POLITECNICO DI TORINO

Master's Degree in Computer Engineering
Cybersecurity Focus

Decentralized Identity Management: Building and Integrating a Self-Sovereign Identity Framework

Supervisor
Prof. Danilo Bazzanella
Company Supervisors
Dr. Alfredo Favenza
Dr. Silvio Meneguzzo

Candidate
Luca Rota

Accademic Year 2023/2024

Summary

In the era of technology transition, the traditional concept of identity has been redefined, giving rise to digital identity. This important shift in the identity management is setting the need for reconsidering how we treat the identities. The most feasible solution in this context is the Self-Sovereign Identity (SSI), which is a model designed to put identity under the control of individuals.

This master's thesis, conducted at the Links Foundation, delves into the growing world of SSI within decentralized systems. By using blockchain technology and a tool called MetaMask,a SSI standalone framework has been developed and later integrated into the Data Cellar project of Links Foundation, showcasing its real-world utility. Going through the aspects related to cryptography and cybersecurity, the research ends in a decentralized application (DApp), combining an easy-to-use interface (developed with React and JavaScript) with a robust foundation (built using NestJs), for a secure and user-centric authentication.

While the decentralized identities discussions keep on evolving, this thesis fits in the ongoing discourse, emphasizing the pivotal role of blockchain technology in creating self-sovereign identity solutions and at the same time supports the commitment of Links Foundation's to digital innovation.

Acknowledgements

Contents

List of Tables

List of Figures

Listings

Chapter 1

Introduction

In a world of constantly advancing technology, the concept of identity has experienced a significant transformation. Managing identity becomes a critical issue in the present time where our lives are becoming bound with digital platforms, services and networks. Centralized identity systems, being popular, is however, encumbered by issues like the lack of user control and the data breaches. As an answer to the abovementioned challenges, the idea of Self Sovereign Identity (SSI) has been gaining more attention. SSI adopts a decentralized and user-centric approach to identity management that enables the individual to effectively assert and manage control over their identity data. Through blockchain technology, SSI enables a secure and trust-based architecture of identity verification, where intermediaries are not required and identity is protected from theft. The present research centers on the terrain of digital identity especially concentrating on the standards and implementations of SSI.

1.1 Objectives

This project delves into the ever-evolving ecosystem of Links Foundation to thoroughly explore the concept of SSI. Our ambitious objectives consist of developing an autonomous SSI framework for managing decentralized identities through the use of MetaMask and the Ethereum blockchain. Furthermore, we strive to seamlessly incorporate this cutting-edge framework into the advanced Data Cellar project at Links Foundation.

This exploration takes us into the intricacies of technology tracing the origins of identity from its non digital beginnings to its current form. By studying the foundations of SSI we gain insights, into the security and privacy measures in place

as well as addressing the prevalent cybersecurity challenges within the SSI domain.

With a focus on practicality this research culminates in developing a SSI framework that incorporates groundbreaking elements like MetaMask and a customized Ethereum smart contract. This framework has the potential to revolutionize identity management as evidenced by its integration into Data Cellar.

As we reach the end of this journey we celebrate achieving an accomplishment, creating a decentralized application (DApp). This achievement not showcases an user friendly interface built with React and JS for frontend development and NestJs for backend development but also seamlessly integrates our SSI framework. This milestone not overcomes standing obstacles but also paves the way, for a future where individuals have greater control over their digital identities.

1.2 Outline

After a brief introduction and description of the goals of the thesis presented in Chapter [1], the remaining parts of the paper are structured as follows:

- TODO

Chapter 2

Blockchain Technology

In the current landscape, blockchain technology has attracted increasing global interest due to its promising applications and potential transformative impacts on various sectors. Founded in 2008 by Satoshi Nakamoto as the mainstay of the Bitcoin system [13], blockchain has evolved from a simple ledger of financial transactions to a fundamental technology that revolutionizes the way information is recorded, shared and managed within decentralized networks.

2.1 What is the Blockchain

The blockchain is a core technology that drives many decentralized systems offering transparency, trust and safety without having to have a central authority. It is basically a distributed ledger that operates through the peer-to-peer [17], [15]. This section talks of the key components and architecture in blockchain, revealing its fundamental features and operating principles.

2.1.1 Core Elements of Blockchain

The blockchain comprises several key elements essential to its functionality:

- **Blocks**: A block is a basic unit comprising of transactional data. Every block is securely connected to the previous one; this connection creates a chain of blocks. The transactions within a block are cryptographically secured; hence, immutability and integrity [17], [15].
- **Transactions**: Transactions are diverse interactions in the blockchain network. These interactions are not limited to the financial transfers only; any

piece of valuable information can be considered as a transaction and diffused within the network [13], [24].

- **Decentralization**: Unlike centralized systems that depend on a single controlling authority, the blockchain is decentralized. It includes a web of interdependent nodes, each replicating the distributed registry. Resilience, transparency and no single points of failure are guaranteed by decentralization [17], [13].

- **Consensus Mechanism**: For verification and agreement of the state of a ledger, blockchain uses consensus mechanisms. The most common mechanism, Proof of Work (PoW), requires miners to solve the complicated cryptographic puzzles in order to add new blocks on the chain. Consensus mechanisms ensure consensus among network participants to prevent attacks by malicious actors [17], [13].

2.1.2 Blockchain Architecture

The structure of the system is carefully crafted to uphold its principles of decentralization, immutability and transparency;

- **Distributed Ledger Technology (DLT)**: At the core of blockchain lies DLT, where all participants, in the network have access to a ledger of transactions. This shared ledger eliminates redundancy. Ensures that everyone has a trusted source of information across the network [15], [13].

- **Immutable Records**: One key feature of blockchain is its records immutability. Once a transaction is recorded on the ledger it cannot be. Tampered with. Any attempt to change a transaction requires adding an one preserving the integrity of data [15].

- **Smart Contracts**: Smart contracts are self executing contracts with predefined rules embedded within the blockchain. These contracts. Enforce agreements between parties speeding up transaction processing and reducing reliance on intermediaries [15], [24].

- **Security Measures**: Cryptography plays a role in ensuring security by protecting transactions from tampering and fraud. Private key pairs authenticate. Enable secure digital identity management. Additionally cryptographic hashing guarantees data integrity and confidentiality [17], [24].

- **Peer-to-Peer Network**: The blockchain functions using a network architecture where participants can directly communicate and interact. This decentralized structure promotes trust and resilience since transactions are verified and validated through consensus, among distributed nodes [13], [24].

2.2 How the Blockchain works

The function of a blockchain system encompasses a complex chain of procedures that safeguard the integrity, transparency, and security of transactions. In this section, we delve into the fundamental mechanisms of blockchain.

2.2.1 Transaction process

Recording transactions

1. **Starting a Transaction:** In a network, users initiate transactions through their wallets. Each wallet has a pair of keys. A key, for starting transactions and a private key for authentication [17].
2. **Creating Blocks:** As transactions take place they are grouped together into blocks of data. These blocks act as containers for recording details like asset movement participants involved and timestamps [15].
3. **Hashing:** Every block contains information and refers to the hash of the previous block. Secure hash functions like SHA 256 play a role in ensuring the integrity and immutability of transactions. Hashing allows each block to have a fingerprint making identification and verification simple [15] ,[24].

