Doctoral Dissertation

Security Weakness and Privacy Preservation Analysis of SSI Management Systems using Information Retrieval and System Modeling

Charnon Pattiyanon

Supervisor : Professor Toshiaki Aoki, Ph.D.

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Information Science]

March, 2023

Abstract

The Self-Sovereign Identity (SSI) model is a cutting-edge approach to identity management that empowers individuals with complete control and sovereignty over their digital identities. This is achieved through the utilization of distributed ledger technology, allowing for autonomous administration without the need for central authorities. A system that implements the key architecture and features defined by the SSI model is commonly referred to as an SSI management system. As a personal information processing system, it is imperative that SSI management systems are designed with sufficient security and privacy measures to ensure the protection of personal information.

In SSI management systems, various security and privacy considerations can be evaluated and enhanced. The process of analyzing security weaknesses is an effective method for identifying the presence of common weaknesses in the target system. Similar to other domain software systems, SSI management systems may have specific weaknesses that require attention. The governance of the SSI management system serves as a framework for enforcing the principles and system properties defined by the SSI model, which are critical to protecting the operation of digital identities in various contexts. However, it has been noted that the current principles and system properties may not address necessary security and privacy aspects. Data sharing events within the SSI management system are unique situations in which data objects are made available with other actors. These events should be aligned with the SSI governance to ensure adequate protection.

This dissertation presents an approach to evaluating the security and privacy of the SSI management system by integrating domain expertise with information retrieval and system modeling techniques. The proposed approach consists of three solutions: mitigating SSI-specific weaknesses, improving SSI system properties, and modeling SSI data sharing events.

The first solution for enhancing security in the SSI management system is to mitigate the unique security weaknesses specific to this system. Currently, there has been limited research on SSI-specific weaknesses, making it challenging to identify them directly from the design of the SSI management system. This dissertation aims to overcome this challenge by utilizing language correlations between descriptions of common security weaknesses published in the well-respected Common Weakness Enumeration (CWE) database and the functional requirements of the SSI management system. The goal is to infer the presence of SSI-specific weaknesses and initiate further analysis and mitigation efforts. To accomplish this, the SSI Weakness Identification Framework (SWIF) is proposed. This framework combines natural language processing and information retrieval techniques with the creation of a cross-domain transfer knowledge graph to identify SSI-specific weaknesses. The results of this study indicate that a recommender system implementing the SWIF is capable of accurately identifying language correlations and inferring valid SSI-specific weaknesses with optimal efficiency.

The second solution of the proposed approach is to improve the security and privacy of SSI system properties. This dissertation leverages laws, regulations, and standards as source documents to achieve this improvement. The principles and system properties of the SSI management system must adhere to these source documents in order to ensure proper governance of the system in terms of security and privacy. However, the definitions of laws, regulations, and standards differ from the SSI model concept, making it challenging to directly align source documents with SSI system properties. To address this challenge, this dissertation presents a systematic analysis method and an improved set of SSI system properties. The analysis method is used to assess the compatibility of security and privacy controls from source documents with existing SSI system properties, and to revise or introduce new properties as necessary. The improved SSI system properties are more globally consistent with source documents than the current set and are applicable to actual scenarios.

The final solution of the proposed approach involves a method for modeling SSI data-sharing events. This dissertation aims to comprehensively analyze these events by extracting the unique types and constraints from the SSI model concept. The events are modeled as a state transition system to provide a foundation for security and privacy analysis. The transformed SSI system properties serve as security and privacy specifications in the context of data sharing. The proposed method involves modeling the SSI data sharing state transition system using the Alloy specification language. The result could report a secure and privacy-preserving data sharing system within a specified scope.

Keywords: self-sovereign identity, weakness identification, compliance property, software modeling, security and privacy

Dedicated to the loving memory of my dearest grandma,

Pranorm Pattiyanon
May 1931 - July 2022

who eagerly anticipated my success and welcomed my hugs whenever I achieved something. I am feeling sorry that I could not show my Ph.D. to her, but I take comfort in the knowledge that she is beaming with pride and happiness at my accomplishments in the most beautiful and tranquil of all heavenly abodes.

Acknowledgement

The pursuit of a doctoral degree has been arduous for me. Getting through difficulties of any kind calls for a combination of physical and mental fortitude. Without the help of everyone who believed in me, encouraged me, and cheered me on, I would not have been able to finish my doctorate.

I cannot thank my supervisor, Professor Toshiaki Aoki, enough for the opportunity to work in the Aoki lab and for the invaluable guidance and enlightenment he provided me with throughout my research period. Aside from that, I can count on him to be by my side whenever I need help with my research. In addition, I would like to express my sincere gratitude to Prof. Kunihiko Hiraishi, Assoc. Prof. Razvan Florin Beuran, Prof. Motoshi Saeki, and Dr. Takaaki Tateishi, who served as my dissertation committee members and gave me invaluable advice and intellectual contributions. Further that, I would like to express my gratitude to Assoc. Prof. Daisuke Ishii, my second supervisor, for his encouragement, insight, and assistance. Also, I would like to express my appreciation to Assoc. Prof. Kiyoaki Shirai, my minor research supervisor, for his constant encouragement and support throughout the project's brief duration.

For the wonderful times we had together at JAIST and for their unending encouragement, I am grateful to everyone in Aoki's lab and my friends, but especially Pun, Gam, and Bell who share great memories during my Ph.D. journey. Also, my sweetest girlfriend, Smithcha Kanjanawattana (Jane), who has helped me through the darkest times with the millions of words of encouragement she has spoken to me.

Last but not least, I would like to express my gratitude to my family for all of the support and encouragement they have given me, as well as for having faith that I will be successful in this endeavor.

Charnon Pattiyanon

Contents

List of Figures

List of Tables

Chapter 1

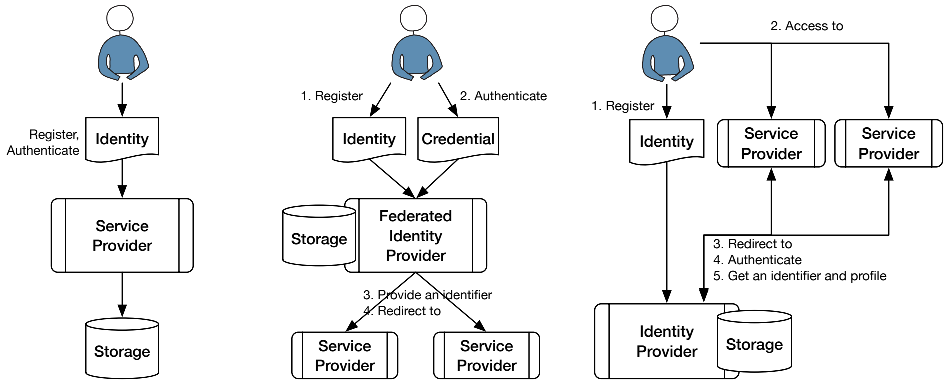
Introduction

1.1 Digital Identity and Identity Model

Digital identities are digital representations of an individual or entity that are used to access services online [1]. They often include information such as a name, age, and credit card information. A single subject can have multiple digital identities for different purposes, such as a student having separate identities for academic and e-commerce services. These identities may consist of both public and private information and are classified as Personally Identifiable Information (PII), which must be securely and appropriately managed by modern information systems.

Identity and Access Management (IAM) is a technological framework that defines the protocols and features for ensuring that only authorized users have access to restricted resources [2]. The framework is governed by the Identity Management (IdM) component, which oversees the administration of users' identities and their attributes, as well as the entitlements associated with those identities [3]. The IdM component is responsible for creating, modifying, provisioning, and revoking digital identities across all relevant environment elements. For instance, it defines a function to create a digital identity for a new employee and provisions it to the user databases of organizational subsystems, such as human resource and payroll systems.

An *identity model* is a system model that outlines the functionalities of the IdM component in a software system. It proposes an architecture and a set of protocols for effectively managing and protecting digital identities within a distributed environment. A variety of identity models have been proposed in response to the limitations of existing models.



(a) Centralized IDM (b) Federated IDM (c) User-Centric IDM

Figure 1.1: Evolution of identity models showing an architecture of the three stages: (a) Centralized, (b) Federated, and (c) User-Centric identity models.

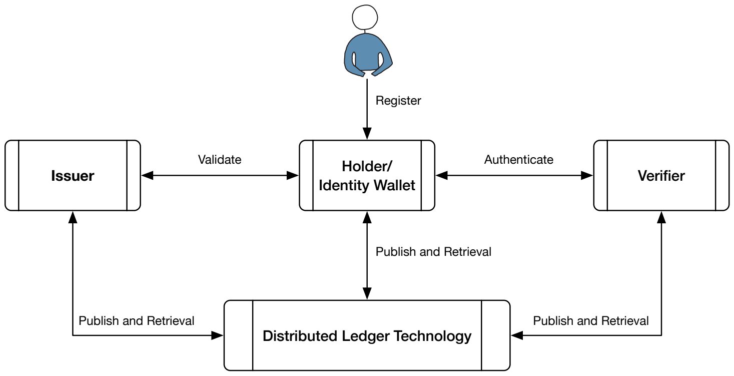


Figure 1.2: Evolution of identity models showing an architecture of the stage of the self-sovereign identity model.

1.1.1 Evolution of Identity Models

The evolution of identity models can be categorized into four stages: centralized, federated, user-centric, and self-sovereign identity. Each stage of the evolution is characterized by distinct architecture and is illustrated in the form of contextual diagrams in Figures 1.1 and 1.2. The progression of identity models is a result of the advancements in technology and the need to address the technological shortcomings of previous models.

In the first stage, the IdM component is commonly treated as a regular feature in software systems, allowing them to administer their own users. This stage is called a centralized or silo-based identity model, in which service providers act as central authorities tasked with collecting and manipulating digital identities on behalf of their users (Fig. 1.1a). The service provider verifies user access to their online services by using credentials that identify the digital identity. Repetitive administration is the major weakness of the centralized identity model. Users must manage their digital identity with each service provider they register with when changes occur.

In the second stage, the concept of federated identity providers was introduced in order to mitigate the need of repetitive administration. The identity model in this stage is commonly referred to as a federated identity model because service providers connect to a federated identity provider that collects, manipulates, and authenticate digital identities of their users (Fig. 1.1b). The federated identity provider stores digital identities and provides authentication credentials to users. When a user accesses an online service, the federated identity provider authenticates the user and provides a user identifier to the service providers proving a valid user account. This characteristic of the federated identity model is also called a single sign-on where users can authenticate once and use multiple services with shared session across them. Nevertheless, this identity model is practically realizable in an enterprise organization where service providers are controllable. It may be difficult to apply this identity model inter-organizationally.

In the third stage, to enable inter-organizational authentication, the user-centric identity model was introduced, in which the identity provider uses standard protocols (e.g., OpenID) to communicate and share user identifiers with service providers (Fig. 1.1c). With this identity model, users can browse directly to the service provider and will be redirected to the identity provider for authentication only. The standard protocol used for sharing user identifiers enables service providers to authenticate users with a variety of identity providers. A well-known example of the user-centric identity model is the authentication with Facebook or Github.

Central authorities (i.e., service providers or identity providers) are necessary for collecting and manipulating digital identities on behalf of the user during the three prior stages. These identity model stages centralize digital identities that may expose security and privacy risks of data leakages to unauthorized actors. Even large technology companies are still facing the proliferation of data leakages, e.g., Microsoft Exchange server data breach in 2021[1] and 139 million users in Canva breached in 2019[2]). Users must have trust that the central authorities will adequately protect their data.

In the fourth stage, with the emergence of the decentralized networks and technologies, researchers have taken a greater interest in the concept of

[1] Microsoft hack: 3000 UK email servers remain unsecured. https://www.bbc.com/news/technology-56372188.

[2] Australian tech unicorn Canva suffers security breach. https://www.zdnet.com/article/australian-tech-unicorn-canva-suffers-security-breach/.

Table 1.1: Comparison of four identity model stages [7].

Characteristic/Feature	**Centralized**	**Federated**	**User-Centric**	**Self-Sovereign**
Individuals can generate their own identifiers	No	No	No	Yes
Individuals are in control of their own authenticators (i.e., private keys)	No	No	Yes	Yes
Individuals are in control of their own digital credentials and certificates	No	No	Yes	Yes
Individuals can have control over their identifiers in case of loss or theft of their keys	No	No	Yes	Yes
Individuals can retrieve their credentials and certificates in case of loss or theft of their keys	Yes	Yes	No	Yes
Individuals can access the data associated with their digital identity	Unclear	Unclear	Unclear	Yes
Enabled zero-knowledge proofs	No	No	No	Yes
PII is minimized	No	No	No	Yes
Right to be forgotten can be easily guaranteed	Unclear	No	No	Yes
Repositories of authenticators and credentials are portable	No	No	Yes	Yes
Identity providers do not keep centralized databases with user's data	No	No	No	Yes
Identity providers do not have access to information about people's access to services or interactions with others	No	Yes	Yes	Yes
Implementations comply with regulatory policies	Yes	Yes	Yes	Yes
Trust frameworks are developed to allow the definition of identity providers and levels of assurance	Yes	Yes	Yes	Yes
Identity is easily retrievable in the case of a natural disaster	Yes	Yes	No	Yes
Data breaches less likely	No	No	No	Yes

decentralized identity in order to eliminate the need of the central authority and mitigate risks of data leakage. As a result, an innovative identity model was introduced. Christopher Allen shared his vision about the decentralized identity model as a Self-Sovereign Identity (SSI) model [4]. The SSI model is intended to restore full sovereignty and control over digital identities from central authorities to the user by using distributed ledger technology [4]. This identity model employs the concepts of verifiable credential [5] and decentralized identifier [6] to formulate a service authentication mechanism.

The four stages of identity models provide different functionalities and benefits that allow implementors to select for their system. Lopéz [7] compared the differences in characteristics and features as shown in Table 1.1. It can be seen that the SSI model achieves most beneficial characteristics and features. The SSI model effectively eliminates all shortcomings of the earlier stages. For example, users (as holders) can independently manage their PII and selectively disclose their data. This is the reason why the SSI model has becoming interested to both researchers and practitioners. A system implementing the functionality defined by the SSI model enables the decentralized IdM functions and it can be called as an *SSI management system*. This dissertation will focus on the SSI management system, which is a product of the SSI model and many viewpoints have not been investigated.

SSI management systems can provide superior characteristics and features compared to other identity models by employing numerous complex and novel technologies, such as blockchain technology, verifiable credentials [5], and decentralized identifiers (DID) [6]. These technologies define functions and operations that must be incorporated into the design of the SSI management system. Blockchain technology, also known as distributed ledger technology, is a peer-to-peer network-based immutable data registry. In the SSI management system, identifiers and proofs of credential are stored using blockchain technology. The verifiable credential is an open standard for decentralized credentials developed by the World Wide Web Consortium (W3C), whereas the decentralized identifier is an open standard for decentralized public exchange mechanisms. Consequently, an analysis of the SSI management system should by default incorporate the concept of these technologies. This dissertation suggests that these technologies will be incorporated into the design of the SSI management system that will be used as input.

Identity models are typically implemented as both a standalone system and a subsystem of an information system. Many well-known technology companies developed and implemented IAM systems to their users. For example, Oracle provides both centralized, federated solutions[3] for managing enterprise digital identities and IBM develop a complete SSI management system named IBM Verify Credentials[4]. Especially for the SSI model, many solutions were promoted and sold in the IT security community, such as Sovrin, Hyperledger Indy, Veramo, and Evernym. Different SSI management solutions or systems provide different features, but they all attempted to align with the same concept of the SSI model. To provide a broad overview of the SSI management system solutions, the Dutch Ministry of the Interior and Kingdom Relations commission two technology companies, INNOPAY and TNO, to survey the landscape of SSI initiatives in Netherlands. They demonstrate the landscape[5] in four aspects, as shown in Fig. 1.3.

[3] Oracle Identity Cloud Services. https://docs.oracle.com/en/cloud/paas/identity-cloud/index.html.

