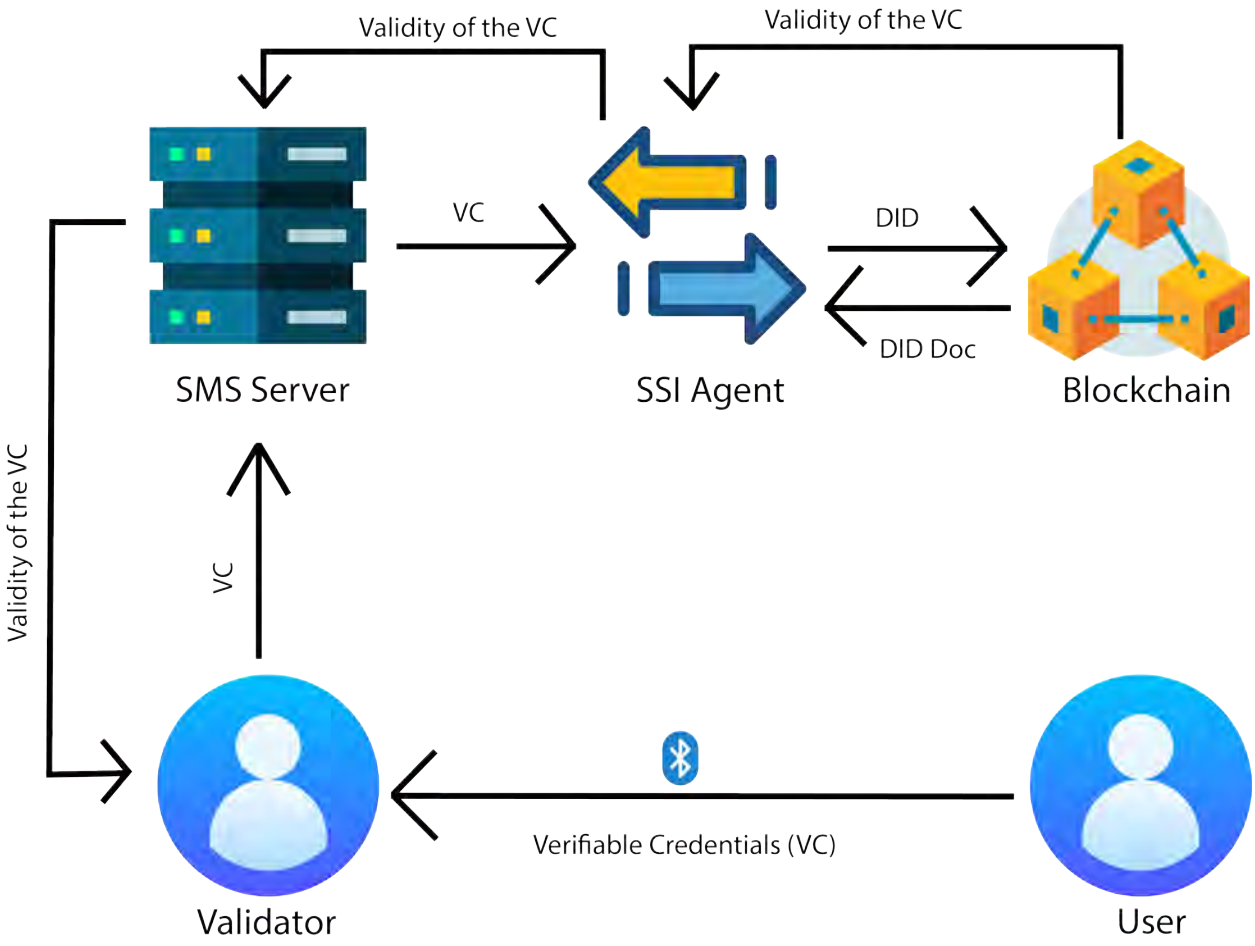


Figure 6.1: High level architecture of OVoS

the corresponding DID Docs from the blockchain as well as the validity status of the VC from the revocation registry of the blockchain. SSI agent resolves the DID and forwards the validity status to the SMS server through the controller. SMS server again forwards that to the validator using SMS.