Consensus Mechanism

4. **Verification Process:** When a transaction is initiated the information is sent to a network of distributed peer, to peer nodes. These nodes work together to confirm the validity of transactions using consensus mechanisms [24], [12].
5. **Formation of New Blocks:** Valid transactions are added to a Memory Pool (MemPool) where they wait to be included in a block. Miners, who are responsible for creating blocks solve cryptographic puzzles in order to mine blocks. This process, known as PoW requires resources and time [17], [12].
6. **Consensus Algorithm:** In order to add a block to the blockchain nodes must agree on its validity through consensus algorithms. The miner who

successfully creates a block is rewarded. Consensus algorithms ensure that all nodes are synchronized and in agreement, about the state of the blockchain [12].

7. **Blockchain Integrity:** The interconnected nature of blocks ensures the immutability and integrity of the blockchain. Each block references the hash value of its predecessor making it practically impossible to tamper with transactions without altering blocks [17] ,[24].

2.2.2 Blockchain Benefits

Due to the execution process described above and its architecture, blockchain is able to offer many benefits in different areas:

- **Greater Trust:** Blockchain helps establish trust across the network through utilizing the decentralized ledger, making sure that data is consistent and timely plus, it doesn't require intermediaries [17].
- **Enhanced Security:** Blockchain's security features are designed to thwart tampering as well as cybercrime. The blockchain structures transactions as read-only data which is difficult to change and ensures that the data is protected even when it is in transit. Moreover, cryptography is used as a mechanism of verification and the cryptographic infrastructure is used to store and transmit secrets [15].
- **Time Savings:** The utilization of blockchain technology decreases greatly the processing time for transactions since they get completed within a quarter of an hour due to the removal of the centralized authority that confirms [15].
- **Cost Savings:** The blockchain adjusts transaction procedure and lowers operational costs by the removal of intermediaries, as tasks are automated and there is lower duplication of activities through the usage of shared ledger [17], [24].
- **Efficiency Improvements:** The distributed architecture of blockchain increases system resilience, decreases processing costs, and removes the need for centralized network control by distributing network operations thinly, facilitating faster settlements, and solving identity management issues [24].
- **Transparency Enhancement:** A blockchain keeps a permanent record of transactions through which it delivers transparency, accountability and trust to the network users via its chronological and transparent nature [24].

2.3 Blockchain Classification

2.3.1 Types of Blockchains

The blockchain technology comes in various forms, each with unique characteristics that adapt to specific needs and application contexts Beyond the well-known public and private blockchains, it's essential to recognize two other significant variants: hybrid chains and consortium chains.

- **Public Blockchain (Permissionless):** A public blockchain is open to everybody who wants to interact and contribute to the consensus protocol, e.g., Bitcoin. This model provides a high degree of transparency though, it is susceptible to 51% attacks. Its open nature fosters a distrustful environment, as no one individual or entity is solely relied on for transaction validation [17], [15].

- **Private Blockchain (Permissioned):** In contrast to public blockchains, private or permissioned (access is restricted to authorized entities) blockchains allow only a limited set of entities to access the network. This method is popular among situations that demand more of such features such as security as well as control where it gets applied in applications like financial transactions that also handle sensitive data. An access to the network is controlled by a single entity or by particular criteria [17],[24].

- **Hybrid Blockchain:** Hybrid blockchains, on the other hand, are described by the property of switching between different modes having different types of operating systems, for example, public and private blockchains that suit specific needs. This approach provides a greater degree of flexibility and customization compared to conventional blockchain configurations; thereby, the actors have the option to decide the degree of decentralization and control that is suitable for them [17].

- **Consortium Blockchain:** A consortium blockchain is governed by more than one organization or entity working together to ensure the distributed ledger. On the other hand, blockchains that are public or private, consortium blockchains are the only ones that give power to a chosen group of nodes to validate and record transactions. This model is suitable where several subjects need to work with data and processes together but they do not want or they cannot fully trust a unique central entity [15], [13].

2.3.2 Types of Consensus Mechanisms

Blockchains also differ based on the type of consensus mechanism used. In the previous section we discussed an example of how blockchain uses a consensus mechanism called PoW [13], [28]. However there are several types of consensus mechanisms that can be used in a blockchain network.

In general, the two main types of consensus mechanisms are: PoW and Proof of Stake (PoS). With PoW miners solve puzzles to add new blocks, which requires significant computational resources and time [13], [28]. This is elaborated further in Section 2.4.1. On the hand PoS assigns the right to create blocks based on participants cryptocurrency holdings providing benefits such, as energy efficiency and scalability advantages. Further information, about PoS can be found in Section 2.4.2.

There are also other consensus mechanisms beyond PoW and PoS; refer to the table for further details.

2.4 Bitcoin versus Ethereum

Bitcoin and Ethereum stand as two big giants in the blockchain technology area, and both of them possess their own characteristics and they make their own contributions to the ever-changing world of decentralized systems. This section takes a comparative approach between Bitcoin and Ethereum; analyzing their basic architectures, transaction mechanisms, consensus methods, cryptographic techniques, and what advantages and lacks they can offer.

2.4.1 Bitcoin

Overview

Bitcoin, launched in 2008 by Satoshi Nakamoto, is a decentralized digital currency that is also accompanied by the great invention, the blockchain. It functions as a peer-to-peer decentralized electronic payment system; it introduces a new way of executing business transactions and the security of data [22]. Bitcoin is an open-source and permissionless blockchain, allowing for involvement of anyone with the required hardware.

Transaction mechanism

When a user sends bitcoins, sender hashed addresses, recipient hashed addresses, transaction amount and fee. Digital signatures prove property rights and the transaction is floated to the network, being submitted to the mempool for confirmation [3]. The mining being the process of validating transactions and so the miners pick transactions from the MemPool in order to create new blocks by solving complex mathematical puzzles called PoW [22]. The winner of the puzzle then broadcasts their newly found block to the network. After a verification done by other network nodes, the block is added to the blockchain, thus finalizing the transaction.

Proof of Work

Bitcoin uses the PoW consensus algorithm to maintain agreement among network participants about the validity of transactions. The PoW was first specified in the Hashcash paper [22]. It requires the miners to search solutions of cryptographic puzzles consuming computational power. Through the cost of computation, the miners establish their commitment to the security of the network as changing the blockchain will require enormous computational resources, hence, making fraudulent endeavors expensive.

Bitcoin Cryptography

Bitcoin use the Secure Hash Algorithm 2 (SHA-2) for cryptographic hashing, which ensures irreversible and collision resistive attributes. This guarantees the data integrity of the blockchain, which makes it impractical to tamper with transactions that are already in the database. Moreover, the Merkle Trees that Bitcoin natively uses for block data encoding take security to the next level [22].

Unspent Transaction Output

Bitcoin transactions are registered in the blockchain as Unspent Transaction Output (UTXO)s, which meaning constant fractions of bitcoins that are linked to certain owner. This UTXO is pervasively distributed over the blockchain and represents ownership. It is subsequently used in other transactions too. Bitcoin balances are not stored in user accounts but rather tracked through the UTXOs associated with their addresses [3].

Advantages and Limitations

Bitcoin is decentralized and provides encryption security that can be used to resist censorship, to ensure accessibility particularly on a global scale and to preserve integrity of the data. Nevertheless, Bitcoin has such limitations as scalability problem, transaction throughput rate without the corresponding increase in the number of transactions and undulatory acceptance and valuation. Furthermore, there are additional security concerns, such as the "51% attack" and human error, which continue to plague the Bitcoin ecosystem [22].