[4] IBM Verify Credentials. https://www.ibm.com/blockchain-identity.

[5] Control over data is still a long way off, according to Dutch Self-

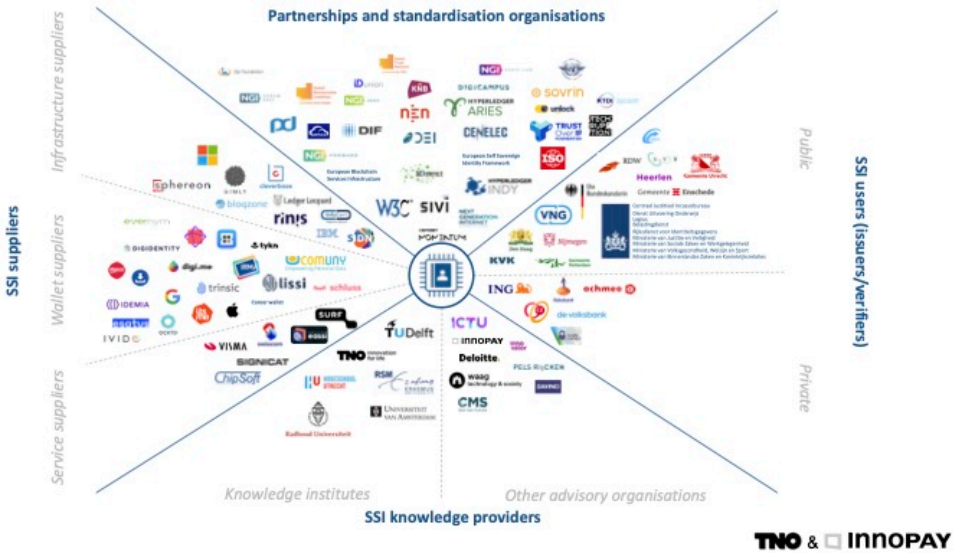


Figure 1.3: Overview of the SSI landscape in Netherlands in 2021 developed by INNOPAY and TNO[5].

1.1.2 SSI Principles and SSI System Properties

In addition to the governance of the decentralized network, sets of principles and system properties were proposed to govern the implementation of SSI management systems. The SSI principles are concise statements that specify what, when, and how the SSI model should be implemented. The scope of these principles extends from functions to quality control and targets on both the system, its platforms, and other surroundings (e.g., users). In his online blog, Christopher Allen was the first to introduce ten SSI guiding principles [4], which are summarized below.

- **Existence**. Any self-sovereign identity ultimately relies on the identity subject. An SSI simply makes public and accessible a subset of the subject's attributes.
- **Control**. The user is the ultimate authority on their identity, subject to explicit and secure algorithms that guarantee the ongoing validity of an identity and its claims. They should be able to always refer to it, update it, or even hide it. They must be able to select their preferred level of disclosure.
- **Access**. A user must always be able to easily retrieve all of their

Sovereign Identity study. https://www.innopay.com/en/publications/control-over-data-still-long-way-according-dutch-self-sovereign-identity-study identity's claims and other data. There must be no hidden data and no gatekeepers. This does not necessarily imply that a user can modify all claims associated with his identity, but they should be aware of them. It does not imply that users have access to the others' data, only theirs.

- **Transparency**. The systems used to administer and operate a network of identities must be transparent with regard to their operation and management. The algorithms should be free, open-source, well-known, and as architecture-independent as possible; anyone should be able to examine how they operate.
- **Persistence**. Identities should ideally last forever, or at least as long as the user requires. Even though it may be necessary to rotate private keys and modify data, the identity remains unchanged. A user should be able to discard an identity if they choose, and claims should be updated or removed as necessary over time.
- **Portability**. Identities must not be held by a single third-party entity, even if that entity is trusted and is expected to act in the user's best interest. Transportable identities ensure that the user retains control of their identity regardless of the circumstances, and can also increase the persistence of an identity over time.
- **Interoperability**. A digital identity system of the twenty-first century aims to make identity data widely accessible, crossing international borders to create global identities without sacrificing user control.
- **Consent**. The foundation of any identity system is the sharing of that identity and its claims, and an interoperable system increases the amount of sharing that occurs. However, data sharing must only occur with the user's consent.
- **Minimalization**. When disclosing data, only the minimum amount of data required to complete the task at hand should be disclosed. This principle can be supported by selective disclosure, range proofs, and other zero-knowledge techniques, but non-correlation is still a difficult (possibly impossible) task; the best we can do is use minimalization to support privacy to the greatest extent possible.
- **Protection**. When there is a conflict between the requirements of the identity network and the freedoms and rights of individual users, the network should prioritize the freedoms and rights of the individuals. Identity authentication must be conducted using independent, censorship-resistant, force-resilient algorithms that are decentralized.

These principles have also been used as evaluation criteria for the SSI model's designs or solutions [8, 9, 10, 11]. To expand the scope of the SSI principles, various improvements have been made. For example, Allen's principles

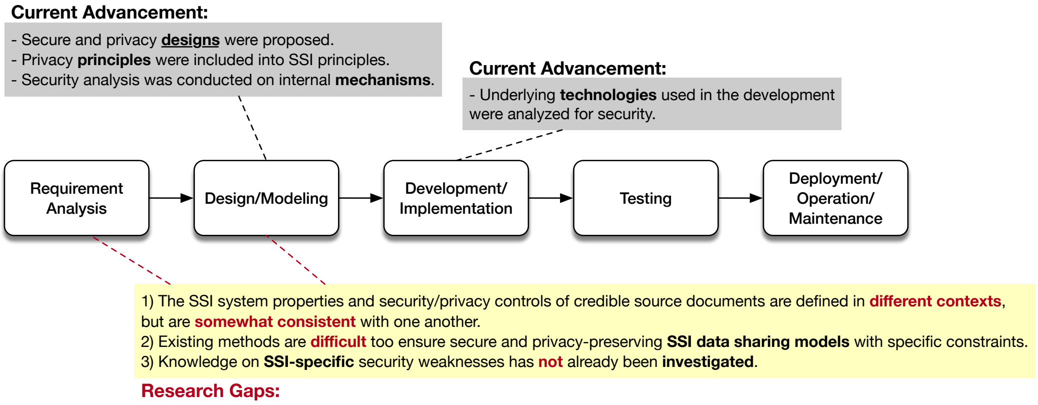

Current Advancement:
- Secure and privacy designs were proposed.
- Privacy principles were included into SSI principles.
- Security analysis was conducted on internal mechanisms.
Current Advancement:
- Underlying technologies used in the development were analyzed for security.
Requirement Analysis
Design/Modeling
Development/ Implementation
Testing
Deployment/ Operation/ Maintenance
1) The SSI system properties and security/privacy controls of credible source documents are defined in different contexts, but are somewhat consistent with one another.
2) Existing methods are difficult too ensure secure and privacy-preserving SSI data sharing models with specific constraints.
3) Knowledge on SSI-specific security weaknesses has not already been investigated.
Research Gaps:

Figure 1.4: Existing approaches to analyzing and improving the SSI model and research gaps.

have been extended to be consistent with Cameron's laws of identity [12] in one work [7] and with the General Data Protection Regulations (GDPR) [13] in another [14]. On the other hand, a work attempted to transform Allen's principles to be system properties [15].

1.1.3 Security and Privacy of the SSI Model

As a system model that deals with PII, it is imperative that the SSI model is equipped with adequate security and privacy measures to ensure user trust. Despite its decentralized nature and security features, the implementation of the SSI model may still pose security and privacy risks.

Several analysis surfaces of the security and privacy of the SSI model are amenable, in which researchers have attempted to investigate using various methodologies. Using a simple waterfall software development lifecycle, Fig. 1.4 illustrates how existing approaches analyzed and improved the security and privacy of the SSI model in different surfaces. First, alternative security and privacy-preserving designs for the SSI model were proposed, such as trust-enhancing designs [10, 11], privacy-preserving designs [9], and advanced security designs [16, 17]. Second, the SSI model's privacy was enhanced by introducing privacy-protecting principles [14] to govern its implementation (as mentioned in Section 1.1.2). Then, security analyses were performed on various components of the SSI model, such as its internal mechanisms (e.g., the DID updating process [18] and DID service properties [19]) and underlying technologies (e.g., the security of a specific blockchain network [20]). These existing approaches show significances of security and privacy of the SSI model.

Existing advancement on the security and privacy of the SSI management system identifies three research gaps (yellow boxes in Fig. 1.4) that motivate us to conduct this research, as follows.

1. Knowledge of SSI-specific weaknesses that are security mistakes within the functionality of the SSI management system has not been completely investigated yet;
2. System properties can be sources of security and privacy constraints to the SSI management systems, but they are not completely aligned by credible source documents to maximize protections; and
3. SSI data sharing events can also be another analysis surface to ensure secure and privacy-preserving properties, but they have not been investigated yet.

This dissertation will begin to fill these research gaps to improve the security and privacy of the SSI model across a wider range of surfaces.

1.2 Security and Weakness Analysis

Security analysis is an assurance technique that identifies the presence of security risks in a target system. The purpose of security analysis, according to Steed and Shore [21], is to identify areas of weakness that could be exploited by a malicious agent. Risk, threat, and vulnerability are three of the most important terms in security analysis. A risk is the probability that a system will be successfully attacked. A threat is the probability of an attack on a system. A vulnerability is a system feature that could be exploited by malicious agents. In common usage, the following equation [21] defines associations among three terms:

$$Risk = Threat \times Vulnerability \tag{1.1}$$

The preceding equation infers that a software system is at risk if it contains vulnerabilities and has some likelihood of being attacked (threat).

In recent years, the term "weakness" has become more prevalent; it refers to a deficiency or flaw in the software code, design, architecture, or deployment that, under the right conditions, could become a vulnerability or contribute to the introduction of additional vulnerabilities [22, 23]. Vulnerabilities can be thought of as a subcategory of weaknesses. In some research publication, "weakness" and "vulnerability" are used interchangeably. This dissertation will refer to a vulnerability as an exploitable weakness.

Weakness analysis is a method for identifying security weaknesses within a target system's functionality. However, security weaknesses are difficult to identify due to the need for extensive experience in software design and development. To overcome this difficulty, numerous organizations have collected and published common security weaknesses in online databases, such as Common Weakness Enumeration (CWE) [22], National Vulnerability Database [24], and Smart-contract Weakness Classification [25]. Typical weakness analysis, also known as *weakness identification*, is a process that maps common security weaknesses to the functionality of the target system.

To facilitate security analysis across multiple domains, researchers attempted to develop novel methods for analyzing and identifying common security weaknesses in online databases. Machine learning is one of prominent approaches to identifying the correlations of common weaknesses and a target system. Common weaknesses are represented in vector space models and uses machine learning to classify the correlations with specific artifacts, e.g., bug reports [26] and business process labels [27]. Another interesting approach is to apply machine learning to consider source code that contains weaknesses [28, 29]. Existing approaches to weakness analysis indicate that a method to identify common weaknesses early in the development process has not been explicitly proposed. Existing approaches, on the other hand, did not consider domain knowledge, which may be useful for identifying weaknesses in the SSI management system. Due to the difficulty of existing approaches to weakness analysis, it is necessary to develop a method for identifying SSI-specific security weaknesses that incorporates domain knowledge.

1.3 Information Privacy Analysis

Information privacy analysis is a privacy assurance technique that determines the levels of privacy preservation in a target system. Information privacy is defined as a strategic objective that seeks to protect the confidentiality of PII stored on software systems and prohibits unauthorized access to PII[6]. Typically, the objective of an analysis of information privacy is to protect the sharing of information between system components and ensure that information is not exposed to unauthorized actors or system components.

The analysis can target a variety of aspects of privacy within an information system. Various techniques and methods can be used to determine the level of privacy. Privacy-by-Design is a method for ensuring privacy by evaluating and mitigating privacy risks during the system's design phase [30, 31]. Moreover, Privacy-by-Design encompasses both system and organizational

[6] Data Privacy (Information Privacy). https://www.techopedia.com/definition/10380/information-privacy.

processes, which will maximize the level of privacy. However, as a decentralized system incorporating a peer-to-peer network, Privacy-by-Design may not completely protect the SSI model's privacy. Compliance with privacy policies is an additional technique for analyzing information privacy. Some researchers attempted to develop a method for analyzing and determining a system's compliance with privacy policies based on its design [32, 33].

As many data sharing events occur within the SSI model or SSI management system, information privacy cannot be neglected. Nonetheless, the SSI management system employs numerous distinct types of data sharing events, such as implicit data sharing with minimal and selective disclosure. Analyzing and determining the privacy level necessitates a customized method.

1.4 Objective and Significance

The main objective of this dissertation is to present a comprehensive approach for analyzing security weaknesses and privacy preservation in SSI management systems. This approach is designed to assist system analysts and designers in ensuring and enhancing their design's security and privacy. To achieve this objective, three sub-objectives have been established:

1. The first sub-objective is to provide a method for identifying SSI-specific weaknesses that inform analysts about design weaknesses.
2. The second sub-objective is to introduce a new set of SSI system properties, which enable implementers to govern their SSI management system while conforming to credible source documents.
3. The third sub-objective is to empower analysts to ensure their design's security and privacy with regards to data sharing, as well as provide them with instances of any violations within a specific scope.

The proposed approach in this dissertation, which incorporates domain knowledge and employs a combination of techniques, is significant for several reasons. Firstly, it helps to prevent overlooks that inexperienced analysts might experience when trying to enhance and ensure the security and privacy of their designs. Secondly, it overcomes limitations of existing analysis approaches when applied to specific systems, such as SSI management systems. Finally, it provides systematic coverage of domain knowledge in security and privacy analysis. This dissertation is confident that the proposed approach will effectively advance the current understanding in this field.

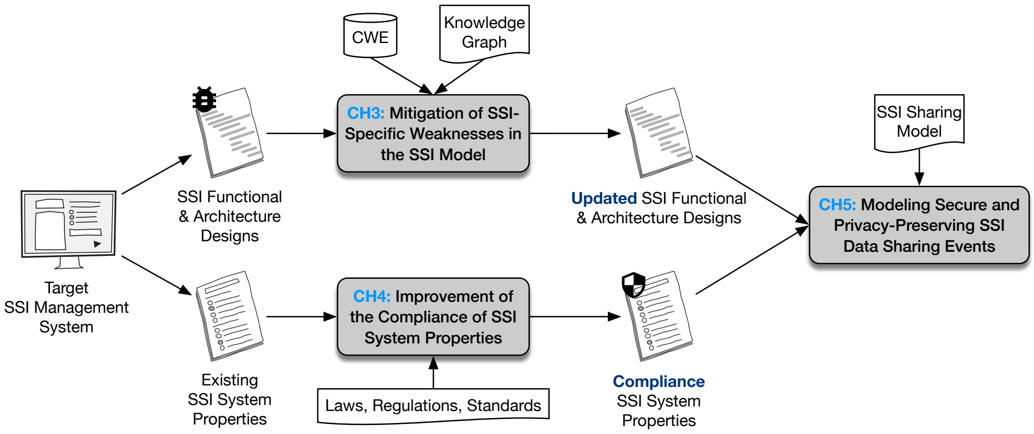

CWE
Knowledge Graph
CH3: Mitigation of SSI-Specific Weaknesses in the SSI Model
SSI Functional & Architecture Designs
Updated SSI Functional & Architecture Designs
SSI Sharing Model
CH5: Modeling Secure and Privacy-Preserving SSI Data Sharing Events
Target SSI Management System
CH4: Improvement of the Compliance of SSI System Properties
Existing SSI System Properties
Laws, Regulations, Standards
Compliance SSI System Properties

Figure 1.5: Overview of the proposed approach for security weakness and privacy preservation analysis.

1.5 Problem and Solution Overview

To achieve the intended objectives, this dissertation seeks to address three technical problems: **P1:** knowledge on SSI-specific security weaknesses has not been investigated yet and it is difficult to employ existing identification approaches to identify without domain knowledge; **P2:** existing SSI system properties are not universally consistent with a vast collection of credible laws, regulations, and standards to cover adequate governance of security and privacy; and **P3:** existing approaches to analyze data sharing events are incompatible to ensure unique security and privacy specifications of the SSI management system. As a result, a solution for each problem have to be researched thoroughly.