We are in progress to developed a Proof of Concept (PoC) of our proposal using a number of tools and frameworks. The main two tools that we will use are Hyperledger Indy and Hyperledger Aries. We will use Hyperledger Aries Cloud Agent Python (ACA-py) as our SSI agent which is used to communicate with the blockchain. We will also use a SMS server to receive the VC from the validator and a controller developed with Node.js to facilitate the communication between the SMS server and the SSI agent. As the wallet, we intend to use the Aries Mobile Agent React Native Project.

6.2 Protocol Flow and use-case

To demonstrate how the proposed OVoS model could function now we will present the protocol flow and use-case of the OVoS model. We start by introducing the cryptographic notation and data model respectively in table 5.1 and table 5.2.

Notation	Description
SP	Service Provider (Plays the role of an issuer)
VD	Validator
U	User (Plays the role of a VC holder)
k_{VD}^{U}	Public key of VD for U
$K_{VD}^{-1\|U}$	Private key of VD to be used for U
k_{U}^{VD}	Public Key of U for VD
$K_{U}^{-1\|VD}$	Private key of U to be used for VD
DID_{VD}^{U}	Decentralized Identifier of VD for U
DID_{U}^{VD}	Decentralized Identifier of U for VD
VC_{SP}^{U}	A Verifiable Credential issued by SP to U
VLDT	Validity of the VC
$\{\}_k$	Encryption operation using public key k
$\{\}_{k^{-1}}$	signature using a private key k^{-}1
H(M)	Hashing of message M using SHA-512
$[...]_K$	Communication over an channel encrypted with key K
[....]	Communication over an unencrypted channel
[[...]]	Optional data
SS	SMS Server
SA	SSI Agent (Plays the role of verifier)

Table 6.1: Cryptographic Notations

Date Model: The proposed model has 3 types of request: inviteReq, credReq and ValidityReq corresponding to invitation request, credential request, attribute request and validity request.
Invite request contains the invitation url (url_{vd} in Table 5.2) of the issuer. In response the inviteReq gets inviteResp which is the invitation response from the U. invitesResp1 contains the DID of the user (DID_{U}^{VD} in Table 5.2) and inviteResp2 contains the DID of the issuer (DID_{VD}^{U} in Table 5.2). Using the DID both the issuer and the user can verify themselves. credResp means credential response which is the response of the credReq generated by the verifier. credResp contains the verifiable credential of the user (VC_{SP}^{U} in Table 5.2). The validator will verify the information of the VC_{SP}^{U}. Validation request (ValidityReq) contains the verifiable credentials that the validator got from the user, in response to the validityReq, the SSI agent (SA) sends the validity of the VC (VLDT) to the validator.

req $\triangleq \langle$inviteReq, credReq, proofReq$\rangle$
resp $\triangleq \langle$inviteResp[1-2], credResp, proofResp$\rangle$
inviteReq $\triangleq \langle \text{url}_{VD}, N_i \rangle$
inviteResp1 $\triangleq \langle N_i, \text{DID}_U^{VD} \rangle$
inviteResp2 $\triangleq \langle N_i, \text{DID}_{VD}^{U} \rangle$
credReq $\triangleq \langle$"RequestingCredential"$\rangle$
credResp $\triangleq \langle \text{VC}_{SP}^{U} \rangle$
attrReq $\triangleq \langle a_1, a_2, \ldots, a_n \rangle$
attrResp $\triangleq \langle (a_1, av_1), (a_2, av_2), \ldots, (a_n, av_n) \rangle$
$\text{VC}_U^{VD} \triangleq \langle \{(a_1, av_1), (a_2, av_2), \ldots, (a_n, av_n)\}_{K_U^{-1\|VD}} \rangle$
ValidityReq $\triangleq \langle \text{VC}_{VD}^{SA} \rangle$
ValidityResp $\triangleq \langle \text{VLDT}_{SA}^{VD} \rangle$

Table 6.2: Data Model

Prtocol Flow: The protocol flow is described in below text and presented in Table 5.3.

1. VD generates a connection request which is encoded as a QR code which is portrayed in step M1 of table 5.3.

2. The user scans the QR code using their wallet application. The wallet application decodes the QR code to extract the connection invitation data. Then prepares inviteResp1 using the U's DID and sends it to the VD via bluetooth portrayed in step M2.