2.4.2 Ethereum

Overview

Ethereum, designed by Vitalik Buterin, is the introduction of Smart Contracts to the crypto-sphere, which is a novelty in the blockchain space, and, obviously, it has been a highly valuable addition. The September 2022 saw the newly born Ethereum 2.0 replace Ethereum 1.0 by incorporating the Beacon Chain and the shift from PoW to PoS consensus mechanism [10]. This change shall constitute one of the really important milestones reached by Ethereum, the blockchain providing scalability, sustainability, as well as security.

Beacon Chain Impact

The introduction of the Beacon Chain revolutionized Ethereum's operating model [10]. It replaced the original execution layer (Mainnet) and introduced a new consensus layer, the PoS which was a major improvement in terms of power consumption and the amount of energy consumed reduced from 99.95%. Validators, staking their ETH, assumed the responsibility of block production and transaction validation, moving away from energy-intensive mining. This transition helped create a sustainable network architecture as well as secure the information connections.

Transaction Mechanism

In Ethereum 2.0, validators play a pivotal role by depositing 32 ETH and operating three essential software components: a execution client, a client for consensus, and a client for validation as well. Here we have the validators appointed randomly for being picked on 12 second timeslots which then make up epochs composed of 32 slots. They formulate and verify blocks and this way they add to the Ethereum

System. This process regulates all the transactions on the blockchain. Transactions include processing stages of creating, verifying, and block proposal irrespective of the fact that the transactions are created, verified, combined into execution payloads, and broadcasted across the network so as to make self-enforcement and fault-tolerance in proper working order. Finality being one of the main features of transaction irreversibility, is sealed by using checkpoints that initiate even the smallest epochs. As requisite effort to checkpoints pairs, validators vote in a supermajority to revert them, consequently, block reversal is discouraged and the entire Ethereum network is protected from brainwashing attacks [9].

Proof of Stake

The Ethereum 2.0 version puts the PoS consensus mechanism at the core of the protocol with validators given the chance to make the most of their investment through staking [9]. Validator's deposit their own ETH as bond to release validator software which helps in the job of validating the transactions and generating new blocks proposals. In contrast to proof-of-work, the PoS is doing well in maintaining both the security and the decentralization of validators that attempt to undermine the network due to its feature of penalization.

Ethereum Cryptography

Ethereum 2.0 still relies heavily on the standards of crypto- security for safety, keeping the Keccak-256 algorithm implemented by Ethereum 1.0. These algorithmic primitives give Ethereum strong security by making sure the data integrity, confidentiality, and authenticity [22].

Advantages and Limitations

Ethereum 2.0 is an upgraded version of the original network as well as other blockchain platforms which provide these advancements. Transitioning from PoW to PoS does substantial energy reduction and making Ethereum more sustainable is one of them. Also, PoS model makes the network more secure, decentralized and scalable in order that a flourishing ecosystem of decentralized applications may thrive in it [9]. But after considering the fact that it is the second generation of Ethereum blockchain, it has certain limitations. The processing time still may be a bottleneck as Ethereum is not able to process transactions in the same rate as traditional procedures such as Visa [22]. Besides, the market fluctuation and rumors become the main problems for the stability and the reputation of the Ethereum that calls for continual technological development and management.

Chapter 3

Evolution of Identity

The notion of identity verification has changed drastically over the course of history and with the advent of digitalization a new era where different principles and problems are met. The chapter aims at dissecting the phase of identity transition from non-digital to digital identity, noting the historical background and major milestones that have shaped present-day identity verification. The development of digital identity models will be reviewed including the centralized, federated and decentralized approaches that will be discussed, with a focus on their impacts on security and privacy in the digital age. By this investigation we will attempt to create a full picture of how identity changes in the face of an increasingly interconnected world.

3.1 Pre-Digital Era

3.1.1 Origin of Identity

The identity concept has ancient origins and it has been expressed in a number of forms in the ancient times, which include jewelry, tattoos and cultural symbols. These material signs not only expressed own individuality but also signified social status, family ties, and community membership [4]. For example, Babylonians, Romans, as well as the Chinese ancestors used the tattoos and fingerprints as the primitive identification methods, and some cultures treated thumbprints as legal signatures.

3.1.2 Early Documentation

The requirement to identify people's origin developed with the growth of civilizations and the need for safe travels and commercial transactions. The earliest recorded types of documentation were in Persia in approximately 450 BCE and these documents, which bordered on rudimentary passports, carried essential information about the traveler [4]. These documents proved to be the most important for securing the safe travels and identify an individual during the journey.

The idea of the official passport took form as the societies reached a certain level of development, especially during Henry V's reign in England. In the beginning, passports started as "safe-conducts" signed personally by the king, while in the 20th century, passports turned into the standardized documents. Later, the use of photography in passports took identity verification a step further, providing a visual reference for identifying individuals accurately [4].

3.2 Emergence of Digital Identity

3.2.1 PINs and Passwords

Personal Identification Number (PIN) and passwords became critical elements of digital identity validation. Starting in the 1960s, PINs were fundamental to secure data and transactions, growing from simple four-digit codes to more complex security measures such as "chip and pin" [14]. Simultaneously, passwords that employ characters and numbers, have been used for decades to provide user authentication across multiple digital services. Together, the PINs and passwords have thus been playing a pivotal role in protecting the digital identities of the individuals.

3.2.2 Introduction of ARPANET and IP

With Advanced Research Projects Agency Network (ARPANET) coming to be in 1969, a new digital era of connectivity began. Nevertheless, it was required to have parallel robust identity protection mechanisms, as a result of this global network. Adopting the username and password, ARPANET resorted this drawback, allowing secured access to the networked resources. On the other hand, Internet Protocol (IP) took over as the standard for device addressing, which facilitates end-to-end data transferability across various networks. IP addresses, which are identifiers of networked devices, played a key role in packet routing and maintaining secure communication over the internet [14].

3.2.3 The Public Key Cryptography

Public Key Cryptography made a revolution in digital identity protection, providing trusted communication over public networks. Diffie and Hellman introduced the idea of public-key cryptography in 1976 [4]. Using this system pairs of interconnected keys, one key for encryption and the other one for decryption, are generated. Public Key Infrastructure (PKI) successfully added a layer to identity verification by matching public keys to specific entities, enabling secure transactions and data exchange.

3.3 Shifting Identity Paradigms

This section focuses on the evolvement of Identity Management (IDM) models, which have been categorized into three main paradigms. These paradigms describe the evolution of digital identity management which show trends for more user-friendly and secure approach.

Centralized IDM models (IDM 1.0), includes organizations issuing credentials to users, using shared secrets, such as usernames and passwords, for authentication. Federated IDM models (IDM 2.0) brings in third-party Identity Provider (IDP) for Single Sign-On (SSO) and so still centralizes user's personal identifiable information. Finally, Self-Sovereign IDM models (IDM 3.0) allows users to have absolute mastery over their digital identities via Digital Wallets using standards such as Verifiable Credential (VC) and Decentralized Identifier (DID) [21].

3.3.1 Pitfalls of Centralized Identity

Centralized IDM models were predominant in the initial stage of the digital identity development process, and these models were managed by organizations and institutions. Under this model, people literally gave their personal data to these centralized authoritative bodies, that in return gave them status or official recognition [16]. Having all personal data in one place was a big plus; on the other hand, it also had some disadvantages.

One of the primary issues with Centralized IDM models is that they are naturally prone to security breaches. The fact that all user data was stored in a single location meant that in case of a breach in the system, victims of this breach would be subject to widespread data leaks and identity theft [4]. Furthermore, users had to deal with the bother of having too many account names and passwords for the different platforms which increased the risks of password-related security issues. Even though it was convenient at first, centralized identity model has proved to be

inadequate in helping to solve the emerging security threats and protecting user privacy.