This dissertation intends that the proposed approach will be used by system analysts or designers after the design and architecture development phase. Consequently, at the moment of the use, a complete collection of design artifacts and deliverables can be accessible. However, this dissertation sets an assumption that security and privacy analysis on the provided design artifacts has not be comprehensively conducted. Therefore, the proposed approach in this dissertation composes of three solutions for addressing the three technical problems. A high-level overview of the proposed approach is illustrated in Fig. 1.5, indicating the mitigation of SSI-specific weaknesses, the improvement of SSI system properties, and the secure and privacy preserving SSI data sharing event modeling. An introduction to each solution will be provided in the following subsections.

Mitigation of SSI-Specific Weaknesses in the SSI Model

Various components of the target system, including functionality, design, and developed source code, may contain security weaknesses. To map common security weaknesses from online databases, a clear and justifiable correlation must be established with the target artifacts. In addition, domain expertise is required to identify security weaknesses. For instance, a common security weakness is the absence of authentication for critical functions, and determining which functions are critical requires domain knowledge. Based on this idea, this dissertation hypothesizes that the presence of SSI-specific weaknesses can be inferred from the language correlations between SSI functional requirements and descriptions of common security weaknesses from the CWE database when domain knowledge is provided. This dissertation proposes a cross-domain transfer knowledge graph between the domains of the SSI model and the CWE weakness, as well as the SSI Weakness Identification Framework (SWIF), which enables the automatic identification of language correlations using information retrieval techniques. This part will be described in detail in Chapter 3.

Improvement of the Compliance of SSI System Properties

The current source of security and privacy governance is SSI principles and system properties. However, the current SSI system properties do not appear to provide sufficient security and privacy protections. Existing proposals have attempted to improve security and privacy in accordance with a handful of credible source documents, such as the identity laws [7], and the General Data Protection Regulation (GDPR) [14]. This dissertation assumes that SSI system properties that comply with as many source documents as possible can be more advantageous for governing security and privacy in SSI management systems. Nonetheless, hundreds of laws, regulations, and standards were adopted and published. These source documents were defined in a different context than the SSI management system, making direct compliance with SSI system properties challenging. To improve the current SSI system properties, this dissertation proposes a method for systematically analyzing the consistency between the applicable source documents' shared content and the existing SSI system properties using domain knowledge. In addition, this dissertation proposes a new Compliance SSI System Property Set (CSSPS) based on the analysis results. This part will be described in detail in Chapter 4.

Modeling Secure and Privacy-Preserving SSI Data Sharing Events

Data sharing events are instantaneous instances of data transmission and access authorization. The sharing of data should take place in a secure and privacy-preserving manner. However, this type of event within the SSI management system is unique and subject to particular security and privacy constraints. The sharing of an identity claim, for instance, implicitly divulges partial information without knowledge. A constraint for this type of event is to ensure that the claim contains minimal or no information. On the basis of this concept and constraint, this dissertation assumes that data sharing events within the SSI management system are secure and privacy-preserving if they can guarantee that the specific constraints are met. Such constraints are related to the SSI management system's data model and can be satisfied by the design-level protocol. This dissertation proposes a method for modeling SSI data sharing events as a state transition system and encoding them in the Alloy specification language, which is adequate for predicating specific constraints and providing automatic and exhaustive model finding capability. The Alloy encoding will be analyzed by the Alloy Analyzer tool, which will provide insight to ensure that the design of data sharing events within a specified model scope is secure and privacy-preserving. This part will be described in detail in Chapter 5.

1.6 Organization of Dissertation

This dissertation consists of 8 chapters, organized as follows.

- **Chapter 1** introduces the SSI model, as well as its security and privacy features. This chapter also provides a brief overview of security and privacy analysis. Then, the objective, problems, and an overview of the proposed solution are described.
- **Chapter 2** provides an introduction to SSI management systems, weakness identification approaches, information retrieval and natural language processing techniques, knowledge graphs, legislative source documents, and the Alloy model finding techniques. Also, a brief overview of the employed techniques is described to enable readers to comprehend the dissertation.
- **Chapter 3** describes the development of the proposed SWIF for identifying SSI-specific security weaknesses using information retrieval and domain knowledge graph. Rationales behind the development of the proposed solutions, including the development of knowledge graph and additional features used, are described in detail.

- **Chapter 4** describes the proposed systematic method conducted to analyze security and privacy protections (i.e., controls) defined by laws, regulations, and technical standards in comparison with SSI system properties, as well as how they are used to improve the compliance of existing SSI system properties.
- **Chapter 5** describes in detail how the SSI data sharing events are modeled as a state transition diagram, how the model is encoded in the Alloy specification language, and how security and privacy properties are specified in the model. A method for using the Alloy Analyzer to identify models that adhere to the specification is also presented.
- **Chapter 6** presents an evaluation of each part of the proposed approach, as well as an evaluation of the proposed approach as a whole, in order to discuss their advantages and limitations in various aspects. The threats to the validity of the proposed approach are also discussed.
- **Chapter 7** provides comparisons between the proposed approach and related work in order to situate our work within the field. The results are also compared to similar work to highlight the benefits.
- **Chapter 8** concludes the research findings and technical contributions. In addition, limitations and future directions are discussed.

Chapter 2

Preliminaries

This chapter will provide preliminarily background on the techniques and approaches required to comprehend this research, including fundamentals of SSI management systems, component diagrams for indicating architectural designs, techniques for identifying weaknesses and security assurance, natural language processing, information retrieval, knowledge graphs, credible source documents, and model finding techniques.

2.1 Self-Sovereign Identity Model

2.1.1 Functionality of the SSI Model

With the emergence of the decentralized technologies, researchers have taken a greater interest in the concept of decentralized identity in order to eliminate the need of the central authority and mitigate risks of information leakage.

Christopher Allen presented his vision of *Self-Sovereign Identity (SSI)* concept back in 2016. An SSI model indicates an innovative identity model that restores full sovereignty and control of the digital identity's subject from the central authority by using blockchain technology [4]. With the adoption of the verifiable credential open standard [5], an overview of the SSI model can be illustrated as a contextual diagram in Fig. 2.1. The SSI model is defined with three interchangeable roles: holder, verifier, and issuer. A holder is a role that is responsible for holding the digital identities of one or more subjects using a locally-owned identity wallet. Not always is the subject of the digital identity the holder. A parent, for instance, may be the holder of their child's identity. The holder is able to generate an identity claim proving their digital identity. An example of an identity claim is a structured declaration confirming that the subject is older than 18 years old. Typically, identity

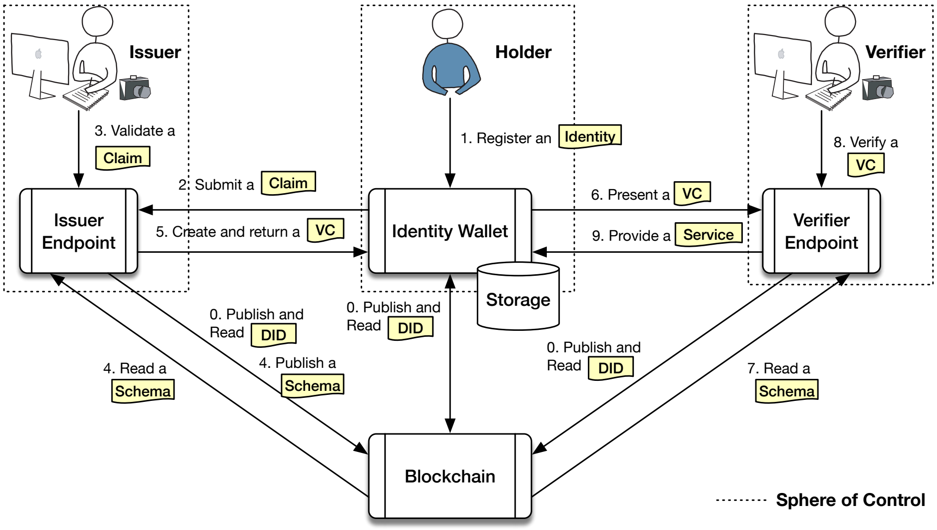


Figure 2.1: Overview of the SSI model depicting in a contextual diagram.

claims are expressed in a structured format, such as JSON objects. In some instances of the SSI model, identity claims may be generated using Zero-Knowledge Proofs (ZKPs). An issuer is either a human or a system entity tasked with validating the accuracy of identity claims. The issuer can inspect a holder's identity claim either offline or online. If the identity claim is valid, the issuer must convert it to a Verifiable Claim (VC) using a cryptographic proof schema published on blockchain technology. The primary function of the proof schema is to enable anyone to verify the validity of a verifiable claim. The verifiable claim will be returned to the corresponding holder, who can then use it for service authentication as a credential. A verifier is a system module that functions as a service provider. Requesting the holder's VC as credentials allows the verifier to authenticate and authorize services. On the blockchain, a VC can be verified against its corresponding proof schema.

In addition to the functions pertaining to the three interchangeable roles, the SSI model provides an additional vital security feature for initiating secure connections between components. The SSI model adopts the open standard Decentralized Identifier (DID) [6], which enables decentralized key management through blockchain technology. The DID standard augments the public-private key infrastructure by introducing a blockchain-based key exchange mechanism. Each component in the SSI model must be identified by a DID string (or DID address), such as `did:ssi:abcdef0123456789....` This DID string must include a one-to-one relationship with a public key. A DID document is a reference artifact that specifies a DID string and its corresponding public key, and it will be published on the blockchain. Using the

DID string as an identifier, each component in the SSI model can exchange their DID string with other components during their initial connection and resolve to the other DID document. The DID enables components to encrypt or verify the integrity of messages.

2.1.2 Minimal and Selective Disclosure

SSI data sharing is an SSI model event that occurs when system components share both typical data objects and personally identifiable information. Data sharing is significant because it may be the origin of PII disclosure. The two types of SSI data sharing events are raw disclosure and selective disclosure. Message encryption and integrity checking are necessary to protect data during raw data sharing events. On the other hand, the selective disclosure mechanism (e.g., zero-knowledge proofs [34]) should be utilized to minimize the amount of data disclosure during data sharing events. On the SSI model, the following four types of zero-knowledge proofs are applicable [7]:

- **Equality or Non-equality**. A magnitude's value is either equal to or not equal to a given value.
- **Inequality**. A magnitude's value is either greater than or less than a given value.
- **Member**. A subject can be found in a list or group.
- **Range**. Whether the value of a magnitude falls within a given interval.

A schema for identity claims that is appropriate for the target data should define selective disclosure mechanisms. For example, a user's age is 20 years old and the user can create an identity claim to attest that the age is in between 18 and 25 years old using the range type of zero-knowledge proof mechanism. This selective disclosure mechanisms minimize the exposure of sensitive information while proving the validity of the information. The concepts of the functionality of SSI management systems and the selective disclosure will be used throughout this dissertation.

2.1.3 Development of SSI Management Systems

The SSI model specifies the essential functionality of the system used to manage digital identities (as in Section 2.1). Consequently, an SSI management system must be developed in accordance with the essential functionalities, although it is not necessary to strictly limit the features. In other words, additional features can be implemented to provide more benefits and strengthen the developing software product in order to compete with others, such as introducing a hash database to securely store digital identities in the identity

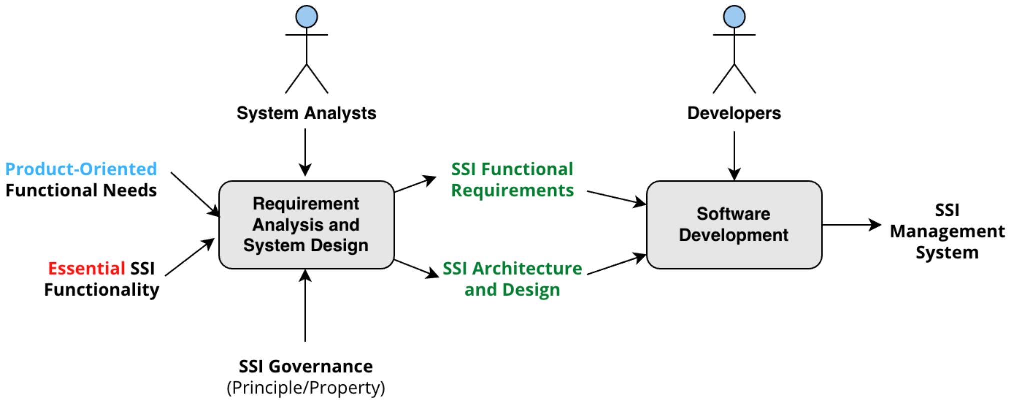

System Analysts
Developers
Product-Oriented Functional Needs
Essential SSI Functionality
Requirement Analysis and System Design
SSI Functional Requirements
SSI Architecture and Design
Software Development
SSI Management System
SSI Governance (Principle/Property)

Figure 2.2: Typical development process of the SSI management system.

wallet. This additional features is not defined by the SSI model, but it is still following the SSI notion.

The above-mentioned idea results in distinct implementations of the SSI management system, which sometimes pose distinct security and privacy risks. uPort [35], for instance, is an SSI management system that uses the inter-planetary file system to manage digital identities on a distributed network. However, the technology may raise privacy concerns regarding the identities of uPort users.

Traditional software development processes (such as waterfall and agile methods) can be used to deliver the final software product of the SSI management system to the market. Fig. 2.2 depicts a typical waterfall model for developing an SSI management system. Using the unique needs, essential SSI model functions, and SSI governance concept, system analysts will analyze and design the desired SSI management system. This activity will produce two important artifacts: SSI functional requirements and SSI architecture and design. In theory, these artifacts are required to adequately provide security and privacy protections, but in practice, it may be challenging to ensure such protections. With an iterative process model, SSI functional requirements, architecture, and design must be analyzed for functional completeness and other quality considerations.

This dissertation assumes that the security and privacy of the SSI management system can be derived from the SSI functional requirements, architecture, and design, which are intermediate development outcomes. These artifacts will serve primarily as input for security weakness and privacy preservation analysis. However, these artifacts are typically kept confidential within the development or implementation organization. It is difficult to obtain and use for this dissertation. In order to make knowledge of SSI functionality ac-

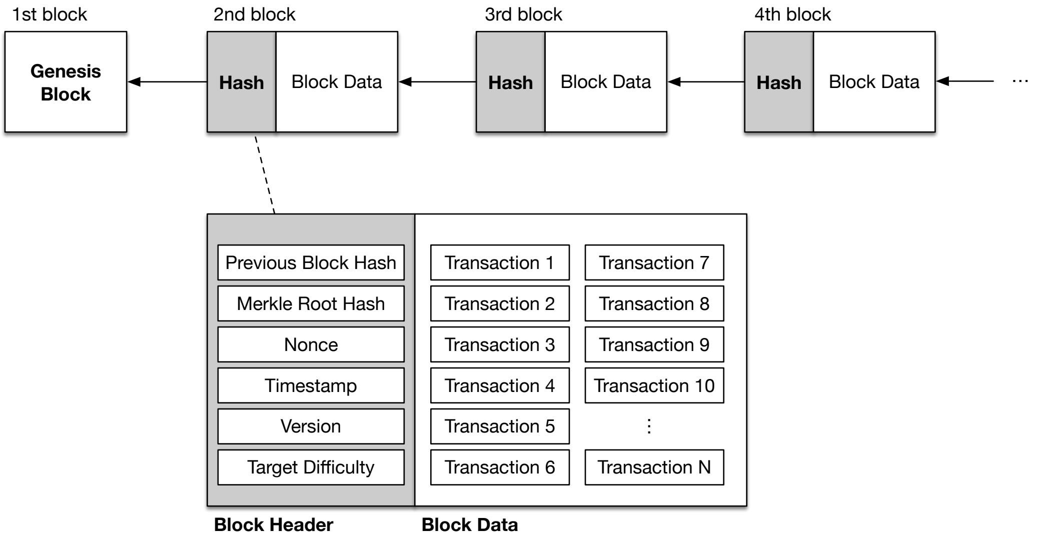


Figure 2.3: Overview of data blocks in the blockchain.

cessible, this dissertation considers online documentation and white papers as sources of SSI functionality and extracts SSI functional requirements, architecture, and design from these sources. In its online documentation [36], the IBM Verify Credentials product publishes its architecture and an explanation of the provided functions. The explanation can be used to derive the SSI functional requirements and the architecture can be taken to be the overview of the design for this SSI management system.