3. The VD receives the inviteResp1 and prepares inviteResp2. Then send inviteResp2 to the U. The user wallet retrieves required data from inviteResp2 and combines it with data from U's DID for VD to save as a new connection to the wallet portrayed in step M3.

4. Once the connection is established U sends the credResp to the VD via Bluetooth.

5. Then the U sends back a credResp to the VD encrypting it with the public key of the VD portrayed in step M5.

6. VD sends a validityReq to the SS through SMS which includes the VC of the U.

7. SS forwards the VC to the SA over HTTPS.

8. SA fetches the corresponding DID docs from the BC and the validity status of the VC from the RR of the BC.

9. SA resolves the DID in order to prove the authenticity and ownership of the VC.

10. SA sends the validityResp to the SS over HTTPS portrayed in step M9.
11. SS sends the validityResp to VD over SMS portrayed in step M10.

M1	$VD \rightarrow U : [N_1, \text{inviteReq}]_{\text{Bluetooth}}$
M2	$U \rightarrow VD : [N_1, \text{inviteResp1}]$
M3	$VD \rightarrow U : [N_1, \text{inviteResp2}]$
M4	$VD \rightarrow U : [N_2, \text{credReq}]_{\text{Bluetooth}}$
M5	$U \rightarrow VD : \{N_3, \text{credResp}\}_{K^U_{VD}}$
M6	$VD \rightarrow SS : [N_4, \text{ValidityReq}]_{\text{SMS}}$
M7	$SS \rightarrow SA : [N_4, \text{ValidityReq}]_{\text{HTTPS}}$
M8	$SA \rightarrow SS : [N_4, \text{ValidityResp}]_{\text{HTTPS}}$
M9	$SS \rightarrow VD : [N_4, \text{ValidityResp}]_{\text{SMS}}$

Table 5.3: OVoS Protocol

Chapter 7

Implementation

float

7.1 Issuer portal

For the Proof of concept we have developed an end to end solution from issuing the VC to verifying the VC in offline scenario. We have implemented a use case of for the PoC which is driving license issuance to a driver and verification of the license by a police sergeant in a remote area where the holder and the verifier both do not have access to internet but have access to cellular network. For that we have first developed the Driving license provider authority's web portal from where the authority can issue a license to the holder. The license contains three attributes which are driver's name, license number and validity.
The portal displays a QR code which includes the invitation to for connection between the holder and the issuer. After the connection has been established, the issuer updates necessary data such as connection id, schema id, issuer DID etc. Then the issuer can issue a credentials to the holders wallet. Under the hood, first the VC is issued in W3C format which is a widely acceptable and interoperable format of verifiable credentials. This W3C vc contains the main attributes such as the attributes of the holder, the DID of the issuer, the signature etc. Then another VC is issued in Indy-Anoncreds format and the W3C VC is included as a property of the Anoncred VC and directly sent to the holder's wallet. These all happen through the Aries Cloud Agent python admin api. The aca-py admin api provides multiple endpoints for issuing the verifying the VC. The endpoints that are dedicated for issuing and verifying anoncred credentials require a credential exchange id in order to verify a credential. A credential exchange id is only generated when the credential is exchanged between the holder and the verifier though the admin api but for that we need internet access which is not possible in our scenario. The API endpoints provided for issuing and verifying a W3C VC does not require any such exchange id. We only need the signed VC in order to verify it. But the problem is, there is no endpoint provided to send a W3C credential directly to a holder's wallet following the protocols of acapy. That is why we first issue the VC in W3C format which is possible to verify without a credential exchange id, then include it as a property of an anoncred VC which can automatically be sent to a holder's wallet following acapy protcols. To develop the user facing frontend of the authority portal we have used modern javascript framework called React.js. Routing between the pages is handled