3.3.2 Emergence of Federated Identity

To deal with the issues caused by the Centralized IDM models, the idea of the Federated IDM models appeared, which is much more flexible and user-centered [16]. Federated, these identity management solutions paved the way for IDP, a system which enables the creation, handling, and implementation of online identities. Through the used of federated identities, the users could use a unique identity registered with an IDP to access different network applications within their domain, and authentication process would be simplified to just clicking once.

With adoption of federated identity solutions user convenience and inter-platform operability has been enhanced but there also arises new security challenges. An expansion of the IDP gave rise to a more vulnerable attack surface for Federated IDM models that made them more prone to data breaches and cyber-attacks [16]. Furthermore, the usage of several IDP suggested the possibility of privacy and user consent problems, as the user's identity info was spread across multiple service providers. Federated IDM models persisted in gaining popularity even though they encountered a lot of challenges owing to their flexibility and ease of use.

3.3.3 Evolution towards User-Centric Identity

In direct response to the rapidly developing issues surrounding traditional Centralized and Federated IDMs, user-centric identity appeared as a concept, which aims to give users back control over their digital identities. Specifically, the so-called Self-Sovereign IDM models have emphasized the importance of user control and consent in identity management, allowing them to store authenticators and certificates that had been issued by a range of service providers in their personal devices [2].

The introduction of blockchain technology was one of the pivotal transformations in user-centric identity and it provides a decentralized and immutable ledger for identity verification. The SSI model was based on blockchain that enabled users to retain authority over their data, discarding intermediaries and centralized authorities [2]. In the SSI model, users can securely and privately own and control data and identity information while sharing them in a way that protects their data sovereignty. Eventually, the deployment of SSI will need to be fully developed and adopted, however, it constitutes a major milestone on the way to a more secure and user-centric model for identity management.

Chapter 4

State of the Art of SSI

As mentioned in earlier chapters, the principle of SSI has become popular in contemporary times, particularly in the context of cybersecurity and identity management. SSI follows the decentralized method of managing digital identities, resulting in increased personal control and independence for individuals.

In this chapter, we discuss current status of the SSI model in the state-of-the-art. First, we give a historical background and then follow with an examination of evolution of SSI. Next we focus on its architecture with an in-depth look at its core components and operational steps. Through this sector, we look to furnish a full picture of the identity management framework and the SSI model in particular.

4.1 History of Self-Sovereign Identity

4.1.1 Origins of SSI

The roots of SSI can be traced back to the early 90s with the introduction of encryption methods like Pretty Good Privacy (PGP) by Phil Zimmerman [4]. PGP introduced the public key type of encryption, which formed the basis of later models of SSI. PGP allowed the users to directly exchange cryptographic keys, and, thus, enabled the creation of a network of trust without any need for centralized intermediaries.

4.1.2 The Seven Laws of Identity

The "Seven Laws of Identity" of Kim Cameron was pivotal in the development of SSI approach in 2005 [11]. These laws described the main building blocks of

the user-oriented and conducive identity framework. The Seven Laws of Identity articulate key principles guiding SSI, including:

1. **User control and consent**: Technical identity systems must only reveal information identifying a user with the user's consent.
2. **Minimum disclosure for a constrained use**: The solution which discloses the least amount of identifying information and best limits its use is the most stable long-term solution.
3. **Justifiable Parties**: Digital identity systems must be designed so the disclosure of identifying information is limited to parties having a necessary and justifiable place in a given identity relationship.
4. **Directed Identity**: A universal identity system must support both "omnidirectional" identifiers for use by public entities and "unidirectional" identifiers for use by private entities, thus facilitating discovery while preventing unnecessary release of correlation handles.
5. **Pluralism of Operators and Technologies**: A universal identity system must channel and enable the inter-working of multiple identity technologies run by multiple identity providers.
6. **Human Integration**: The universal identity metasystem must define the human user to be a component of the distributed system integrated through unambiguous human-machine communication mechanisms offering protection against identity attacks.
7. **Consistent Experience Across Contexts**: The unifying identity metasystem must guarantee its users a simple, consistent experience while enabling separation of contexts through multiple operators and technologies.

4.1.3 Modern Development

Christopher Allen had also made the term SSI more popular than ever with his 2016 article "The Road to Self-Sovereign Identity" [4]. Allen's work was built on Cameron's principles that highlighted the crucial aspects of user control, longevity, portability and limited disclosure in SSI frameworks. These principles constitute the foundation of a system that is about individuals having the ultimate power over their personal information.

For the last few years, an increase in the use of blockchain technology has led to a rapid development of SSI. With blockchain being a decentralized and immutable system, it forms the backbone of SSI systems that secures digital identities due to their immutable and transparent nature. DID and VC are among others some of

the components of modern SSI architectures, which enables peer-to-peer transactions and having no need of central authorities.

4.2 Advantages and Principles

4.2.1 Advantages of Decentralized Identity

Advantages for Organizations

- **Efficient Verification**: Organizations thus can verify information faster without involving the manual verification procedures, improving operational efficiency [8].
- **Prevention of Certificate Fraud**: Decentralized identity systems prevent abuse of fake certificates minimizing the chances of credentials being forged .
- **Enhanced Data Security**: Public-key cryptography is utilized by organizations to encrypt data safely, which therefore helps reduce the risk of data breaches.
- **Reduced Cybersecurity Risks**: Minimizing user data information makes organizations less sensitive to cybersecurity attacks. The overall cybersecurity posture improves.

Advantages for Individuals

- **Data Ownership and Control**: People retain both ownership and control of their data, and they can independently manage their digital identities [8].
- **Self-Verification**: Individuals can establish their statements without the help of third parties, thus confidence and autonomy are promoted.
- **Privacy Protection**: Decentralized identity management provides better privacy because it can shield from indiscriminate tracking and allows for selective disclosing of data.
- **Immutable Identity**: Identities will be stored and kept in the decentralized digital wallets in which they cannot be arbitrarily deleted and provide every person with a reliable digital persona.

Advantages for Developers

- **Enhanced User Experience**: The developers can build applications for users with short and easy user identification processes, resulting in increased ease of use.
- **Privacy-Preserving Data Requests**: Developers can request the data directly from users while respecting their privacy and increasing trust-users and transparency.
- **Streamlined Transactions**: Developers can simplify transactions by reaching the relevant information securely via decentralized identity wallets without any time-consuming data collection activities [8].

4.2.2 Principles of SSI

Foundational Properties

- **Existence**: People can digital assets for their characteristics which exist in the digital domain. This is what the SSI achieves [1].
- **Autonomy**: SSI provides full independence for self-governance in identity management to issue, edit, and revoke digital identities independently.
- **Ownership**: Users as the final decision-makers own their identities, including self-asserted and third-party-attributed claims.
- **Access**: Users don't have to worry about their identity; they can freely control when it is needed.
- **Single Source**: Since individuals are the single point of reference for their identities, they also prevent unauthorized information exchange without their consent.

Security Properties

- **Protection**: SSI provides strong protection via cryptographic means, thus ensuring trustworthiness, privacy and integrity of stored information.
- **Availability**: Readily accessible identities must be cross-platform compatible, while being resistant and recoverable [1].
- **Persistence**: Identities should be preserved as long as needed, protected through secure identity storage and transmission.

Controllability Properties

- **Choosability**: Potential users could decide what data relating to identity they ought to disclose, granting access to information only in line with their preferences.
- **Disclosure**: Users have the luxury of sharing identifiable data in a selective manner to third party as long as it is done in a structured way that allows for fine-grained control [1].
- **Consent**: Identity information is delegated only with user consent, ensuring privacy protection and personal autonomy.