2.2 Distributed Ledger Technology (DLT) and Blockchain

The Distributed Ledger Technology (DLT) has been in use for years and is well-known. However, the Bitcoin trend has encouraged the development of a blockchain (a specific type of DLT) that specifies how transactional data are recorded and shared across nodes in a distributed network [37]. The DLT provides cryptographic mechanisms to ensure the decentralization and immutability of transactional data shared across the network.

A blockchain is a distributed ledger of data blocks containing transactional data, with each block linked to the previous block. Fig. 2.3 depicts an overview of data blocks in the blockchain. The overview demonstrates that the blockchain connects data blocks via their previous block's hash. These links make data blocks immutable. Within a peer-to-peer network, each node

Table 2.1: Comparison of different types of distributed ledgers.

Access Data 'read' data	Public			Private	
Submit Transactions	Unrestricted			Restricted or Unrestricted	
Validate Transactions 'write' data	Permission-less	Semi-permissioned		Permissioned	
Consensus	Proof-Based	BFT-Based & Others		BFT-Based & Others	
Data Structure	Blockchain	Blockchain	Non-Blockchain	Blockchain	Non-Blockchain
Example	Bitcoin, Ethereum	Stellar	Ripple	Quorum, Hyperledger	Corda

Note: BFT stands for Byzantine Fault-Tolerance.

will share a copy of the blockchain.

The DLT defines a mechanism to seek consensus from network nodes whenever a new data block is appended to the blockchain. The consensus mechanism will distribute an updated copy of the blockchain to all nodes and allow them to validate the copy. Typically, with the proof-of-work concept, nodes that successfully address the challenge and validate the copy are rewarded. If more than fifty percent of network nodes accept the copy, the updated blockchain will be synchronized across all nodes. This consensus mechanism combines the blockchain's immutability with its decentralization.

Multiple DLT types have been proposed to govern the distributed ledger in a variety of ways. In Table 2.1 from [38], a comparison table of distributed ledger types is presented. The table demonstrates that distributed ledgers permit data access from both public and private parties. Different data structures and consensus mechanisms have been defined for various DLT types. In the SSI management system, the type of DLT can be selected based on the objectives and implementation scopes. Hyperledger Indy is an example of a DLT that is specifically designed for identity data only.

Even though the SSI management system does not require extensive knowledge of blockchain and DLT, this dissertation must include such knowledge in an analysis of security weaknesses and privacy preservation. Specifically, blockchain and DLT knowledge must be utilized when formulating SSI functional requirements.

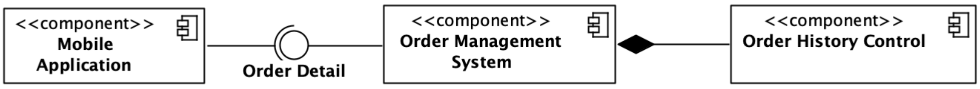


Figure 2.4: Example of a component diagram for the order management system.

2.3 Component Diagram and Analysis

In system and software modeling, diagrammatic representations are utilized to visualize various facets of the software system being modeled. Object Management Group (OMG) specified and standardized Unified Modeling Language (UML) [39, 40] for modeling object-oriented architecture and design. Several types of diagrams, including class diagrams, sequence diagrams, and component diagrams, were specified in the specification.

Component diagrams are used to represent the structural aspects of the target system. It is applicable at various levels of abstraction. For instance, it may represent a high-level design of system components and describe their interactive interfaces, or it may represent the structure of technical components that will be developed for the final software product. Fig. 2.4 displays an example of a component diagram.

According to the specification, a component diagram can be constructed using three fundamental graphical notations: components, interfaces, and relationships. Commonly, a component is depicted as a rectangle containing the component's name, icon, and stereotype. The order management system depicted in Fig. 2.4 is the component notation. An interface can be subdivided into required and provided interfaces, indicating the data object for which the component is required and, respectively, provided. The interface for order details in Fig. 2.4 indicates, for instance, that the mobile application provides order details to the order management system (which is required). A relation denotes the semantic relationship between components or an interface and a component. Fig. 2.4 indicates that components of the order management system and the order history control have a composition relationship, denoting that the order management system contains the order history control.

This dissertation uses component diagrams to illustrate the SSI architecture and design of SSI data sharing events. This work will prepare input for SSI architecture and design from SSI management system documentation and transform it into component diagrams.

2.4 Weakness Identification and Assurance

2.4.1 Typical Weakness Identification and Database

Identifying weaknesses is a part of analyzing security weaknesses. The primary objective of weakness identification is to map security weaknesses to target artifacts, such as source code or designs. However, no organizations or communities have standardized how security weaknesses are identified. The Common Weakness Scoring System (CWSS) [22] was developed to support the assignment of a severity score to each common security weakness listed in the CWE database based on the target system. However, the CWSS is manual and user-perspective.

Almost every proposed method for identifying security weaknesses makes reference to common security weaknesses documented in online databases. At the time of this study, four commonly used databases of weaknesses and vulnerabilities were discovered, namely:

- **Common Weakness Enumeration (CWE)** [22] is an online public database containing more than 1000 common security weaknesses. Each weakness is represented in the CWE database as an entry with multiple text fields. This database lists security weaknesses in hardware, software, and technology. It can be utilized in numerous fields.
- **Common Vulnerability Exposure (CVE)** [41] is a publicly accessible online database for accumulating vulnerability exposure. Since 1999, they have provided more than 100,000 instances of vulnerability exploitation. In each vulnerability exposure, a detailed description of the exploited event and references to the exploitation are provided.
- **National Vulnerability Database (NVD)** [24] is an online public database that collects CVE analysis results that provide a summary of severity scores based on the Common Vulnerability Scoring System (CVSS) [41], a vulnerability type (based on the CWE database), and an applicability statement.
- **Smart-contract Weakness Classification (SWC)** [25] is an online database that analyzes and collects smart contract-related security weaknesses from the CWE database. This database currently contains 36 security weaknesses and is regularly updated.

This dissertation only refers to security weaknesses from the CWE database, which contains both textual descriptions and source code examples. The CWE database is sufficient for our proposed framework for identifying security weaknesses, which seeks to correlate SSI functional requirements with weakness descriptions.

Table 2.2: Example of a weakness entry from the CWE database entitled “CWE-306: Missing Authentication for Critical Function” [22].

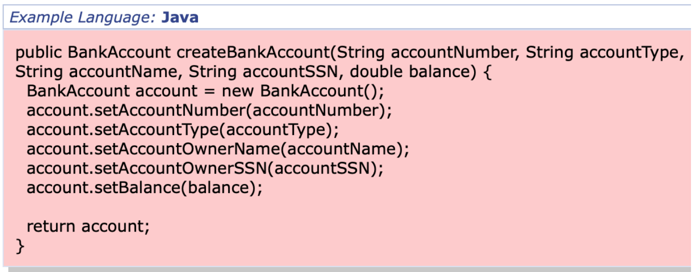
CWE-306: Missing Authentication for Critical Function

Description: The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.

Modes of Introduction: Architecture and Design

Common Consequences:
Scope - Access Control Other
Impact - Exposing critical functionality essentially provides an attacker with the privilege level of that functionality. The consequences will depend on the associated functionality, but they can range from reading or modifying sensitive data, access to administrative or other privileged functionality, or possibly even execution of arbitrary code.

Demonstrative Examples:
In the following Java example the method createBankAccount is used to create a BankAccount object for a bank management application.



Example Language: **Java**

```
public BankAccount createBankAccount(String accountNumber, String accountType,
String accountName, String accountSSN, double balance) {
  BankAccount account = new BankAccount();
  account.setAccountNumber(accountNumber);
  account.setAccountType(accountType);
  account.setAccountOwnerName(accountName);
  account.setAccountOwnerSSN(accountSSN);
  account.setBalance(balance);

  return account;
}
```

However, there is no authentication mechanism to ensure that the user creating this bank account object has the authority to create new bank accounts. Some authentication mechanisms should be used to verify that the user has the authority to create bank account objects.

A *weakness entry*, which is a collection of text fields describing how the weakness manifests itself in a software system, is a source of weaknesses that we obtain from the CWE database. Typically, it includes the weakness's name, description, extended description, common consequences, modes of introduction, applicable platforms, and demonstrative examples. Table 2.2 illustrates a weakness entry titled "CWE-306 Missing Authentication for Critical Functions." When analyzing this weakness entry, security analysts must justify whether the target system performs authentication for functionality that requires a provable identity (as indicated in the description text field) or whether it contains source code similar to the Java source code in the demonstrative example text field. In this dissertation, the description and extended description fields are utilized to represent security weaknesses and compute language correlations to SSI functional requirements. This dissertation claims that the description and extended description from the CWE database are sufficient to compute the language correlations, whereas the other databases for security weaknesses and vulnerabilities may not provide these text fields adequately.

2.4.2 Entity Authentication Assurance Framework

Identifying security weaknesses in SSI management systems aids in locating perceived risks. However, there may be an additional security risk that must be identified and mitigated. Entity authentication assurance framework was proposed by the telecommunication standardization section of the International Telecommunication Union (ITU) in Recommendation ITU-T X.1254 [42] in order to recommend controls based on entity authentication assurance levels and threat categories and to provide a broad aspect of security in the identity authentication process. Three levels of digital identity assurance are displayed in Table 2.3: identity assurance, authentication assurance, and federation assurance. These levels have varying concerns for a variety of controlled activity scopes. The authentication assurance level, for instance, focuses on authentication and credential management activities. On the basis of the levels of digital identity assurance, the recommendation defines three entity Authentication Assurance Levels (AALs) for the authentication assurance component only, as follows:

- **AAL1:** AAL1 authentication provides *some* assurance that the entity controls an authenticator bound to the entity's account and requires single-factor or multi-factor authentication.

Table 2.3: Levels of digital identity assurance [42]

Assurance Component	Related Activities
Identity Assurance. Robustness of the identity proofing process and the binding between the authenticator and the identity-proofed individual.	• Identity Proofing – Resolution, Validation, Verification • Enrollment • Binding
Authentication Assurance. Confidence that a given claimant is the same as the previously authenticated subscriber.	• Authentication • Credential Management – Credential Issuance, Suspension, Revocation, Destruction, Renewal, and Replacement
Federation Assurance. Combines aspects of the federation model, assertion protection strength, and assertion presentation.	• Key Management • Runtime Decisions • Attribute Management

- **AAL2:** AAL2 authentication provides *high* confidence that the entity controls an authenticator bound to the entity's account and provides proof of possession and control of two distinct authentication factors.
- **AAL3:** AAL3 authentication provides *very high* confidence that the entity controls an authenticator bound to the entity's account, provides proof of possession and control of two distinct authentication factors, is based on proof of possession of a key, and uses a hardware-based cryptographic authenticator that provides impersonation resistance.

This recommendation provides controls for five categories of threats to authentication in order to achieve the desired AAL: authenticator compromise, transaction compromise, credential service provider impersonation, entity impersonation, and authentication service compromise. For instance, a control for the authenticator compromise threat is defined as "AC-1: For the highest AAL, authentication should use a hardware-based cryptographic authenticator and an authenticator that provides resistance to verifier impersonation; the same device can satisfy both requirements."

This recommendation's assurance framework is well-known and should be mentioned when conducting a security analysis of the SSI management system. In general, security weaknesses in the SSI management system should correspond to the threat categories outlined in this recommendation. This

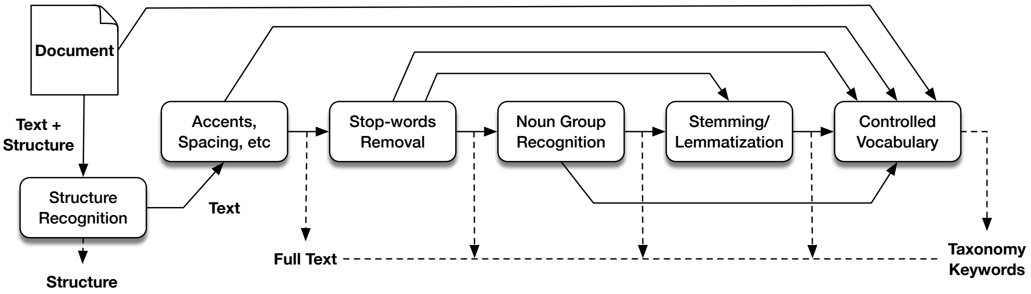


Figure 2.5: Logical overview of a document and the text preprocessing tasks defined in [44].

dissertation employs the security weakness analysis to cover all assurance components; however, the controls cannot be applied to the property improvement because they are too specific to entity authentication and some controls are incompatible with the SSI concept. For instance, the requirement for a cryptographic authenticator based on hardware may not be compatible with the decentralized nature of the SSI management system.

2.5 Natural Language Processing

Natural Language Processing (NLP) is a subfield of computer science and computational linguistics that enables computers to analyze and process human spoken language from text [43]. NLP has multiple applications in various domains. For instance, speech recognition, machine translation, and sentiment analysis of social media. To process and analyze human language with NLP, text inputs must undergo a number of tasks, including preprocessing, feature extraction, and similarity measures. The following subsections will explain each task in detail.

2.5.1 Text Preprocessing

Human spoken language is creative and may contain ambiguity that interferes with computer analysis and processing; as a result, it must be preprocessed to remove such interferences. Typically, text preprocessing tasks are designed to convert text documents into a set of index terms, which is a vocabulary collection representing the document's unique presence. Fig. 2.5 depicts a typical and logical overview of the text preprocessing tasks [44]. Details of each preprocessing task can be described in detail as follows.

- **Structure Recognition**. The purpose of this task is to locate and remove the document's structure. This task produces unstructured and schema-free texts. For instance, when the document is represented as an eXtensible Markup Language (XML) document with topic and chapter tags, this task removes those tags and leaves only the values contained within the removed tags. This task also requires removing all figures, tables, and other non-text elements.
- **Accent, Spacing, etc**. This task is intended to remove special characters (such as exclamation marks) and spaces (also known as text segmentation or tokenization). This task assists in condensing lengthy texts into a collection of words or sentences. By using whitespace and periods as delimiters, for example, English texts can be tokenized into a list of words.
- **Stop-words Removal**. Stop-words are commonly used but meaningless words in written texts, such as the English articles "a," "an," and "the." As a result of their multiple occurrences, these stop words can affect the outcome of text analysis and processing as well as the overall performance. This task is to remove all stop words from texts. For example, "a cat plays with a dog" is preprocessed to "cat plays dog."
- **Noun Group Recognition**. Noun groups or noun chunks are groups of terms that carry significant meaning and must be processed together, such as "critical function," which should be processed as a single term rather than as two separate nouns. The objective of this task is to analyze a list of words and observe meaningful noun groups.
- **Stemming/Lemmatization**. Stemming is a strategy for reducing words to their root terms, irrespective of tense or context. For instance, the words "historical" and "history" are shortened to "histori." Stemming may produce meaningless terms. Lemmatization, on the other hand, is another strategy to reduce words to their lemmas regardless of tense and context. Lemmas are meaningful root word (based on a dictionary). This task is to reduce terms by stemming or lemmatization. For example, the words "requires" and "required" are shortened to "require." In certain instances, stemming and lemmatization should be incorporated with the Part-of-Speech (POS) tagging to prevent certain errors, such as "understanding" being reduced when it is a noun.
- **Controlled Vocabulary**. A controlled vocabulary is a collection of words and phrases used to index and retrieve content through browsing or searching [44]. When the corpus contains documents, the controlled vocabulary contains all terms that appear in the corpus. This task entails analyzing each document and generating the controlled vocabulary accordingly.