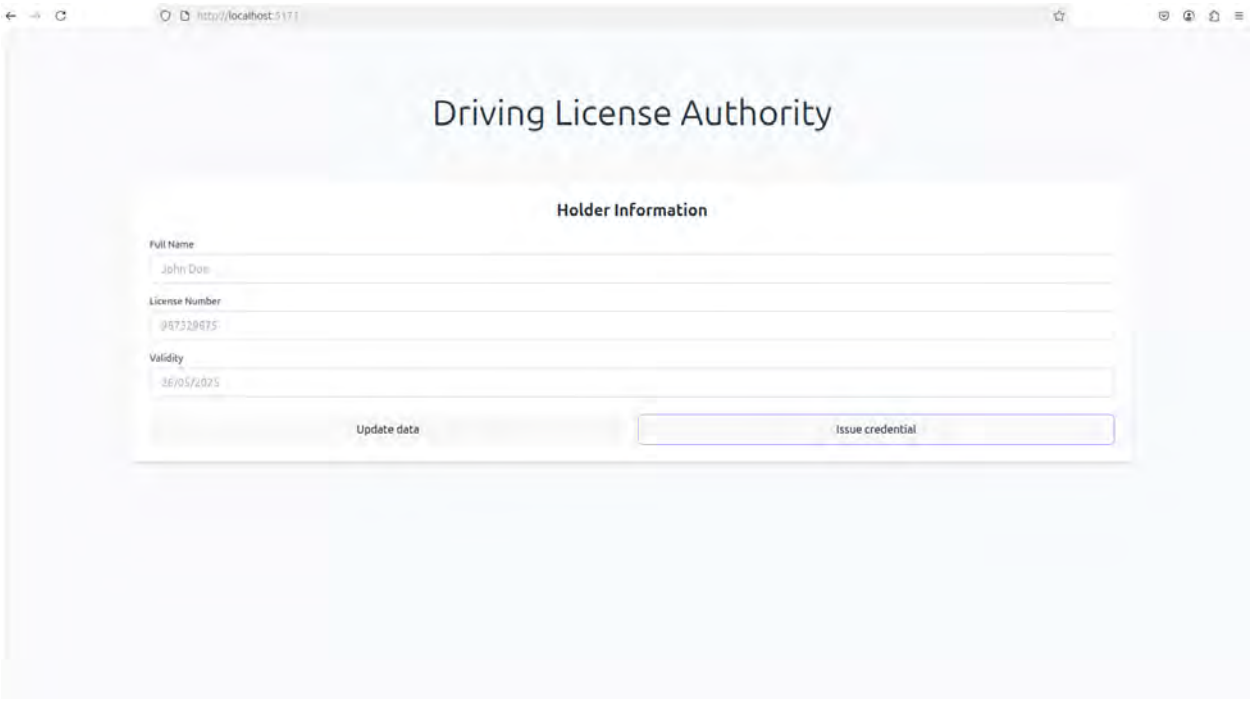


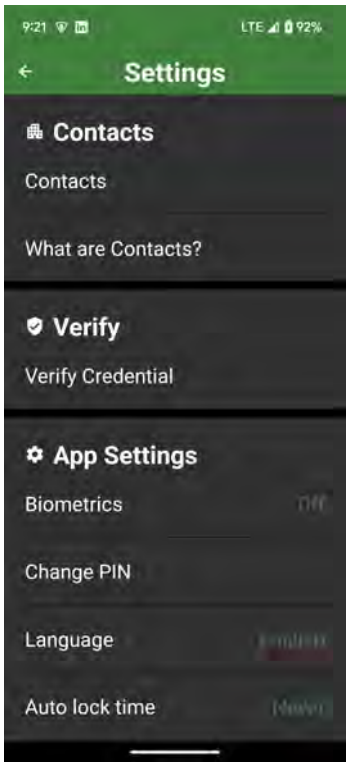
Figure 7.1: Issuer portal

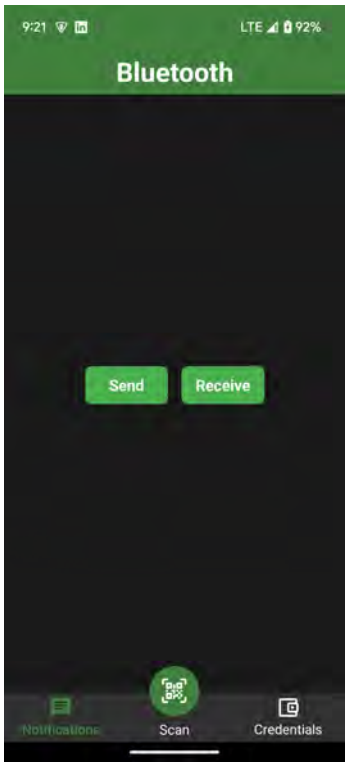
using React-router and for designing we have used Tailwind CSS. Communication between the frontend and the controller server is handled with a javascript library called Axios over http. All the techologies used in development are open sourced.