Flexibility Properties

- **Portability**: The identity projects need the portability across platforms in order to ensure its longevity and inter-operability.
- **Interoperability**: Maximum interoperability should be achieved by SSI systems, which in that case would allow for effortless communication with existing identity systems [1].
- **Minimization**: User objectives are meant to be covered, which minimizes data disclosure, thus ensuring data protection and efficiency of the data processing.

Sustainability Properties

- **Transparency**: SSI should be transparent, open-source, and accessible, which will produce trust, and participation of the community
- **Standardization**: The Identity should be compliant with open standards that enables it to have better portability and interoperability.
- **Cost**: Identity solutions should be cost-effective or free to make them affordable and that can lead to their wide adoption and inclusivity [1].

4.3 Key Components of SSI

This section elaborates on the three main pillars of SSI model, each of which is of critical relevance to the revolution of digital identity management. We shall properly discuss DIDs, VCs, and Verifiable Data Registries, highlighting their importance and functionality in the identity ecosystem which is decentralized.

4.3.1 Decentralized Identifiers

DIDs represents a cutting edge feature of SSI. Unlike the conventional identifiers such as emails and usernames that are centralized, DIDs provides a secure and decentralized way of verifying digital identities. DIDsis an individual identifier, which can be used separately for people, organizations, and data models, without the need of any centralized authority. DIDs are owned by users and are held in identity wallets, enabling people to retain control of their original data verified by certified issuers. [5] They are as well a solution to some of the problems linked to centralized identifiers such as identity theft and data leakage. Significantly, DIDs do not contain personally identifiable information, that is why they are secure [18].

Syntactic Components of DIDs

DIDs are composed of distinct syntactic components, delineating their structure and facilitating interoperability across decentralized systems [6] :

- **Schema**: The schema serves as a guideline to provide a uniform structure and format of the DIDs thereby ensuring compatibility to standardized conventions and guidelines. These frameworks are mostly created by organizations like the World Wide Web Consortium (W3C), and they provide a bedrock for the deployment of DIDs across various ecosystems.
- **Method**: Method part defines a protocol or mechanism used for creating and maintaining DIDs within a certain ecosystem. Varying methods may involve different cryptographic algorithms or diverse distributed ledger technologies to develop and complete DIDs. Examples of methods are "ethr" for EthDIDs-based DIDs and "key" for standard key-based DIDs.
- **DID Method-Specific Identifier**: This identifier being unique for this method, differentiates the corresponding DIDs within the indicated ecosystem. It works as the reference point on DID documents finding and network interactions enabling within the decentralized systems.

DID Document

A DID Document (DDO) outlines the basic details connected to a certain DID, being the main part of the identity system. It typically includes [1]:

- **Verification Methods**: Public keys or cryptographic objects serving as authentication and verification purposes.
- **Authentication Methods**: Holder of the DID is verified through specific

validation mechanisms applied.

- **Service Endpoints**: Identifiers referring to services or endpoints associated with the DID subject, allowing interactions and data exchange to be completed.
- **Timestamps**: Proof data records, which host the verification history data or temporal metadata, enhancing transparency and auditability.
- **Signature**: Cryptographic signatures used for the purpose of security and trust of the DDO.

DID Resolution

DID resolution is the procedure for transforming a DID into the corresponding DDO that allows for retrieval of information related to a particular DID. This procedure of querying distributed ledgers or repositories seeks the relevant DDO [6]. Upon the resolution, verifiers obtain crucial attributes and secure materials, thus being able to have secure interactions and identity verification within the delegated systems. In addition to that, DID resolution enables interoperability by standardizing mechanisms for use of DID associated data as well as for its access and interpretation.

4.3.2 Verifiable Credentials

In a SSI framework, VCs represent the most crucial element as they permit identification and authentication in a decentralized manner. Briefly, a VC represents a claimed certified identity, stored by an authorized user, residing in their digital wallet, and comprising integrity features that can be verified by issuing entities [19]. They consist of attributes denoting assertions as well as ensuring the truthfulness of data provided to create a user profile. VCs have an important role in enabling individuals to direct their identity thus able to conduct secure and privacy-preserving communication in the digital ecosystem.

Syntactic Components of VCs

The syntactic structure of a VC encompasses several key components [26]:

- **Context**: This element defines a shared set of terms for interoperability among different systems giving option of using short aliases mapped to complex Uniform Resource Identifier (URI)s that define the attributes and values for specific credentials.

- **Id**: An additional attribute which allows the unique identification of entities within a credential, usually by using a URI or a DID.
- **Type**: Mandatory specification which identifies the kind of credential, allowing software systems in processing and verification.
- **CredentialSubject**: Functional for stating the claims focused on one or several subjects of the credential, as well as for presenting the needed details.
- **Issuer**: Represents the entity issuing the credential and including information like issuer identifier and further metadata.
- **IssuanceDate**: Shows the date and time when the credential is firstly valid, particularly important to determine whether it is still valid.
- **Proof**: Comprises cryptographic proofs needed to verify the authenticity and integral of the credential, including mechanisms like digital signatures
- **ExpirationDate**: May be included optionally to mark the validity duration of the credential which must be relevant over the time.
- **CredentialStatus**: Gives the current valid status of the credential, whether active, suspended, or expired.

Verifiable Presentations

Verifiable Presentation (VP)s are verifiable proofs of a person's claims that only provide selective disclosure of identity data to verifiers [26]. They are usually derived or generated from one or more VCs that have metadata and crypto signatures that can be verified by the recipient. Credentials from different sources are combined through VP which allow fast and privacy-preserving interactions within the SSI framework [19].

Zero-Knowledge Proof

Zero-Knowledge Proof (ZKP)s are a type of proof that allows one to demonstrate a value without actually revealing the value itself [26]. Essentially, ZKPs provide privacy and support selective discloser of credential attributes in the VCs ecosystem. The ZKPs enable hidden proving with the provision holding claims of participants without revealing the sensitive information, they provide privacy-protecting interactions in the SSI ecosystems. This provides the opportunity to issue zero-knowledge verifiable proofs of possession, to allow holders to disclose important information without revealing their secret. This conceals the information disclosed to verifiers by individuals utilizing ZKPs while remaining in charge

of their personal data.

4.3.3 Verifiable Data Registries

Verifiable data registries form a key pillar in SSI systems which provide a decentralized mechanism for the secure storage of identity-related data such as DIDs and VCs. These registries guarantee the validity and reliability of the data without any need to the central authority [26].

Blockchain as a Distributed Verifiable Data Registry

Blockchains provide distributed verifiable data registries as a critical feature for SSI infrastructures. Being a distributed ledger technology, blockchains ensures immutability and decentralization, which makes them a perfect choice, for recording and validating transactions in a trustless environment [1].

In the case of blockchain-based SSI frameworks, the transactions containing the assertions of the identity and the verifications of these assertions are recorded in blocks linked chronologically. Each block is given a cryptographic hash of the previous block, thus the blockchain is made secure and the ledger is done without interruption. Through cryptographic signatures of stored data and events, Blockchains provide a tamper-proof encryption repository for identity data.

Distributed Hash Table in SSI

As in blockchain, Distributed Hash Table (DHT)s ensures a decentralized way of storage and retrieval of data within the scope of SSI frameworks. DHTs are the decentralized data store components and are known for their rapid lookup capabilities based on the key-value pair [1]. Technology platforms such as the Interplanetary File System (IPFS) utilize DHTs to make the storage and retrieval of archival data more efficient, increasing the overall performance and improving the decentralization of content delivery.