These text preprocessing tasks are recommended, but not required. Other tasks, such as n-gram or word-sense disambiguation, may be utilized if they are appropriate for the dataset. This dissertation prepares texts from SSI functional requirements and weakness descriptions using text preprocessing in order to eliminate unnecessary contents and structures that influence the computation of language correlations.

2.5.2 Feature Extraction and Vector Space Model

In addition to the connotations of the controlled vocabulary, another task in NLP is feature extraction, which aims to find a feature representing of each document. Typically, a document's feature is a collection of keywords or terms that distinguish it from other documents, such as a vector space model, the prevalent numerical representation in NLP.

Several approaches to represent text documents in vector space models have been proposed recently. The Bag-of-Words (BoW) model is a weighting scheme to represent the document by using the controlled vocabulary as vector dimensions and encoding the occurrence of words in the document as a vector [45]. Suppose a corpus of documents $D = \{d_1, \ldots, d_j, \ldots\}$ is provided. The weight of a term i for the BoW count vector can be calculated using the following equation:

$$w_{i,d_j} = f_{i,d_j} \tag{2.1}$$

where w_{i,d_j} denotes the weight of a term i in a document d_j and f_{i,d_j} denotes the raw count of a term i in a document $d_j \in D$. Moreover, Term Frequency and Inverse Document Frequency (TF-IDF) is another weighting scheme to determine the importance of words based on the frequency with which they appear in a document [46]. The weight of a term for the TF-IDF vector can be calculated using the following equation:

$$w_{i,d_j} = tf_{i,d_j} \cdot idf_i = (1 + \log f_{i,d_j}) \cdot \log \frac{N}{df_i} \tag{2.2}$$

where w_{i,d_j} denotes the weight of a term i in a document $d_j \in D$, $tf_{i,d_j} = (1 + \log f_{i,d_j})$ denotes the normalized term frequency where f_{i,d_j} indicates the raw count of a term i in a document d_j, $idf_i = \log \frac{N}{df_i}$ denotes the inverse document frequency, N denotes the number of document in the corpus D, and df_i denotes the number of documents containing a term i.

Text representation should be determined by the document corpus' characteristics. If a corpus contains a medium-sized document set, the BoW and TF-IDF weighting schemes may be superior to other machine learning approaches, such as word embedding and global vectorization. This dissertation employs the TF-IDF weighting to represent textual descriptions of common security weaknesses and SSI functional requirements because it is the most appropriate for such data.

2.5.3 Similarity Measure

As a vector space model, term vectors that are represented by different weight scheme can be utilized to determine partial matching from others based on the degree of similarity [44]. Several distance functions can be utilized to compute the degree of similarity between two vectors, such as Euclidean similarity or cosine similarity.

In NLP, similarity measures have been used to compute how two term vectors of document are semantically similar to each other. Suppose we have two term vectors $\vec{d_1} = (w_{1,d_1}, w_{2,d_1}, \ldots, w_{n,d_1})$ and $\vec{d_2} = (w_{1,d_2}, w_{2,d_2}, \ldots, w_{n,d_2})$. The degree of similarity based on the angular distance can be calculated by using the cosine similarity function [44] as follows.

$$cosine\_sim(\vec{d_1}, \vec{d_2}) = \frac{\vec{d_1} \cdot \vec{d_2}}{|\vec{d_1}| \times |\vec{d_2}|} = \frac{\sum_{i=1}^{n} w_{i,d_1} \times w_{i,d_2}}{\sqrt{\sum_{i=1}^{n} w_{i,d_1}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,d_2}^2}} \tag{2.3}$$

The cosine similarity function $cosine\_sim(\vec{d_1}, \vec{d_2})$ determines the degree of similarity between two document term weight vectors where $w_{i,d_1} > 0, w_{i,d_2} > 0$, and $0 < cosine\_sim(\vec{d_1}, \vec{d_2}) < 1$. If the similarity is close to 1, two documents are semantically similar (according to the occurrence of terms). For example, similarity scores between three documents (d_1, d_2, and d_3) can be calculated as: $cosine\_sim(\vec{d_1}, \vec{d_2}) = 0.67$ and $cosine\_sim(\vec{d_1}, \vec{d_3}) = 0.15$. This example indicates that the first document (d_1) is more relevant to the second document (d_2) than the third document (d_3). This similarity measure can be applied to any kind of vector space models, but it should also be tested to the suitability to the target dataset. In this dissertation, the degree of similarity between common security weaknesses and SSI functional requirements is computed using the cosine similarity measure.

2.6 Information Retrieval

NLP is a technique for handling textual data in a way that is comprehensive for computers. NLP's capabilities can be applied to a variety of theoretical and practical domains. Information Retrieval (IR) is an application domain of NLP concerned with the representation, storage, organization, and acces-

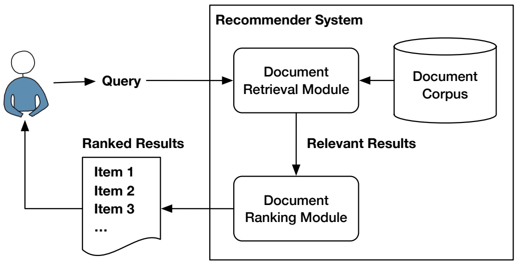


Figure 2.6: Overview of typical recommender systems.

sibility of textual information assets. The representation and organization of information assets should facilitate user access to those assets [44].

A Recommender System (RS) or a recommendation system is an alternative term for a particular implementation of information retrieval (also known as a search engine) that is used to recommend pertinent information assets based on the user's needs or queries [43]. In Fig. 2.6, an overview of a typical recommender system model is presented. It has two primary modules: document retrieval and ranking. The purpose of the document retrieval module is to retrieve relevant documents from the corpus in response to a given query. The document ranking module, on the other hand, is designed to place the documents that are most pertinent to the user's needs at the top of the resulting list. Some filtering approaches have also been used to fine-grain the resulting results based on user's preferences or contents. Technically, the recommender system will create an index of documents in the corpus and access a query from users. Then, the query will be vectorized and compute its similarity score to every document in the corpus. For instance, if the cosine similarity measure is utilized, a similarity score between a query vector ($\vec{q}$) and a document vector of index terms ($\vec{d}$) can be calculated as follows.

$$cosine\_sim(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{n} w_{i,q} \times w_{i,d}}{\sqrt{\sum_{i=1}^{n} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,d}^2}} \tag{2.4}$$

Typically, the nature of users will interest in only some top results that relevant to their query. As a result, the similarity score of each document will be used to rank which documents are highly relevant to the given query.

A Cross-Domain Recommender System (CDRS) is an extended type of RS that utilizes information or preferences transferred from one source domain to recommend information assets from a different domain [47]. For instance, books are recommended in accordance with movie preferences [48]. The CDRS is useful when two distinct domains share common relationships, and it makes more precise recommendations without requiring preferences or knowledge of the target domain. Technically, interrelationships between features of two domains are predefined in a data model, such as matrix or graph. For example, suppose the first domain has a feature set $F_A = \{f_{A,1}, \ldots, f_{A,p}\}$ and the second domain has a distinct feature set $F_B = \{f_{B,1}, \ldots, f_{B,q}\}$. interrelationships between features in two domains can be predefined in the following weighted matrix.

$$M_{\text{transfer}} = \begin{bmatrix} 0 & 0.451 & 0.112 & \ldots & 0.667 \\ 0.112 & 0 & 0 & \ldots & 0.125 \\ 0.125 & 0.667 & 0.412 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0.333 & 0 & \ldots & 0 \end{bmatrix} \tag{2.5}$$

The preceding matrix of knowledge transfer (M_{transfer}) indicates, for instance, that the first feature of the first domain ($F_{A,1}$) is irrelevant to the first feature of the second domain ($F_{B,1}$), represented by $m_{1,1} = 0 \in M_{\text{transfer}}$. These predefined interrelations can be utilized in a variety of ways. For instance, the CDRS can influence the interrelationships when calculating a similarity score or use them to conduct a knowledge expansion in the target domain feature set.

It is realized that the SSI model belongs to a different domain than security weaknesses. Nevertheless, knowledge of the SSI model can be correlated with common security weaknesses to support the identification. This dissertation will utilize cross-domain knowledge transfer to implement a CDRS-based recommendation of SSI-specific weaknesses.

2.7 Knowledge Graph

A knowledge graph is a graph-based data model that captures knowledge about entities and their relationships [49]. Kroetsch and Weikum, as guest editors for the journal of web semantic, share a definition of the knowledge graph on the website[1] as:

> "Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities."

Three basic terms are necessary for employing knowledge graphs: nodes, edges, and facts. Nodes represent knowledge entities of interest. Edges are relationships between entities. A fact is the knowledge of a relationship

[1] Journal of Web Semantics: Special Issue on Knowledge Graphs: https://iccl.inf.tu-dresden.de/web/JWS_special_issue_on_Knowledge_Graphs/en

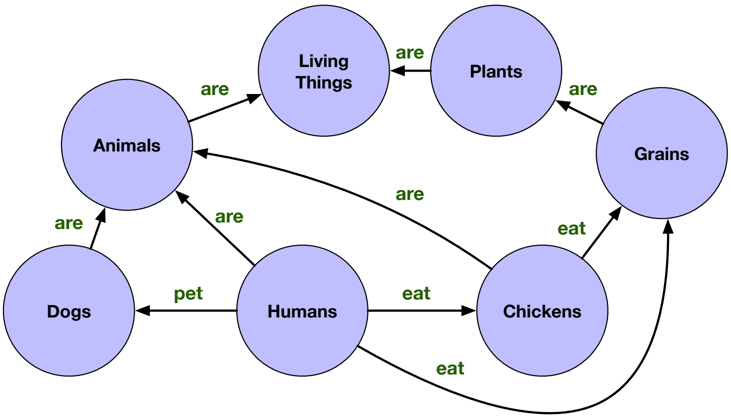


Figure 2.7: Example of a knowledge graph representing relationships among living things.

between two nodes with a particular edge type. In Fig. 2.7, an example of a knowledge represents that chickens eat grains, for instance. Further that. knowledge graph were defined formally in [50], as follows.

Definition 2.7.1. A knowledge graph is a graph-based data model characterized by a triple $KG = \langle E, R, F \rangle$, in which:

- E denotes a set of entity types that represent graph nodes;
- R denotes a set of relation types that link entity nodes in the graph;
- F denotes a set of fact so that $F \subseteq E \times R \times E$.

According to the above definition, a knowledge graph KG can be represented by three sets: entities E, relations R, and facts F. Referring to Fig. 2.7, this graph can be represented as: $E = \{\text{Animals, Humans, Dog,} \ldots\}$, $R = \{\text{are, eat, pet}\}$, and $F = \{(\text{Humans, pet, Dogs}), \ldots\}$. The fact (Humans, pet, dogs) can be interpreted as humans pet dogs.

In comparison to other graph-based data models, such as ontologies, knowledge graphs lack sophisticated querying and searching capabilities, but they provide simplicity and quick access. This dissertation will use the knowledge graph to establish cross-domain knowledge transfer by predefining interrelationships between specific term entities in the SSI model domain and generic term entities in the common security weakness domain.

2.8 Laws, Regulations, and Standards

A law is a legally binding written document that is enacted by parliament. It has the authority to regulate, authorize, prohibit, provide (funds), sanction,

Table 2.4: Examples of information security and privacy controls in a law, a regulation, and a standard.

A Law	**A Regulation**	**A Standard**
PIPL [51]: *Security* - Processors of personal information shall be accountable for their information processing activities, and implement necessary measures to ensure the security of the information that they process.	**GDPR** [13]: *Article 5.1.(b)* Personal data shall be collected for specified, explicit, and legitimate purposes and not further processed in a manner that is incompatible with those purposes.	**ISO/IEC 27001** [52]: *User Registration* - There should be a formal user registration and de-registration procedure in place for granting and revoking access to all information systems and services.

Note: PIPL = Personal Information Protection Law of the Mainland.

award, proclamation, and restrict. Everyone must adhere to the law in order to be legal. A regulation is a detailed instruction for carrying out the law. Typically, regulations are obligatory and include penalties for violations. In contrast, a standard is an established requirement or norm for products, services, systems, or processes. If applicable standards are consistently applied, certain quality aspects can be ensured. Standards are not requirements, but in some instances they are necessary to conduct business.

Several industries, from manufacturing to information technology, have utilized laws, regulations, and standards. These documents focus primarily on quality control, security, safety, and privacy. The laws, regulations, and standards governing information security and privacy are fundamental documents applicable to modern information systems. Typically, they define *a control*, a criterion or item on a checklist that indicates the target system's conformance with the applicable law, regulation, or standard. In Table 2.4, three controls from a law, a regulation, and a standard are exampled.

Table 2.4 shows different controls from three documents: the Personal Information Protection Law of the main land (PIPL) [51], General Data Protection Regulation (GDPR) [13], and ISO/IEC 27001:2013 standard [52]. It can be seen that controls are defined in evaluable, concise statements. This dissertation refers to such a statement as *an endorsed task* that must not be disregarded or should be concerned with. For laws and regulations, all controls must be implemented, whereas it is recommended that controls be implemented to meet the standards.

This dissertation examines laws, regulations, and standards as knowledge sources for the SSI system's properties' improvement. Laws, regulations, and standards will be referred to as *source documents* in the following content.

2.9 Model Finding

Model finding, also known as model discovery, is a formal automated method for instantiating system models that guarantee facts or essential properties. Typically, model finding techniques transform a model into a Conjunctive Normal Formula (CNF) and use a SATisfiability (SAT) solver to determine the satisfiability of facts and properties.

2.9.1 Alloy Modeling and Specification Language

Alloy is a declarative specification language that is lightweight, precise, and tractable [53]. Alloy describes constraints and behaviors in the target system model using first-order relational logic and describes components in the model using specific syntaxes. The grammar[2] of the Alloy specification language is denoted in Backus-Naur Form (BNF), as shown in Fig. 2.8.

The grammar is described using the following standard BNF operators as well as the additional operators:

- x* means zero or more repetitions of x;
- x+ means one or more repetitions of x;
- x | y means a choice of either x or y;
- [x] means an optional x;
- x,* means zero or more comma-separated occurrences of x; and
- x,+ means one or more comma-separated occurrences of x.

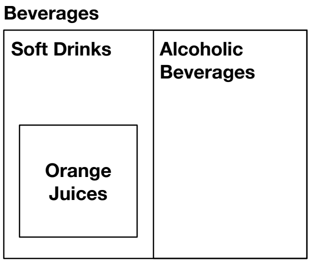
A *signature* is initially used to describe an entity atom of interest. To increase the signature's inheritability, it can be declared as an *abstract signature*. Fig. 2.9 illustrates the distinctions between the extension of abstract signature and generic signature.

Fig. 2.9a depicts the scope of the entity atoms with distinct signature declarations from the Alloy code block depicted in Fig. 2.9b. It demonstrates that all beverages are either soft drinks or alcoholic beverages, whereas no other inheritances exist. Orange juice, on the other hand, is a type of soft drink because it is in the soft drink signature. However, this inheritance is distinct from the first in that additional soft drink varieties may be further declared in the Alloy code block.