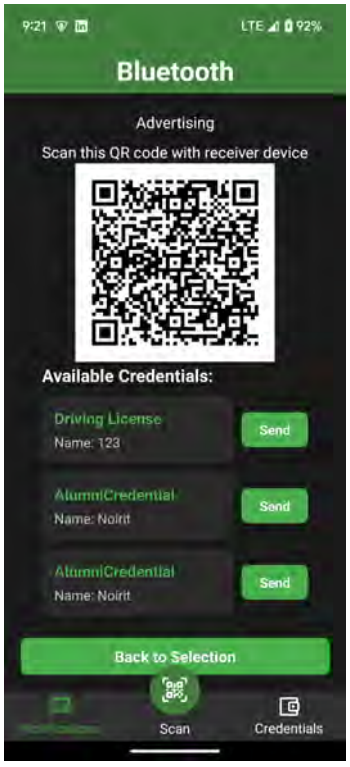
7.2 Mobile Wallet

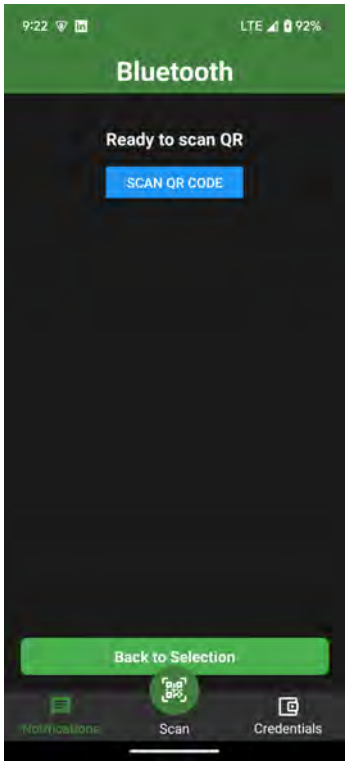
We have taken the basic open sourced mobile wallet by OpenWallet Foundation called Bifold and developed our custom modules in it to facilitate out required features which is offline credential exchange over bluetooth and offline verification of credentials through SMS. We have developed a button called "Verify credentials" in the menu of the app. Upon clicking the button the user can either choose to become a verifier or a holder. A wallet has both features so one general wallet is enough for both verifiers and holders. The holders advertise their bluetooth connection details through a QR code and also show the lists of available credentials for verification. On the other hand the other user can choose to become a verifier and open the QR scanner. By scanning the QR code, a connection between the holder and the verifier is established over bluetooth low energy. Then the holders can send the VC from their wallet and the verifiers will receive the VC in their wallet. The verifiers then can press the verify button in their wallet which will copy the credential details and paste it automatically inside their phone's text messaging app. By pressing the send button the verifiers can send the credentials to the sms server for verification. In the reply sms they will receive the verification status of the credentials. The basic wallet is developed by OpenWallet Foundation in React Native which is a Javascript framework for developing cross platform mobile application. So we have alsoe developed our features with React Native. For Bluetooth low energy transfer we have used two open source packages called ceact-native-ble-plx and react-native-