IPFS utilizes the DHT technology, distributing data across multiple nodes of a decentralized peer-to-peer network. Such an approach not only increases data resilience and availability, but also minimizes the dependence on a centralized server for data retrieval. In addition, the DHT structure used in systems like IPFS facilitates content addressing, which makes it possible for users to retrieve the data using the unique hash addresses while not having to depend on any centralized storage infrastructures.

4.4 Architecture of Decentralized Identity

The Decentralized Identity architectural framework consists of four layers that correspond to different building blocks, each of them being a critical part in implementation and operation of the system. Such layers are built to offer trust and smoothen interactions among parties existing in the decentralized identity ecosystem. The implementation of this infrastructure is based on the works of the pioneers in the field, including papers from the Trust over IP Foundation [7].

4.4.1 The Four Layers

1. **Blockchain Layer (Layer 1)**:
    - It works as a base layer, making a retrievable data registry and storage for DIDs and the associated DDOs [7].
    - Deals with the definition, storage and management of VCs and their definitions, schemas and descriptions respectively.
    - It facilitates credential revocation by maintaining the Revocation Registry record when a credential issuer revokes a credential.
    - Incorporates proof of consent mechanisms for the exchange of data among entities securely.
2. **Secure Pairwise Connections Layer (Layer 2)**:
    - Enables secure communication between agents and digital wallets through share pairwise links.
    - Is in charge of the creation and maintenance of secure links between two parties, which are the agents.
    - It makes communication of personal messages safely through encrypting and decrypting.
    - Its job is to deal with the digital wallet information, guaranteeing the secure storage and management of credentials.
3. **Credential Issuance and Verification Layer (Layer 3)**:
    - Enables the issuer to issue VCs to holders in order to form the trust triangle.
    - Enables the holders to have their digital wallets and accommodate the credentials from multiple issuers.

- Provides ability to combine different claims from several VCs into one complex compound proof for attestation.
- Enables verifiers to verify the holder's proof using the VCs and disregarding the issuer.

4. **Trust Governance Layer (Layer 4)**:
   - Establishes governance structures, policies, and contracts to build up trust among parties participating in the ecosystem.
   - Specify a set of rules in which entities can stay within the guidelines of the decentralized identity system.
   - Uses Trust Anchors, Auditors, or known governance organizations which are responsible for issuer and verifier's integrity.

The security measures are implemented across the layers 1 and 2, using cryptography primarily [7]. This helps improve security to a great extent. These tasks range from the generation of public and private keys through the associated cryptographic operations during the interval of credential lifecycle (issuing, generating, verifying, and revocation), and to the encryption and decryption of messages that are exchanged between wallets and agents.

4.5 The SSI Trust Triangle

In the realm of decentralized identity management, the SSI architecture operates on the basis of a trust triangle involving three primary actors: Holder, Issuer and Verifier. Clearly, the role and interactions of these components are the determining factor in understanding the dynamics of decentralized identity systems.

4.5.1 Key Actors of SSI

- **Holder**: The Holder is the identity of the individual user within the decentralized identity ecosystem. Equipped with a digital wallet application, the Holder generates their DIDs. The DID acts as a unique identifier for the user and belongs to a digital wallet in which VCs are stored. VCs are cryptographic attestations, which are issued by trusted parties, the Issuers [8].
- **Issuer**: Issuers are the organizations or entities that are responsible for verifying the identities of users and issuing VCs. These credentials are signed with the Issuer's private key and therefore establish the authenticity and

integrity of the information contained within. After the verification, the VC is linked with the Holder's DID and it is stored in their digital wallet.

- **Verifier**: Verifiers are entities that use the provided VCs to verify the users' identities. Upon the presentation of the Holder's credentials by the Holder, the Verifier runs the public DID on the blockchain to confirm the credibility of the issuer. The verification of the cryptographic signatures embedded within the credentials by the Verifier ensures that the presented information was not tampered with and originates from a trusted source.

4.5.2 Workflow of Verifiable Exchange

1. **Creation of Decentralized Identity**: Users start the whole procedure by generating a DID for each piece of data that they want to share. This DID functions as the user's identification post within the decentralized network in a cryptographic manner [5].
2. **Issuance and Validation of Verifiable Credentials**: On request, an Issuer validates the Holder's DID usually by referring to an immutable public ledger. After validation, the issuer creates and signs the VC containing the relevant information. This credential is paired with Holder's DID and stored in their digital wallet.
3. **Presentation of Verifiable Credentials**: The Holder uses the stored VCs in their digital wallet to establish a specific claim. Through signing the provided credential, the Holder builds a VP of their claimed identity.
4. **Verification by the Verifier**: After receiving the presentation, the Verifier performs a set of verifications to ensure its authenticity is intact. This process entails checking the cryptographic signatures appearing in the credentials and authenticating the nearest public DIDs on the blockchain. The Verifier confirms the authenticity of the delivered information and thus guarantees the credibility of the reported identity [1].

Chapter 5

Cryptography and Cybersecurity Aspects

Cryptography and cybersecurity serve as vital elements in a digital identity management system, in particular, in the decentralized systems of SSI, which are based on blockchain. Cryptography enables the data to be tamper-proof, secure and privately shared through mechanisms such as hashing and digital signatures. This chapter explains the role of cryptography in SSI while providing steps on how to create and verify integrity proofs, also, we demonstrate a digital signature algorithm implementation of Edwards-curve Digital Signature Algorithm (EdDSA).

As mentioned earlier, cybersecurity also plays a key role in protecting SSI systems from various threats and vulnerabilities. In this chapter we will highlight how, despite the inherent security features of blockchain technology, SSI models still face several security challenges. Once this is done, potential attacks on SSI systems, which arise precisely from these vulnerabilities, will be evaluated.

5.1 Cryptography behind SSI

5.1.1 Data Integrity in SSI

Among numerous cryptographic aspects in the SSI approach, one of the most significant is perhaps the concept of proof used to verify the integrity and authenticity of VCs within the trust triangle, which forms the core of the SSI model. In the rest of this subsection, we delve into how a proof is created, verified, and what are the parts that make it up.

Proof Creation and Verification

The essential steps for creating a data integrity proof in an SSI system entail [27]:

1. **Transformation**: This initial step entails the preparation of data inputs through converting them using canonicalization algorithms and binary-to-text encoding.
2. **Hashing**: Cryptographic hash factories, for instance SHA-3, BLAKE-3 etc., provides hash identifiers that are resistant against hash collision and hence ensure integrity and safety of data.
3. **Proof Generation**: Implementing the proof serialization algorithms the values are calculated to make the input data secure from any inadmissible change. Well-known examples are, digital signature and proofs of participation.

Next, verifying the proof requires three steps to be completed [27]. The initial steps comprise the Transformation and Hashing processes which were previously presented, followed by the Cryptographic Proof Verification which involves some specific algorithms that confirm the reliability of the data being put in. It could mean checking up on the validity of digital signatures or verifying participation proofs.

Proof Data Model

A data integrity proof in SSI systems consists of a number of parameters, some mandatory and some optional; among the first ones there are [27]:

- **Type**: Identifies the exact proof type for cryptographic proof, thus, allows the relevant fields that are used to validate and verify the provided proof.
- **Proof Purpose**: Explicates the purpose of the proof, protects against the misuse and guarantees its proper implementation.
- **Verification Method**: Provides a description of the procedure and the data that may be required to verify the proof, which may involve cryptographic proofs or other verification tools.
- **Proof Value**: It contains the encoded binary data that is required for proof verification, which is the process that uses the specific verification method.