[2] Specifications of the Alloy 6 language. https://alloytools.org/spec.html.

```
alloyModule ::= [moduleDecl] import* paragraph*
moduleDecl ::= module qualName [[name,+]]
import ::= open qualName [[qualName,+]] [as name]
paragraph ::= sigDecl | factDecl | predDecl | funDecl
    | assertDecl | cmdDecl
sigDecl ::= [var] [abstract] [mult] sig name,+ [sigExt] {fieldDecl,*}[block]
sigExt ::= extends qualName | in qualName [+ qualName]*
mult ::= lone | some | one
fieldDecl ::= [var] decl
decl ::= [disj] name,+ : [disj] expr
factDecl ::= fact [name] block
predDecl ::= pred [qualName .] name [paraDecls] block
funDecl ::= fun [qualName .] name [paraDecls] : expr { expr }
paraDecls ::= ( decl,* ) | [ decl,* ]
assertDecl ::= assert [name] block
cmdDecl ::= [name :] ( run | check ) ( qualName | block ) [scope]
scope ::= for number [but typescope,+] | for typescope,+
typescope ::= [exactly] number qualName
expr ::= const | qualName | @name | this
    | unOp expr | expr binOp expr | expr arrowOp expr
    | expr [ expr,* ]
    | expr [! | not] compareOp expr
    | expr ( => | implies ) expr else expr
    | let letDecl,+ blockOrBar
    | quant decl,+ blockOrBar
    | { decl,+ blockOrBar }
    | expr '
    | ( expr ) | block
const ::= [-] number | none | univ | iden
unOp ::= ! | not | no | mult | set | # | ~ | * | ^
    | always | eventually | after | before | historically | once
binOp ::= ∨ | or | ∧ | and | ≤> | iff | => | implies |
    & | + | - | ++ | <: | :> | . | until | releases | since | triggered | ;
arrowOp ::= [mult | set] -> [mult | set]
compareOp ::= in | = | < | > | =< | =>
letDecl ::= name = expr
block ::= { expr* }
blockOrBar ::= block | bar expr
bar ::= |
quant ::= all | no | sum | mult
qualName ::= [this/] ( name / )* name
```

Figure 2.8: Grammar of the Alloy specification language denoted by the standard BNF operators.



(a) Scope of signature.

```
abstract sig Beverages {}
sig SoftDrinks extends Beverages {}
sig AlcoholicBeverages extends Beverages {}
sig OrangeJuices in SoftDrinks {}
```

(b) Example of Alloy's signatures

Figure 2.9: Differences between abstract signatures and generic signatures.

Relations to other signatures can be declared in the block of a signature using a *field* syntax. In the following code block, the relation that a package (or set) of orange juices is contained in the refrigerator at a given time is declared.

```
sig OrangeJuices in SoftDrinks {}
one sig Refrigerator { contain: set OrangeJuices -> Time }
```

Then, always-presumed invariant constraints on the entity atom can be declared as first-order relational logics in the *fact* syntax. For instance, all orange juice be refrigerated is mandated. The following Alloy code block can be used to declare the fact.

```
fact { all o : OrangeJuices | some t: Time | o->t in Refrigerator.contain }
```

The *predicate* syntax is a named declaration of zero or more argument-required first-order relational logics. In the following Alloy code block, a predicate to ensure that the orange juice is in the refrigerator before it can be consumed can be declared.

```
pred drinkCold [o : OrangeJuices, t: Time]{o -> t in Refrigerator.contain}
```

The *function* syntax is a named expression containing zero or more arguments and capable of returning a value in response to the given relational logic. In the following Alloy code block, a function to randomly retrieve the brand of orange juice from the predefined brand signatures is declared.

```
abstract sig JuiceBrands {}
one sig Sweeties, Fresheny, Mirando extends JuiceBrands {}
fun getBrand [o: OrangeJuices]:JuiceBrands {Sweeties + Fresheny + Mirando}
```

Then, the *enumeration* syntax can be declared to predefine singleton values that can be assigned or used in field, predicate, fact, and function syntaxes. In the following Alloy code block, the state of the beverages as an enumeration of cold, warm, and hot values, can be declared.

```
enum BeverageState { HOT, WARM, COLD }
abstract sig Beverages { state: one BeverageState }
```

The *assertion* syntax is intended to be a constraint that follows from the models' implicit facts. In the more recent version of the Alloy specification language, the assertion syntax can be used to determine whether the Alloy model conflicts with the predefined specifications. For instance, orange juices from the Sweeties brand must always be refrigerated and in a cold state. The following Alloy code block can be used to declare this assertion.

```
assert SweetiesIsCold {
        all o: OrangeJuices, t: Time | o.state = COLD and
        o -> t in Refrigerator.contain
}
```

A tool called the Alloy Analyzer can analyze and execute the Alloy code blocks with various commands. The Alloy Analyzer offers two features applicable to Alloy code blocks or Alloy models: the finding of model instances and the validation of assertions.

A *run* command to instruct the Alloy Analyzer to find model instances that comply with the predefined facts and predicates can be declared. The run command is necessary to specify the model's scope using the *for* syntax. For instance, the following run command to find model instances that are consistent with the "NotFullRefrigerator" property within the scope of 15 for each signature is declared.

```
pred NotFullRefrigerator { #Refrigerator.contain < 10 }
run NotFullRefridgerator for 15
```

The "NotFullRefrigerator" property is declared as a predicate to restrict the number of beverages in the refrigerator to 10. Using the *exact* syntax, the scope of each signature individually can be specified. For instance, the scope of the following run command to five orange juices and fifteen other signatures is declared.

```
run NotFullRefrigerator for 15 but exactly 5 OrangeJuices
```

On the other hand, we can declare the *check* command to only validate the predefined assertion within the specified scope. The check command will only report the assertions' validity. As counterexamples, the Alloy Analyzer will provide model instances that violate the assertions only when the assertions are invalid. For instance, the following check command can be declared to validate the "SweetiesIsCode" assertion for 5 orange juices and 15 other signatures.

```
check SweetiesIsCold for 15 but exactly 5 OrangeJuices
```

```
abstract sig Color {}
one sig Red, Yellow, Green extends Color {}
fun colorSequence (): Color->Color {
        Color<:iden + Red->Green + Green->Yellow + Yellow->Red }
sig Light {}
sig LightState {color: Light->one Color}
sig Junction {lights: set Light}
fun redLights (s: LightState): set Light {s.color.Red}
pred mostlyRed (s: LightState, j: Junction) {
        lone j.lights - redLights[s] }
pred trans (s, s': LightState, j: Junction) {
        lone x: j.lights | s.color[x] != s'.color[x]
        all x: j.lights |
                let step = s.color[x]->s'.color[x] {
                        step in colorSequence
                        step in Red->(Color-Red)=>j.lights in redLights[s]
                }
assert Safe { all s,s': LightState, j: Junction |
        mostlyRed[s,j] and trans[s,s',j]=>mostlyRed[s',j] }
check Safe
```

Figure 2.10: Example of the Alloy model for the traffic light [53].

To further clarify the usage of the Alloy specification language, another example of a traffic light model encoded in Alloy specification language is demonstrated in Fig. 2.10. The model define an abstract signature for `Color`, which is latterly extended to `Red`, `Yellow`, and `Green`. The `colorSequence` function declares how the color can be changed from one to another. Additionally, some predicates are declared to define light transition constraints. With the "Safe" assertion, we can determine if the model is safe. This example shows how to model a system using Alloy model. However, it is evident that some Alloy-related keywords and syntaxes, such as *iden* and *lone*, are used, but we do not employ them in this dissertation.

2.9.2 Alloy Analyzer Tool

Massachusetts Institute of Technology (MIT) developed the Alloy Analyzer model finding tool [53]. This tool converts an Alloy model into a CNF and uses a SAT Solver, such as SAT4J or miniSAT, to determine whether the CNF is satisfiable. As mentioned earlier, one of the purposes of the Alloy Analyzer is to identify model instances that are valid to the predefined facts, predicates, and assertions. If the Alloy model depicted in Fig. 2.10 is ran to check the "Safe" property, the Alloy Analyzer will generate an output message indicating the validity of the specified assertion. The instantiation of a system model that adheres to the constraints is another function. If the `check` command at line 20 is replaced with the `run` command (e.g., `run {} for 5`), the resultant model instances will be represented in graph

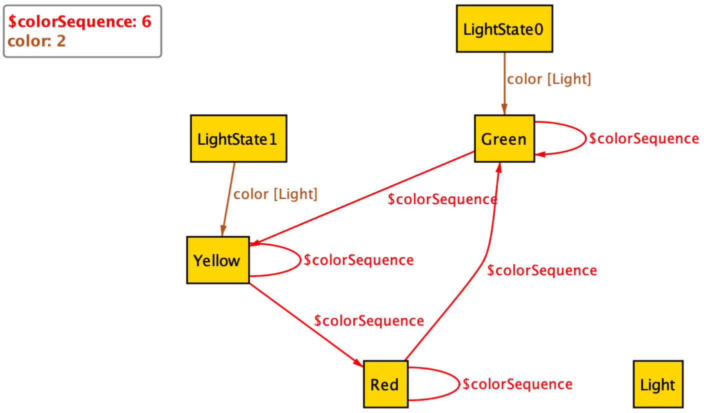


Figure 2.11: Example of a model instance for the traffic light.

models, as an example shown in Fig. 2.11. In the model instance, it can be observed that two light states are initiated. As declared in the predicates and functions of the Alloy model, the sequence of light states is satisfiable with the specification because the green light changes to yellow. With more complex Alloy models, facts, predicates, and assertions can be analyzed more thoroughly in a given model scope.

In this dissertation, the Alloy specification language and the Alloy Analyzer are utilized to define secure and privacy-preserving SSI data sharing models. The Alloy model can aid in exhaustively ensuring the security and privacy properties of SSI data sharing events.

Chapter 3

Mitigation of SSI-Specific Weakness in the SSI Model

This chapter will present an automatic framework for identifying SSI-specific weaknesses, which is intended to infer knowledge of common security weaknesses associated with the SSI model. The proposed framework is designated to empower system analysts in analyzing security weaknesses in the functionality of the developing SSI management system. This chapter will consist of three sections. Initially, a detailed description of the proposed framework's activities is provided. Second, the development of the cross-domain transfer knowledge graph for expanding knowledge based on interrelationships between common security weaknesses in the CWE database and the SSI model concept. Lastly, an innovative semi-automatic recommendation procedure is described for inferring knowledge of SSI-specific weaknesses by utilizing information retrieval techniques.

3.1 A Framework for Identifying SSI-Specific Weakness

As mentioned in Section 2.4.1, typical weakness identification maps weakness descriptions or code examples to the artifacts of the target system. Security weaknesses can occur as early as the earliest phases of development. This dissertation assumes that functional requirements of the SSI management system that are incomplete, incorrect, or absent may contain SSI-specific security weaknesses. This dissertation intends to infer SSI-specific weaknesses, with the support of domain knowledge, based on language correlations between CWE common security weaknesses and SSI functional requirements. Section 3.1.1 will describe how SSI functional requirements are collected and prepared. Then, Section 3.1.2 will provide an overview and detailed explanations of each proposed framework module.

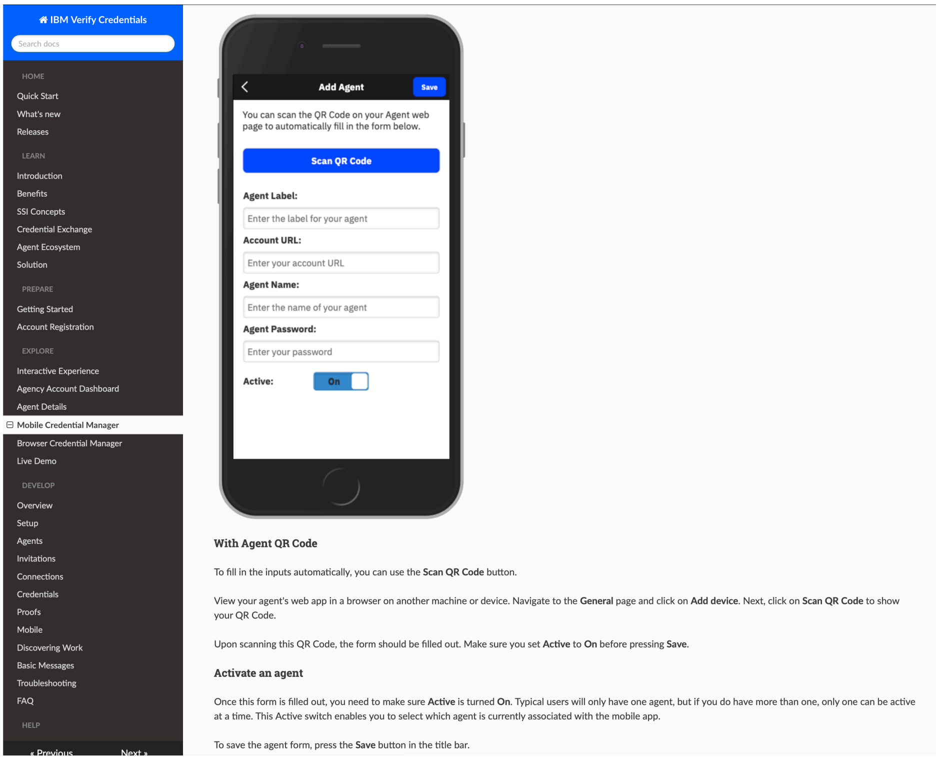
3.1.1 Preparation of SSI Functional Requirements

According to the idea to infer SSI-specific weaknesses, SSI functional requirements are artifacts that the proposed framework will be used to identify language correlations to CWE common security weaknesses. SSI management systems may contain different functionalities even if the functionalities are still based on the SSI model's notion. To prepare an input of SSI functional requirements, it is expected that system analysts who suppose to be the user of the proposed framework have to extract the functional requirements of the developed SSI management system or elicit the ones for the developing SSI management system.

However, in the time of this work, it is difficult to access to either developed or developing SSI management systems. This dissertation considers that SSI functional requirements can be manually extracted from online documentations or white papers of the developed SSI management system, which are usually explained the functionality for their users. A screenshot of the online documentation of the IBM Verify Credentials product is shown in Fig. 3.1. It can be seen in the figure that the documentation explains how users can activate their cloud agent in corresponding to their architecture (Fig. 3.2). An SSI functional requirement can be extracted from the above explanation as "The SSI management system shall allow users to activate their cloud agent in the identity wallet by turning the Active switch on." By analyzing this kind of documentation, a collection of SSI functional requirements can be prepared in correspondence with the target SSI management system and used as an input for the proposed framework.

3.1.2 An Overview of the SSI Weakness Identification Framework

This dissertation proposes an automatic framework known as the *SSI Weakness Identification Framework*, or *SWIF*, that employs information retrieval and NLP techniques to recommend common security weaknesses from the CWE database that have language correlations with a set of SSI functional requirements. As depicted in Fig. 3.3, the proposed SWIF defines a suggested procedure for implementing a recommender system to report SSI-specific weaknesses. In addition, the overview illustrates that the proposed SWIF consists of three main modules: text preprocessing, knowledge expansion, and recommendation modules.



You can scan the QR Code on your Agent web page to automatically fill in the form below.

Agent Label: Enter the label for your agent

Account URL: Enter your account URL

Agent Name: Enter the name of your agent

Agent Password: Enter your password

Active: On

With Agent QR Code

To fill in the inputs automatically, you can use the **Scan QR Code** button.

View your agent's web app in a browser on another machine or device. Navigate to the **General** page and click on **Add device**. Next, click on **Scan QR Code** to show your QR Code.

Upon scanning this QR Code, the form should be filled out. Make sure you set **Active** to **On** before pressing **Save**.

Activate an agent

Once this form is filled out, you need to make sure **Active** is turned **On**. Typical users will only have one agent, but if you do have more than one, only one can be active at a time. This Active switch enables you to select which agent is currently associated with the mobile app.

To save the agent form, press the **Save** button in the title bar.

Figure 3.1: Screenshot of the online documentation of the IBM Verify Credentials product [36].