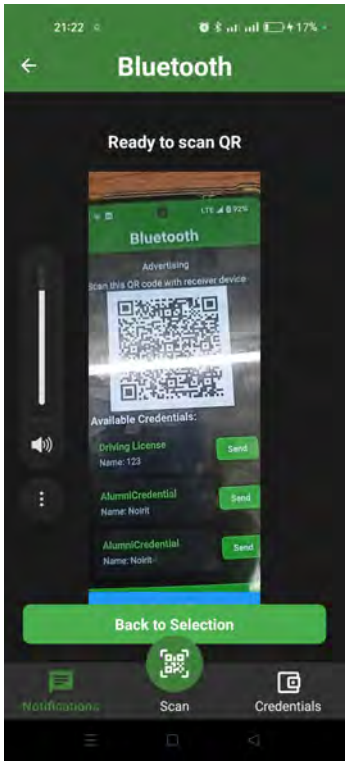
ble-peripheral. The peripheral package is used so that the holder can advertise its invitation details and the plx package is used so that the verifier can connect to the holder using those details. A native function of react-native called Linking is used to copy and automatically paste the credentials in the messaging application.

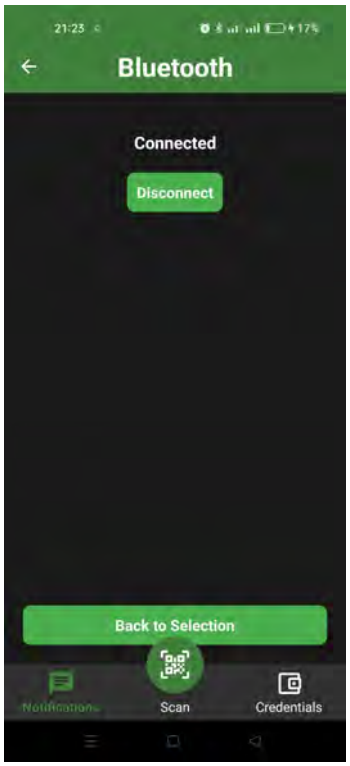


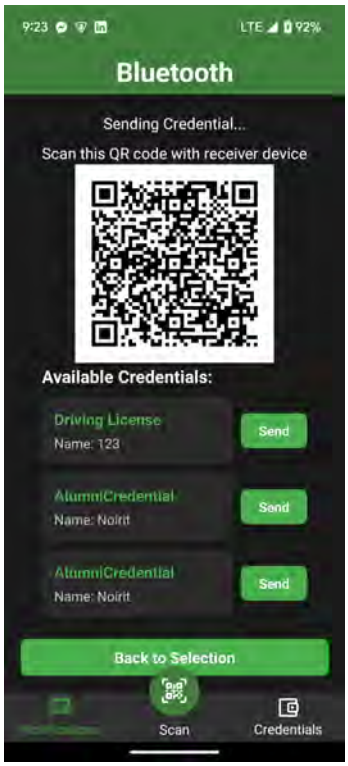


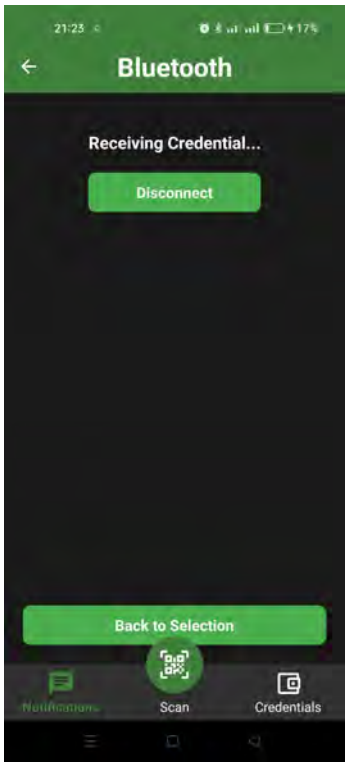


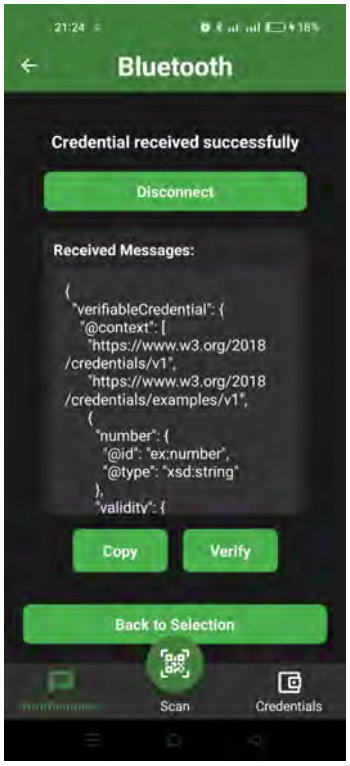












7.3 Controller Server

We have developed a controller Server with Node.js and its framework Express. This controller server is the backbone of the overall system. It receives the credentials