In addition, other optional parameters can also be included within a proof, such as [27]: id, created, expires, domain (useful for representing the security domains in which the proof is meant to be used, ensuring its proper application

within specified contexts) ,challenge (only if the domain is specific, to mitigate replay attacks), previousProof and nonce (useful to increase privacy by decreasing linkability, that is the result of deterministically generated signatures).

5.1.2 Digital Signature using EdDSA

In the manner before, the usage of the approach of digital signature is very common on the way of creating a proof. As a part of different algorithms, one that has the most outstanding impact is EdDSA, which its implementation we will cover in this subsection. Using the elliptic curve cryptography, the EdDSA is one of the most known for its speed and security.

EdDSA Key Generation

Key generation in EdDSA consists in the creation of a private-public key pair. The private key ($priKey$) consists of 32 octets of cryptographically secure random data (256 bit). For the public key ($pubKey$) the hashing algorithm (SHA-512) is applied to the private key as the first step in the process. And finally, the hashing product is converted using specific bit operations, as well as scalar multiplication, in order to get the public key [23]. Specifically, this process involves the following steps:

1. **Hashing**: The first step is hashing the private key (priKey) of 32-byte, using SHA-512, and store the result in a 64-octet buffer denoted as h'; only the first 32-byte will be considered for the $pubKey$.

2. **Buffer Pruning**: Then, some bit manipulations are performed on the buffer to ensure compliance with specific encoding requirements, for example some specific bits are clear in the first and last octets.

3. **Scalar Multiplication**: The pruned buffer is then considered as a secret scalar represented by a little-endian integer and a fixed-base scalar multiplication $[s']B$ is performed, where B represents a base point on the elliptic curve.

4. **Encoding**: Finally, the resulting point $[s']$ is encoded, through manipulation of the y-coordinate values of the curve. In this way the $pubKey$ has also been generated.

EdDSA Signing

Signatures are formed by using the private key to sign the message. This process is based on combining the message and the private key by using different cryptographic operations [23]. In our case, the message corresponds to the value of the proof to be signed. Specifically, this process involves the following steps:

1. **Hashing and Scalar Derivation**: The private key ($priKey$) is hashed using SHA-512 to obtain a digest (h'), of which the first half is used to create the secret scalar s', while the second half is denoted as prefix.

2. **Message Hashing**: The message (M') is hashed using SHA-512, along with prefix, context information (CTX) and a flag (FLG), so as to obtain a 64-octet digest (i).

3. **Point Calculation**: The point $[i]B$ is calculated, where B is the base point on the elliptic curve. This calculation includes the reduction of i modulo L, that is, the group order of B, and finally the result is encoded, obtaining R.

4. **Digest Calculation**: Then, another digest is computed by hashing the concatenation of CTX, FLG, R, the public key (A), and a modified hash of the message ($PH(M')$), and from this is obtained the little-endian integer k.

5. **Scalar Calculation**: $S' = (i + k \cdot s') \mod L$ is calculated.

6. **Signature Construction**: Finally, the signature is created by concatenating R (32 octets) and S' (32 octets, with the three most significant bits at 0) in little-endian encoding.

EdDSA Verify Signature

Signature verification attests to the authenticity of a certain signature by utilizing the public key, message (in our case the proof), and signature data. This method is based on decoding the signature, forming a digest out of the message and public key, and finally, verifying the group equation to ensure the integrity of the signature [23]. Specifically, this process involves the following steps:

1. **Signature Decoding**: Using the public key A as reference point, the signature is decoded into two 32-octet parts, which represent the point R and integer S', respectively.

2. **Digest Generation**: Then, a 64-octet digest is generated by hashing the concatenation of CTX, FLG, R, public key (A), and a modified hash of the message ($PH(M')$).

3. **Group Equation Verification**: Finally, the validity of the signature is verified by checking the group equation $[8][S']B = [8]R + [8][k]A'$, or alternatively, $[S']B = R + [k]A'$; where A' is the public key encoded.

5.2 Cybersecurity within SSI

5.2.1 Security in Blockchain and SSI

As previously outlined, in the context of SSI, blockchain may be used as a distributed ledger to implement certain security features for the system. Nevertheless, this might not be sufficient enough because the SSI model still contains some vulnerabilities. Next, we present the main features of blockchain technology, useful for the SSI system, and after that we examine the security challenges according to the security assessment of the SSI system.

Security Features in Blockchain

Among the key security features of blockchain, those most crucial for decentralized identity management include [25]:

- **Tamper Resistance**: Data immutability is ensured via blockchain thus cryptographic hashing methods, linking each block cryptographically. Consensus Protocols like Nakamoto consensus and digital signature algorithms such as Elliptic Curve Digital Signature Algorithm (ECDSA) mitigate the tampering of data.

- **DDoS Resistance**: The decentralized architecture of blockchains consensus protocols allows the reduction of DDoS attacks' impact, since they permit transaction processing even with offline network nodes. In this scenario, attackers must compromise a significant portion of the network to make it inoperable.

- **Double Spending Resistance**: Consensus protocols and transparent transactions permit the mitigation of double-spending attacks, while verification mechanisms guarantee transaction's validity, maintaining network integrity.

- **%51 Resistance**: Attacks made against consensus protocols require the attacker to gain majority control, threatening the integrity of the transaction history. Various consensus protocols have different susceptibility thresholds, so several robust security measures must be developed.

Security Assessment of SSI

The main challenges that SSI models must approach are [25]:

- **Dependency on Manufacturer Reliability**: SSI systems depend on Trusted Execution Environment (TEE)s supplied by the manufacturers, which is crucial for the security of the systems. The reliance put on the security model manufacturers come up with is also dubious and might introduce further vulnerabilities associated with single points of failure.
- **Data Availability and Memory Risks**: Storing verifiable credentials in a local storage in the SSI systems can decrease the accessibility and can expose issues related to memory corruption or misuse. The availability of the data spot on without ruining the integrity is a big question.
- **Confidentiality Protection and Information Disclosure Risks**: Although SSI systems are using different methods such as anonymous authentication and ZKPs to protect privacy, risks for disclosing sensitive identity information still remain. The reasonable selective disclosure mechanisms are still under development. The implementation of these mechanisms should be improved to mitigate the risks.

5.2.2 Potential Attacks on the SSI System

The vulnerabilities identified in the previous segment could give malicious actors the push to conduct different types of attacks and lead to damage or steal identities of other people to the SSI system. These attacks can be grouped into three categories, and an attack tree can be defined for each of them in order to better analyze them.

Fake Identity Attacks

Fake identity attacks pose a major threat to SSI systems, using fake identities to gain access to services they are not authorized for. Attackers exploit vulnerabilities in the SSI architecture to create fake credentials. These attacks can damage the trust and reliability of the identity management system as a whole [25].

Attack Tree Analysis: In this attack scenario, attackers pretend to be trusted issuers by creating fake credentials, like DIDs and public keys. Once inside the network, they trick it into believing these fake credentials are real. In addiction, vulnerabilities in the architecture, like hacked network machines, can give attackers access to secret administrative credentials and keys, allowing them to change them. For instance, in the Eclipse Attack, attackers take control of the peer-to-peer

network by redirecting connections to fake nodes, making the network accept the false credentials [20].

Identity Theft Attacks

Identity theft attacks use vulnerabilities, in the SSI system to steal sensitive and personal information, from user wallets without permission. These actions put individuals' privacy at risk and could result in types of fraudulent activities and improper use of personal data [25].