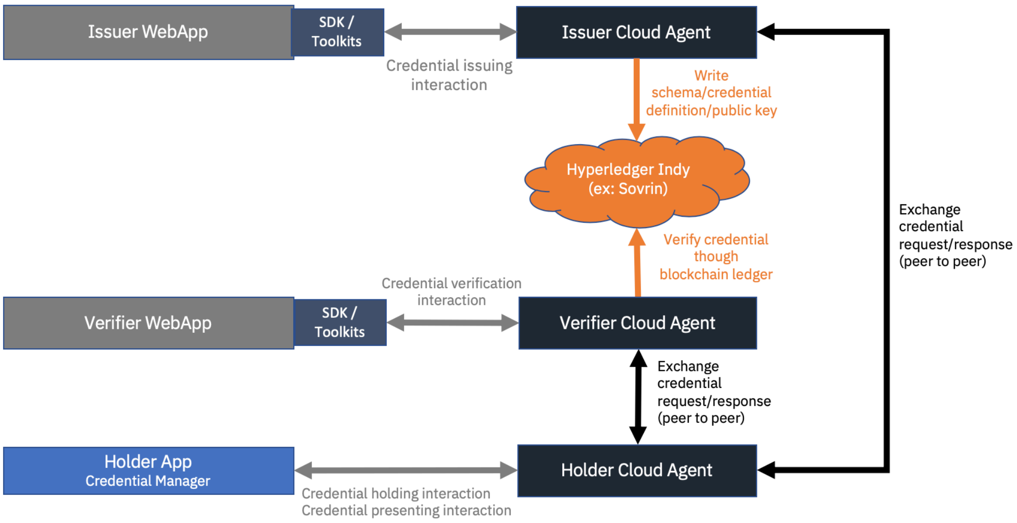

Issuer WebApp
SDK / Toolkits
Credential issuing interaction
Issuer Cloud Agent
Write schema/credential definition/public key
Hyperledger Indy (ex: Sovrin)
Verify credential though blockchain ledger
Exchange credential request/response (peer to peer)
Credential verification interaction
Verifier WebApp
SDK / Toolkits
Verifier Cloud Agent
Exchange credential request/response (peer to peer)
Holder App
Credential Manager
Holder Cloud Agent
Credential holding interaction
Credential presenting interaction

Figure 3.2: Architecture of the IBM Verify Credentials product from the documentation [36].

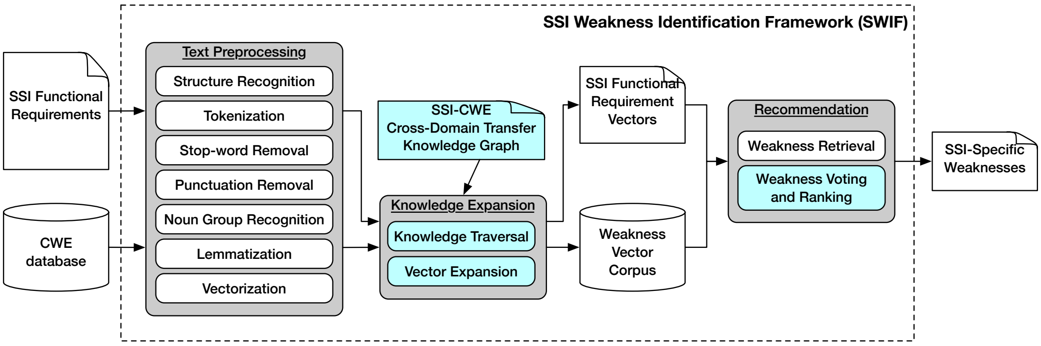


Figure 3.3: Overview of SWIF illustrating in a contextual diagram of three modules.

The *text preprocessing* module is responsible for representing textual data using NLP techniques as term weight vectors. Manually preparing two machine-readable files of the input SSI functional requirements and CWE weakness descriptions in an accessible directory and structured format is required. As described in Section 2.5.1, a number of text preprocessing tasks are suggested for preparing and transforming textual data into numerical models, in this case term weight vectors. This module will automatically transform the manually prepared input files into two output files containing term weight vectors. The file containing the SSI functional requirement vectors will be referred to as the query, and the file containing the CWE weakness description vectors will be referred to as the weakness vector corpus. Based on our trials, this module's automatic preprocessing is suggested to include six suitable tasks: structure recognition, tokenization, noun group recognition, punctuation removal, lemmatization, and TF-IDF vectorization. These tasks are conventional and required no extensions. Fig. 3.4 depicts an example of preprocessing the CWE-306's description (illustrated in Table 2.2), and the details of each task are as follows:

1. The *structure recognition* task is designated to eliminate these tags, leaving only text. This task also entails the elimination of unnecessary data, such as the `id` value. Suppose weakness descriptions were manually prepared in XML format from the CWE database. Each weakness is structured using tags, including the `id`, `name`, and `description` tags. However, these tags are not necessary for the recommendation and should be removed.
2. The *tokenization* task is designed to split the remaining text into a collection of words. Since different numbers of distinct words are used, these occurrences can be used to index each text data. For exam-

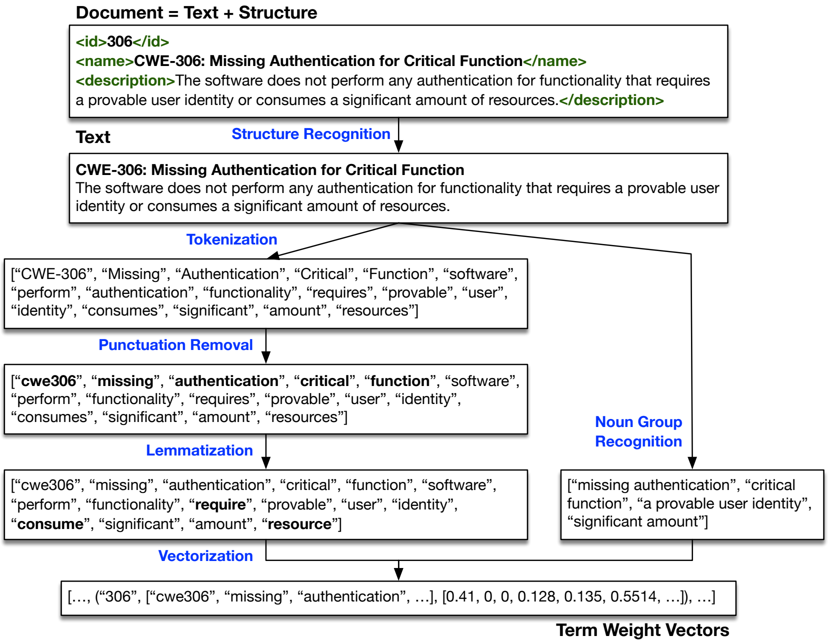

Document = Text + Structure
<id>306</id>
<name>CWE-306: Missing Authentication for Critical Function</name>
<description>The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.</description>
Text
Structure Recognition
CWE-306: Missing Authentication for Critical Function
The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.
Tokenization
[“CWE-306”, “Missing”, “Authentication”, “Critical”, “Function”, “software”, “perform”, “authentication”, “functionality”, “requires”, “provable”, “user”, “identity”, “consumes”, “significant”, “amount”, “resources”]
Punctuation Removal
[“cwe306”, “missing”, “authentication”, “critical”, “function”, “software”, “perform”, “functionality”, “requires”, “provable”, “user”, “identity”, “consumes”, “significant”, “amount”, “resources”]
Lemmatization
[“cwe306”, “missing”, “authentication”, “critical”, “function”, “software”, “perform”, “functionality”, “require”, “provable”, “user”, “identity”, “consume”, “significant”, “amount”, “resource”]
Noun Group Recognition
[“missing authentication”, “critical function”, “a provable user identity”, “significant amount”]
Vectorization
[..., (“306”, [“cwe306”, “missing”, “authentication”, ...], [0.41, 0, 0, 0.128, 0.135, 0.5514, ...]), ...]
Term Weight Vectors

Figure 3.4: Example of the preprocessing of the CWE-306 weakness's description through six suggested tasks.

ple, the text data of the CWE-306 contain words, including “Missing,” “Authentication,” “software,” etc. Typical library can be used to implement automatic tokenization in this module.

3. The *punctuation removal* task is designed to remove punctuation and other distractions. Punctuation is a group of symbols that includes commas, dashes, and exclamation marks. The following regular expression can be used to define the punctuation symbols to be removed: `r‘[/(){}\[\]\|@,;]’`. Other distractions include the case-sensitive nature of English text, such as “Function” and “function,” which distinguishes words. This module can employ a string processing library to remove punctuation and lowercase every word in the collection.
4. The *lemmatization* task is responsible for analyzing and transforming words into their root form. In most cases, the form of words is altered based on context and tense, such as when the term “require” is changed to “requires” when the singular third-person subject is used in the present simple tense or to “required” when the text is in the past tense. This module can utilize a lemmatization library, such as Python's NLTK, to transform every word in the collection into its root form automatically.

5. The *noun group recognition* task is intended to identify meaningful noun groups or chunks in the text. In some instances, noun groups are required to identify language correlations to the SSI functional requirements and are more effective than word splitting. For instance, the phrase “critical function” is meaningful and can be supported as a correlation with SSI functional requirements, as opposed to two split words (i.e., “critical” and “function”). This module can utilize a common NLP library for recognizing noun groups, such as noun chunk objects from the Spacy library, to automatically locate meaningful noun groups in text.
6. The *vectorization* task is tasked with converting a collection of words representing a textual data object into a term weight vector. All words present in every text data object will be compiled as the controlled vocabulary, which will be used as the term weight vector’s dimensions. For example, the controlled vocabulary is an array [“software”, “data”, “authentication”, ...] and the term weight vector of a weakness description can be represented as [0.111, 0, 0.785, ...], indicating that the weight of the term “software” in the weakness description is 0.111. This module will automatically vectorize text data using a common weighting scheme from a library, such as the `TfIdfVectorizer` module from NLTK. According to the observations made in this work, TF-IDF is suggested to be the suitable weighting scheme for SSI functional requirements and CWE weakness descriptions.

Since the size of the CWE database and the number of SSI functional requirements are moderate, the suggested preprocessing tasks are appropriate for the target datasets (i.e., report the best performance). The typical NLP techniques are adequate for this scope, as opposed to the deep learning approaches that require a massive dataset. This module applies an existing approach and does not demonstrate the originality of the proposed SWIF.

The *knowledge expansion* module is responsible for expanding the knowledge in term weight vectors from the knowledge transfer between two domains: the SSI management system and the CWE common security weakness. This module is based on the notion that domain-specific terms in SSI functional requirements could be predefined with their interrelationship to generic terms in CWE weakness descriptions. This module is designed to use domain-specific terms to expand the controlled vocabulary and re-weight the terms based on their interrelationships with generic terms. The SSI-CWE cross-domain transfer knowledge graph is created to support this module as a representation of predefined interrelationships between terms based on explicit domain knowledge. In Section 3.2, the development of the knowledge graph is described in detail, and in Section 3.3, the innovative automatic knowledge expansion method underlying this module is described.

functional requirement vector and each CWE weakness description vector in the corpus, this module shares the same concept. This module can automatically compute the similarity score using a vector processing library, such as the `cosine_similarity` function from the SKLearns library. This dissertation argues, however, that an SSI-specific weakness may have language correlations to multiple SSI functional requirements and may have a significant impact on the target SSI management system. As a result, this dissertation assumes that the SSI-specific weaknesses that correlate to multiple requirements are regarded as severe because they relate to numerous SSI management system components. This group of weaknesses should be addressed and mitigated at first. In order to select only the most severe SSI-specific weaknesses when time and effort are limited, this module introduces an automatic, innovative voting mechanism. A collection of automatic algorithms is described in detail in Section 3.4.

As a result of the proposed SWIF, a collection of severe SSI-specific weaknesses are reported automatically based on language correlations. These weaknesses must be mitigated in the design and functionality of the target SSI management system in order for the results to be utilized. This dissertation also proposes a manual procedure to effectively support the mitigation of identified SSI-specific weaknesses based on the cause of the weakness as described in Section 3.5.

3.2 SSI-CWE Cross-Domain Transfer Knowledge Graph

This section will describe about the SSI-CWE cross-domain transfer knowledge graph that predefines interrelationships between SSI-specific and CWE generic terms. To make the development of the knowledge graph systematic, this dissertation uses the formalization to apply the notion of typical knowledge graph and uses a systematic analysis to develop the knowledge graph from the existing domain knowledge. In Section 3.2.1, the formalization of the proposed knowledge graph is explained, whereas, in Section 3.2.2, the development process of the knowledge graph is described in detail.

3.2.1 Formalization of Knowledge Graph

A triple $\langle E, R, F \rangle$ can be used to represent sets of entities, relations, and facts, respectively, according to the formal definition of knowledge provided in Definition 2.7.1. In accordance with the existing idea, the formal definition of the SSI-CWE cross-domain transfer knowledge graph will be presented in a similar fashion. Initially, the proposed knowledge graph seeks to establish connections between two domains. Thus, entities from both domains must be incorporated into the knowledge graph. Suppose set E_{SSI} is a set of SSI-specific words or word groups defined in the SSI management system's functionality, and set E_{CWE} is a set of system-related generic words or word groups defined in CWE weakness descriptions. There are relations between these two sets of words or word groups. This dissertation identifies the aforementioned interrelation as the "relate_to" relation and places it in the set R. However, this dissertation proposes that both SSI-specific and CWE-generic words or word groups may be changed by their synonyms because they are manually defined and the choice of words or word groups is up to the writer. To reduce this subjectivity, this dissertation proposes that synonyms of every word or group of words be added to the knowledge graph, and the set E_{SYN} is created to collect the added synonyms. This type of interrelation is classified as a "synonym_of" relationship and is also included in the set R. The types of relations in the set R are indirected and will be fixed for only the two specified.

This dissertation permits facts in the set F that will be represented by the knowledge graph to only two categories: (1) interrelationships between two domains in which SSI-specific and CWE generic words or word groups are related, and (2) interrelationships between entities that are synonyms of each other. On the basis of this justification, this dissertation provides the following formal definition for the SSI-CWE cross-domain transfer knowledge graph:

Definition 3.2.1. An SSI-CWE cross-domain transfer knowledge graph is a graph-based data model for visualizing the predefined interrelationships between the CWE weakness domain and the SSI management system domain, denoted by a triple $KG_{SSI \leftrightarrow CWE} = \langle E, R, F \rangle$ where:

- $E = E_{SSI} \cup E_{CWE} \cup E_{SYN}$ denotes a set of entity nodes, combining the set of SSI-specific words or word groups E_{SSI}, the set of CWE generic words or word groups E_{CWE}, and the set of their synonyms E_{SYN};
- $R = \{relate\_to, synonym\_of\}$ denotes a set of relation edges between nodes; and
- $F = \{f | f = (p, relate\_to, q)$, s.t. $p, q \in E_{CWE} \cup E_{SSI}\} \cup \{f | f = (p, synonym\_of, q)$, s.t. $p \in E_{CWE} \cup E_{SSI}$, and $q \in E_{SYN}\}$ denotes a set of facts that show knowledge from the two relation types among groups of words or word groups.

The SSI-CWE cross-domain transfer knowledge graph can be developed based on the above definition by defining the domain knowledge-based sets $E_{SSI}, E_{CWE}, E_{SYN}$, and F. For instance, the word group "identity wallet" is an element of the set E_{SSI}, and the word "software" is an element of the set E_{CWE}. With domain knowledge, it appears that these two elements are interrelated, and that fact ("identity wallet", relate_to, "software") will be included in the set F. With the above definition, it is possible to develop and include all the necessary information in the proposed knowledge graph.

3.2.2 Development of Knowledge Graph

To provide a comprehensive cross-domain knowledge graph for influencing the weakness identification, sufficient knowledge from the interesting domains is necessary to be sufficiently collected. In the case of the SSI model and CWE weakness domains, knowledge on the words and word groups should be collected and prepared properly. This section will describe how domain knowledge from the two domains are collected, analyzed, and interrelated.

The development process is designated to be semi-automatic and depends on real data from domain-specific sources. This dissertation organizes the development process of the SSI-CWE cross-domain transfer knowledge graph in four steps, as follows.

1. In order to formulate the representation of the SSI domain, this dissertation suggests that SSI-specific words and word groups must be collected from real-world SSI management systems' documentation. Four white papers explaining features and functionalities of the EverID, uPort, LifeID, and IBM Verify Credentials products are collected[1]. 367 sentences are collected, preprocessed, and broken down into distinct 1322 words and word groups.
2. Words and word groups collected from the previous step are not always SSI-specific, but they are just words or word groups that are used to describe the SSI management system's function. In order to find the actual SSI-specific words or word groups, this step is used to manually select only words or word groups that are directly relevant to the SSI model's notion. For example, the word group "identity wallet" is relevant, but the word "cryptocurrency" is not. At this step, 138 words or word groups are collected in the set E_{SSI}.
3. On the other hand, textual weakness descriptions that explain weaknesses on requirements, architecture, design, and policy phases are col-

[1] Collected explanations on SSI features: https://doi.org/10.5281/zenodo.7423717.