from the issuer portal's frontend and communicates with the SSI agent to issue and send the crednetials to the holder's wallet. When a sms is received in the sms server, the sms server also calls a webhook from the controller server. This hook then verifies the credentials by querying the ledger for the issuer's DID and then cryptographically proving the signature on the credentials with the public key it gets from the DID doc.

7.4 SMS Server

For the SMS server we are using two third party services. First we had used a Twilio for receiving the credentials sms and sending the verification status response. But the we faced problem receiving SMS from an international number to our twilio virtual number. But we could easily send SMS from the twilio virtual number to our international number. So in order to forward the credential SMS to our web hook we used an android app called Forwarder which can forward SMS of an android phone to any web hook. This way we could turn an android phone with a domestic number into a SMS forwarder server and it could correctly call the webhook each time it the phone received any SMS. The web hook calls the verification API to veriy the credentials and then replies to the sender with the verification status.

Chapter 8

Discussion

8.1 Satisfaction of System Requirements

Our proof of concept meets most of the research objectives, functional and non-functional requirements. The fulfillment of all the requirements have been thoroughly tested. The systems meets the requirements in this way:

Research objectives:
RO1: We have assessed the existing methods and found research gaps which was the scope of our work.

RO2: We have developed the novel SSI infrastructure where an issuer can issue a credential to a holder, Holder can send the VC to a verifier in offline scenarios and verifier can verify the VC without any internet access.

RO3: We have analyzed the threats of our infrastructure.

RO4: We analyzed the functional requirements and we have met all of the requirements.

RO5: We analyzed the security requirements and most of the requirements are fulfilled.

RO6: We have evaluated the functionalities and vulnerabilities of the system but permformance evaluation is not done yet.

Functional requirements:
F1: SSI functionalities are implemented.

F2: "Bifold" SSI wallet is accompanied.

F3: Cross platform framework React Native is used to develop the wallet which can be used in android as well as iOS. The issuer portal is a web application which can be run on any browser.

F4: Cross platform framework React Native is used to develop the wallet which can be used in android as well as iOS. The issuer portal is a web application which can be run on any browser.

F5: Users can get verifiable credentials issued by the issuer.

F6: Issuers can issue credentials to users.

F7: Holder can share the VC with verifier through wallet via bluetooth.

F8: Verifier can send the credential to the SMS server via SMS.

F9: Verifier is able to receive he verification result to the verifier's device via SMS.

Security requirements:
Spoofing identity: The verifiable credentials are signed by the issuer's secret key. So the system can prevent identity theft.

Tampering data: The data is signed with the issuer's secret key and the signature is included int the VC. Besides the public key is stored in a ledger and fetched for signature verification. If any data is tampered then the signature will not match.

Repudiation: Schema. DID these are stored in the blockchain so the issuer and the holder's action is traceable.

Information disclosure: The issued signature is encrypted and can only be decrypted by the secret key of the issuer.

Information disclosure: The signature is encrypted and can only be decrypted by the secret key of the issuer.

Denial of service: Not implemented yet.

Elevation of privilege: Not implemented yet.

8.2 Comparison With Related Work

This section will highlight the relative comparison between this study and other relevant studies. The comparison will highlight the attributes that make the proposed solution different from other relevant studies.

Paper name	Both parties offline	Partial offline	Query on Ledger	Timestamp with verification	Proof of Concept
Revocable and offline verifiable SSI	✓	✓	✓	✓	✗
Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials	✗	✗	✓	✓	✗
An offline mobile access control system based on self-sovereign identity standards	✓	✓	✗	✓	✓
Offline Verification of SSI	✓	✓	✓	✗	✓

Table 8.1: Comparison with relevant studies

The above table demonstrates a comparison of this research paper with other relevant studies, From the comparison it can be noticed with this research paper proposes a proper offline verification of SSI solution with prove of concept. One of the studies (Revocable and offline verifiable SSI) had similar concept but did not provide prove of concept. The most relevant study is the paper titled "An offline mobile access control system based on self-sovereign identity standards". There, the

verification of credentials is done locally in the digital wallet. So there is lack of trust in their proposed solution as their solution does not include getting the verification result from a public ledger. In our proposed solution, we get the verification result from a public ledger through our SMS server which makes the verification more reliable and temper resistance.