Attack Tree Analysis: In this type of attack, malicious actors might access data in wallets without permission by exploiting vulnerabilities in the SSI infrastructure. To do that, They could exploit authentication weaknesses or misuse credential verification methods to obtain additional personal information, a practice known as Credential Creep. The impact of identity theft attacks goes beyond users impacting the credibility and reliability of the SSI environment as a whole [20].

DDoS Attacks

Distributed Denial of Service (DDoS) attacks pose a serious risk to the availability and reliability of SSI system services. Through attacking the system with a large volume of traffic, the attackers intend to cripple the users' access to the system and breach the system's function [25].

Attack Tree Analysis: This type of attacks aim at several parts of the SSI system, such as issuer, holder, and verifier hosts, along with the distributed ledger nodes. Malicious actors can exploit vulnerabilities to attacks using vulnerabilities in the blockchain infrastructure, for example flooding the nodes or disrupting the consensus process. Furthermore, operational frameworks can be targeted, leading to difficulties in governance and regulatory processes that are imperative for the functioning of the SSI ecosystem [20].

Chapter 6

Framework

Chapter 7

Integration

Chapter 8

Conclusions

Bibliography

[1] Morteza Alizadeh, Karl Andersson, and Olov Schelén. "Comparative Analysis of Decentralized Identity Approaches". In: *IEEE Access* 10 (2022), pp. 92273–92283. DOI: 10.1109/ACCESS.2022.3202553.

[2] Yirui Bai et al. "Decentralized and Self-Sovereign Identity in the Era of Blockchain: A Survey". In: *2022 IEEE International Conference on Blockchain (Blockchain)*. 2022, pp. 500–507. DOI: 10.1109/Blockchain55522.2022.00077.

[3] Bitcoin.com. *How Bitcoin Transactions Work*. URL: https://www.bitcoin.com/get-started/how-bitcoin-transactions-work/ (visited on 02/14/2024).

[4] Business Reporter. *The History of Digital Identity*. URL: https://www.business-reporter.co.uk/technology/the-history-of-digital-identity (visited on 02/20/2024).

[5] Cisco FPIE. *Decentralized Identities Demystified*. URL: https://medium.com/cisco-fpie/decentralized-identities-demystified-49a65159196c (visited on 02/22/2024).

[6] *Decentralized Identifiers (DIDs) v1.0*. World Wide Web Consortium (W3C). URL: https://www.w3.org/TR/did-core/ (visited on 02/22/2024).

[7] Blockchain for Decentralized Identity. *Blockchain for Decentralized Identity: Conceptual Architecture*. URL: https://medium.com/blockchain-for-decentralized-identity/blockchain-for-decentralized-identity-conceptual-architecture-982c41e446d9 (visited on 02/26/2024).

[8] Dock. *Decentralized Identity and Self-Sovereign Identity: What's the Difference?* URL: https://www.dock.io/post/decentralized-identity#decentralized-identity-and-self-sovereign-identity-whats-the-difference (visited on 02/22/2024).

[9] Ethereum Foundation. *Proof of Stake (PoS)*. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos (visited on 02/14/2024).

[10] Ethereum Foundation. *The Merge*. URL: https://ethereum.org/en/roadmap/merge (visited on 02/14/2024).

[11] Md Sadek Ferdous, Farida Chowdhury, and Madini O. Alassafi. "In Search of Self-Sovereign Identity Leveraging Blockchain Technology". In: *IEEE Access* 7 (2019), pp. 103059–103079. DOI: 10.1109/ACCESS.2019.2931173.

[12] GeeksforGeeks. *How does the Blockchain Work?* URL: https://www.geeksforgeeks.org/how-does-the-blockchain-work/ (visited on 02/08/2024).

[13] Varun Chandra Gupta et al. "An Intrinsic Review on Securitization using Blockchain". In: *2021 International Conference on Computational Performance Evaluation (ComPE)*. 2021, pp. 971–976. DOI: 10.1109/ComPE53109.2021.9752154.

[14] Humanizing the Singularity. *A Brief History of Digital Identity*. URL: https://medium.com/humanizing-the-singularity/a-brief-history-of-digital-identity-9d6a773bf9f5 (visited on 02/20/2024).

[15] IBM. *Blockchain Technology*. URL: https://www.ibm.com/topics/blockchain (visited on 02/06/2024).

[16] Yue Jing et al. "The Introduction of Digital Identity Evolution and the Industry of Decentralized Identity". In: *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)*. 2021, pp. 504–508. DOI: 10.1109/IAECST54258.2021.9695553.

[17] Rajesh Kumar Kaushal et al. "Immutable Smart Contracts on Blockchain Technology: Its Benefits and Barriers". In: *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. 2021, pp. 1–5. DOI: 10.1109/ICRITO51393.2021.9596538.

[18] Kyung-Hoon Kim et al. "Analysis on the Privacy of DID Service Properties in the DID Document". In: *2021 International Conference on Information Networking (ICOIN)*. 2021, pp. 745–748. DOI: 10.1109/ICOIN50884.2021.9333997.

[19] Seungjoo Lim et al. "A Subject-Centric Credential Management Method based on the Verifiable Credentials". In: *2021 International Conference on Information Networking (ICOIN)*. 2021, pp. 508–510. DOI: 10.1109/ICOIN50884.2021.9333857.

[20] Nitin Naik, Paul Grace, and Paul Jenkins. "An Attack Tree Based Risk Analysis Method for Investigating Attacks and Facilitating Their Mitigations in Self-Sovereign Identity". In: *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2021, pp. 1–8. DOI: 10.1109/SSCI50451.2021.9659929.

[21] Nitin Naik and Paul Jenkins. "Governing Principles of Self-Sovereign Identity Applied to Blockchain Enabled Privacy Preserving Identity Management Systems". In: *2020 IEEE International Symposium on Systems Engineering (ISSE)*. 2020, pp. 1–6. DOI: 10.1109/ISSE49799.2020.9272212.

[22] Bharti Pralhad Rankhambe and Harmeet Kaur Khanuja. "A Comparative Analysis of Blockchain Platforms – Bitcoin and Ethereum". In: *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*. 2019, pp. 1–7. DOI: 10.1109/ICCUBEA47591.2019.9129332.

[23] Sampath S et al. "Decentralized Digital Identity Wallet using Principles of Self- Sovereign Identity Applied to Blockchain". In: *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. Vol. 7. 2022, pp. 337–341. DOI: 10.1109/ICRAIE56454.2022.10054286.

[24] Sheetal Sinha, Kumkum, and Ruchika Bathla. "Implementation of Blockchain in Financial Sector to Improve Scalability". In: *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*. 2019, pp. 144–148. DOI: 10.1109/ISCON47742.2019.9036241.

[25] Mustafa Takaoğlu et al. "The Impact of Self-Sovereign Identities on Cyber-Security". In: Apr. 2023.

[26] *Verifiable Credentials Data Model*. World Wide Web Consortium (W3C). URL: https://www.w3.org/TR/vc-data-model/ (visited on 02/26/2024).

[27] W3C. *Verifiable Credentials Data Model 1.1*. URL: https://w3c.github.io/vc-data-integrity/ (visited on 02/28/2024).

[28] Xuesen Zhang et al. "Research on blockchain consensus algorithm for large-scale high-concurrency power transactions". In: *2022 9th International Forum on Electrical Engineering and Automation (IFEEA)*. 2022, pp. 1221–1225. DOI: 10.1109/IFEEA57288.2022.10037907.