Chapter 9

Conclusion

Digital identity has become an integral part of users' lives. By using this digital identity users get access to many services but the services require them to verify themselves. And to do this, the internet is needed in the current scenario.But there are many situations where the internet may not be available. So offline verification of credentials is a significant requirement. Only a few researches have proposed solutions to offline verification of credentials. In one approach the user downloads the credentials with a timestamp which states up until when the credentials were valid. This partially solves the problem as the user might be offline for a longer time so the validity becomes old and has chances to be revoked after the user becomes offline. To the best of our knowledge no one proposed an approach to get the latest update of the validity of the credentials after the user or the validator goes offline. We have solved this problem by our new framework called OVoS (Offline Verification of SSI) through which users can get their credentials validated even after going offline.
Future work should include Near Field Communication (NFC) technology as an alternate way of exchanging credentials beside Bluetooth. NFC is a short-range wireless protocol which will enable tap-based data transfer. NFC has the potential to enhance the user experience. NFC also comes with its own challenges like multi-verifier handling. The future work should also address those challenges. Currently, the implementation is using a thrit-party SMS server for receiving the message from the verifier and sending the result back to the verifier. The future work will also include building a SMS server using GSM module. Having local server will ensure better connectivity and response. Also, it will enhance the data privacy of our users.

Bibliography

[1] K. Cameron, "The laws of identity," *Microsoft Corp*, vol. 12, pp. 8–11, 2005.

[2] C. Allen. "The path to self-sovereign identity." (2016), [Online]. Available: https://www.coindesk.com/markets/2016/04/27/the-path-to-self-sovereign-identity/.

[3] D. S. Cruzes, M. G. Jaatun, K. Bernsmed, and I. A. Tøndel, "Challenges and experiences with applying microsoft threat modeling in agile development projects," in *2018 25th Australasian Software Engineering Conference (ASWEC)*, IEEE, 2018, pp. 111–120.

[4] V. Baya, *Digital identity: Moving to a decentralized future*, Citi, https://www.citi.com/ventures/perspectives/opinion/digital-identity.html, 2019.

[5] 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.

[6] A. Abraham, S. More, C. Rabensteiner, and F. Hörandner, "Revocable and offline-verifiable self-sovereign identities," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2020, pp. 1020–1027.

[7] Z. A. Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, IEEE, 2020, pp. 71–78.

[8] A. Abraham, K. Koch, S. More, S. Ramacher, and M. Stopar, "Privacy-preserving eid derivation to self-sovereign identity systems with offline revocation," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2021, pp. 506–513.

[9] R. Chotkan, "Industry-grade self-sovereign identity: On the realisation of a fully distributed self-sovereign identity architecture," 2021.

[10] N. Mohammadzadeh, S. Dorri Nogoorani, and J. L. Muñoz-Tapia, "Decentralized factoring for self-sovereign identities," *Electronics*, vol. 10, no. 12, p. 1467, 2021.

[11] R. Soltani, U. T. Nguyen, and A. An, "A survey of self-sovereign identity ecosystem," *Security and Communication Networks*, vol. 2021, pp. 1–26, 2021.

[12] J. Bambacht and J. Pouwelse, "Web3: A decentralized societal infrastructure for identity, trust, money, and data," *arXiv preprint arXiv:2203.00398*, 2022.

[13] A. Enge, A. Satybaldy, and M. Nowostawski, "An offline mobile access control system based on self-sovereign identity standards," *Computer Networks*, vol. 219, p. 109 434, 2022.

[14] M. S. Ferdous, A. Ionita, and W. Prinz, "Ssi4web: A self-sovereign identity (ssi) framework for the web," in *International Congress on Blockchain and Applications*, Springer, 2022, pp. 366–379.