Table 3.1: Subset of predefined interrelationships between CWE generic terms and SSI specific terms.

CWE Generic Terms	**SSI-Specific Terms**
application, software, system, ...	SSI management system, identity wallet, endpoint, issuer endpoint, verifier service endpoint, service provider, blockchain, ...
user, owner, actor, ...	holder, issuer, verifier, administrator, ...
object, resource, information, data, ...	personal information, SSI, identity attribute, identity claim, credential, DID, DID document, ...

lected directly from the CWE database[2]. 464 entries of weakness are collected, preprocessed, and broken down into distinct 3728 words and word groups, which will be stored in the set E_{CWE}.

4. Interrelationships between SSI-specific (E_{SSI}) and CWE generic (E_{CWE}) words and word groups are identified manually by the authors based on three criteria:

   (a) Words or word groups shares the same meaning to one another.
   (b) One word or word group is in a subset or superset of one another.
   (c) Words or word groups are justifiable to be related based on domain knowledge and specificity.

   The manually identified interrelationships are collected as facts in the set F to develop the knowledge graph. Examples of interrelationships identified are shown in Table 3.1.

5. Synonyms of each word and word group contained in the sets E_{SSI} and E_{CWE} are determined by using a NLP library with the WordNet corpus, and 695 synonymous words are identified and added into the set E_{SYN} with corresponding facts into the set F.

As a result of the above-mentioned development process, this dissertation is able to develop the SSI-CWE cross-domain transfer knowledge graph to transfer knowledge between the two domains, as shown in Fig. 3.5. The statistical information of the proposed knowledge graph is also provided in Table 3.2. The knowledge graph demonstrates, for example, that the "application" entity is related to the "software" entity through the "*synonym_of*" relation edge. This results in the fact that $f_1 =$ (application, *synonym_of*, software) $\in F$. An additional instance is a fact between the "system"

[2] Collected weakness description: https://doi.org/10.5281/zenodo.7423741.

Table 3.2: Statistical information of the SSI-CWE cross-domain transfer knowledge graph.

#Node	#SSI-specific term	#CWE generic term	#Synonym	#Fact
252	134	84	34	244

Note: Full version is in https://github.com/chnpat/SWIF-Implementation.

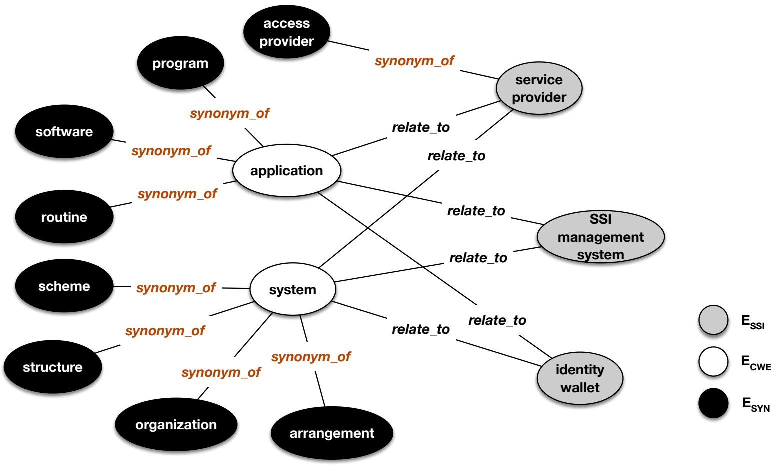


Figure 3.5: Part of the SSI-CWE cross-domain transfer knowledge graph indicating component-related interrelationships.

and "identity wallet" entities denoted by f_2 = (system, *relate_to*, identity wallet) $\in F$, which illustrates the interrelationships between an SSI-specific word group and a CWE generic word.

The SSI-CWE cross-domain transfer knowledge graph is useful for identifying the relative terms between two domains that could impact the identification of weaknesses. First, the proposed knowledge graph provides mappings between terms that pertain to SSI-specific components and terms commonly employed in a generic context. In Fig. 3.5, for example, the SSI-specific term for the "identity wallet" component corresponds to the generic term "application." When an analyst maps a description containing the word "application," the analyst should look for the identity wallet component of the SSI management system. The predefined interconnections between functionalities is another advantage of the proposed knowledge graph. Fig. 3.6, for instance, depicts a part of the proposed knowledge graph that relates SSI-specific functionality to the terms used in the weakness description, such as critical function or privileged function. These interrelationships facilitate the process of identifying weaknesses by selecting which SSI-specific functionality is related when analyzing the weakness that refers to critical functions.

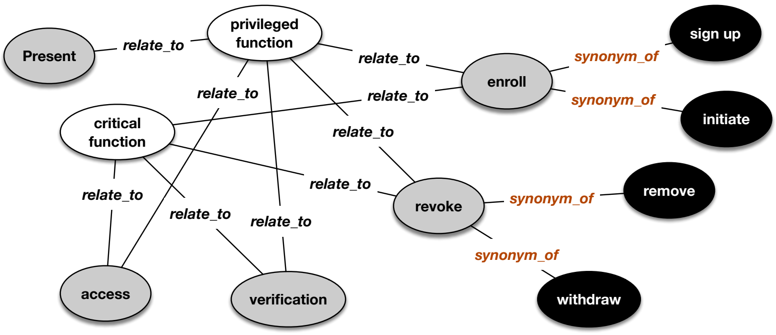


Figure 3.6: Part of the SSI-CWE cross-domain transfer knowledge graph indicating functional interrelationships.

3.3 Knowledge Expansion Method

With the SSI-CWE cross-domain transfer knowledge graph, a knowledge transfer capability between two domains can be established. To integrate with the recommendation module of SWIF, however, these pieces of knowledge must be adequately incorporated into term weight vectors. This section will describe the proposed method for embedding interrelated words or word groups within term weight vectors and expanding knowledge that influences the recommendation, as shown in Algorithm 1.

By incorporating related terms into the controlled vocabulary, Algorithm 1 seeks to expand knowledge in the weakness vector corpus. The controlled vocabulary and weakness vector corpus are updated based on the following inputs: the initial controlled vocabulary, the initial weakness vector corpus, and the SSI-CWE cross-domain transfer knowledge graph. First, two empty sets for the updated controlled vocabulary ($V_{expanded}$) and the updated weakness vector corpus ($D_{expanded}$) are initiated. The set of facts (F) will then be extracted from the knowledge graph ($KG_{SSI \leftrightarrow CWE}$) using the RETRIEVEFACT() function. The expansion will iterate through each term (v) in the controlled vocabulary (V) and compare such terms to each fact $(e_a, r, e_b) \in F$ in the set. At line 6, each controlled vocabulary term v will be compared to the entity nodes e_a, e_b in the knowledge graph. If the term is present in the knowledge graph, the UPDATEVOCABULARY() and REWEIGHT() functions will add the related term from the fact to the updated controlled vocabulary and recalculate the term weight vectors based on the new term's relationship. The UPDATEVOCABULARY() function will add the term "program" to the vector dimensions if the term "application" is in the controlled vocabulary and has a *synonym_of* relationship with the term "program" that is not currently in the controlled vocabulary. The *CallReweight* function will then search

Algorithm 1 Knowledge expansion procedure for embedding cross-domain knowledge into term weight vectors

```
Input:
    $V$ : controlled vocabulary,
    $D$ : weakness vector corpus, and
    $KG_{SSI \leftrightarrow CWE}$ : SSI-CWE cross-domain transfer knowledge graph.
Output:
    $V_{expanded}$ : updated controlled vocabulary, and
    $D_{expanded}$ : updated weakness vector corpus.
procedure KNOWLEDGEEXPAND($V, D, KG_{SSI \leftrightarrow CWE}$)
    $V_{expanded} \leftarrow \emptyset, D_{expanded} \leftarrow \emptyset$
    $F \leftarrow$ RETRIEVEFACT($KG_{SSI \leftrightarrow CWE}$)
    for $v \in V$ do
        for $(e_a, r, e_b) \in F$ do
            if $e_a = v$ or $e_b = v$ then
                $V_{expanded} \leftarrow$ UPDATEVOCABULARY($e_a, e_b$)
                $D_{expanded} \leftarrow$ REWEIGHT($v, V_{expanded}, D$)
    return $(V_{expanded}, D_{expanded})$
```

all term weight vectors in the corpus for vectors that have a weight for the term “application” and update the weight for the term “program” for those vectors. This procedure will then return the updated controlled vocabulary and updated weakness vector corpus.

Algorithm 1 is crucial for transferring knowledge from CWE weaknesses to the SSI management system context. It enables the automatic comparison of common security weaknesses in the CWE database and SSI functional requirements without requiring domain knowledge or expertise. Algorithm 1 may be executed concurrently and automatically with the text preprocessing module to reduce redundancy.

3.4 Weakness Recommendation Algorithms

Given that the main objective of SWIF is to identify SSI-specific security weaknesses based on language correlations, it is necessary to retrieve and rank common security weaknesses that are specific to the functionality of the SSI management system. With the text preprocessing and knowledge expansion modules, a corpus of expanded CWE weakness vectors is obtained. Information retrieval techniques can now be used to determine language correlations based on SSI functional requirements input. The following sub-sections will

Algorithm 2 Retrieval procedure for calculating cosine similarity between queries and weakness vectors

```
Input:
    $r$ : an SSI functional requirement,
    $x$ : a threshold indicating the acceptable number of top results, and
    $D_{expanded}$ : the expanded weakness vector corpus.
Output:
    $D_{retrieved,\vec{r}}$ : a set of relevant weakness vectors for the requirement $r$.
procedure RETRIEVEWEAKNESS($r, x, D_{expanded}$)
    $\vec{r} \leftarrow$ VECTORIZE($r$)
    $D_{retrieved,\vec{r}} \leftarrow \emptyset$
    for $(id_{\vec{d}}, v_{\vec{d}}, \vec{d}) \in D_{expanded}$ do
        $sim_{\vec{r},\vec{d}} =$ COSINESIMILARITY($\vec{r}, \vec{d}$)
        $D_{retrieved,\vec{r}} \leftarrow D_{retrieved,\vec{r}} \cup \{(\vec{r}, (id_{\vec{d}}, v_{\vec{d}}, \vec{d}), sim_{\vec{r},\vec{d}})\}$
    $D_{retrieved,\vec{r}} \leftarrow$ GETTOPRESULT($x, D_{retrieved,\vec{r}}$)
    return $D_{retrieved,\vec{r}}$
```

describe both retrieval and ranking algorithms for weakness identification through language correlations in detail.

3.4.1 Weakness Retrieval Algorithm

A recommendation system will typically receive a query and calculate similarity scores across the entire document corpus. Nevertheless, it can be realized that, based on the typical behavior of users, only a few of the most relevant results would be of interest. Due to this idea, the retrieval procedure is intended to target only a subset of the most relevant results.

The proposed retrieval procedure for SWIF is displayed in Algorithm 2. The RETRIEVEWEAKNESS() procedure retrieves the most x relevant weaknesses from the corpus based on the following inputs: an SSI functional requirement (r), a threshold indicating the acceptable number of top results (x), and the expanded weakness vector corpus ($D_{expanded}$). First, the procedure prepares an empty set of relevant weakness vectors ($D_{retrieved,r}$). The input SSI functional requirement (r) must be vectorized using the expanded controlled vocabulary, resulting in a vector ($\vec{r}$) with the same dimensions as those in the weakness vector corpus. Then, each triple of the weakness vector $(id_{\vec{d}}, v_{\vec{d}}, \vec{d})$ consisting of a weakness ID ($id_{\vec{d}}$), a collection of terms representing the weakness ($v_{\vec{d}}$), and the weakness vector ($\vec{d}$) will be iterated and its similarity score $sim_{\vec{r},\vec{d}}$ against the SSI functional requirement vector ($\vec{r}$) will be calculated using the cosineSimilarity() function. At line 6, the calculated similarity score $sim_{\vec{r},\vec{d}}$, its corresponding SSI functional requirement vector $\vec{r}$, and a weakness vector triple $(id_{\vec{d}}, v_{\vec{d}}, \vec{d})$ are retrieved and added to the results collection ($D_{retrieved,r}$). Before returning the retrieved results, the getTopResult() function will filter the top results with the highest similarity score according to the specified threshold x. At the conclusion of this procedure, each SSI functional requirement will be represented by an equal number of relevant weaknesses.

3.4.2 Weakness Voting Algorithm

Due to the fact that the typical recommendation system only accepts a query, this dissertation argues that the target SSI management system cannot be represented by a single SSI functional requirement. As a result, the query for the weakness identification is a collection of multiple SSI functional requirements. It is possible that some weaknesses may be related to multiple SSI functional requirements. In addition, weaknesses that relate to multiple SSI functional requirements may demonstrate their severity and should be addressed first. On the basis of this justification, this dissertation hypothesized that the weight of each weakness relative to all SSI functional requirements can be calculated and use this weight to determine which weaknesses should be prioritized. This section will describe the proposed weakness voting procedure for selecting addressed-first SSI-specific weaknesses.

The proposed automatic ranking procedure is depicted in Algorithm 3. The rankWeight() procedure selects SSI-specific weaknesses with an acceptable weight based on the following inputs: a collection of SSI functional requirements (R), a collection of retrieved weaknesses ($D_{retrieved}$), a collection of expanded weakness vectors ($D_{expanded}$), and a threshold indicating an acceptable weakness weight (y). The collection of weaknesses retrieved $D_{retrieved}$ is a merged version of the outcomes of Algorithm 2 for all SSI functional requirements in R. At first, the procedure iterates over all weakness vectors in the collection $D_{expanded}$ and the computeWeight() function will compute the weight of each weakness, denoted as follows:

$$w_{\vec{d}} = \frac{\sum_{\vec{r} \in R} n_{\vec{d},\vec{r}}}{|R|}, \text{ such that } n_{\vec{d},\vec{r}} \begin{cases} 1, & \text{if } \exists(\vec{r}, (id_{\vec{d}}, v_{\vec{d}}, \vec{d}), sim_{\vec{r},\vec{d}}) \in D_{retrieved,\vec{r}} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

where $w_{\vec{d}}$ represents the weight of the weakness $\vec{d}$ relative to all SSI functional requirements, R represents a collection of SSI functional requirements, and $n_{\vec{d},\vec{r}}$ represents a flag indicating the presence of the weakness $\vec{d}$ in the retrieved

Algorithm 3 Ranking procedure for voting weaknesses that relate to multiple SSI functional requirements

```
Input:
    $R$ : a collection of SSI functional requirements,
    $D_{retrieved}$ : a collection of retrieved weaknesses,
    $D_{expanded}$ : a collection of expanded weakness vectors, and
    $y$ : a threshold indicating the acceptable weight of weakness.
Output:
    $D_{recommended}$ : a collection of recommended SSI-specific weaknesses.
1: procedure RANKWEIGHT($R, D_{retrieved}, D_{expanded}, y$)
2:     $D_{recommended} \leftarrow \emptyset$
3:     for $(id_{\vec{d}}, v_{\vec{d}}, \vec{d}) \in D_{expanded}$ do
4:         $w_{\vec{d}} \leftarrow$ COMPUTEWEIGHT($\vec{d}, D_{retrieved}, R$)
5:         if $w_{\vec{d}} > y$ then
6:             $D_{recommended} \leftarrow D_{recommended} \cup \{(id_{\vec{d}}, v_{\vec{d}}, \vec{d})\}$
7:     return $D_{recommended}$
```

result $D_{retrieved,\vec{r}}$ corresponding to the SSI functional requirement $\vec{r}$. $n_{\vec{d},\vec{r}} = 1$ if the weakness $\vec{d}$ is contained in the requirement $\vec{r}$'s retrieved result. This weighting scheme reflects the fact that if a weakness $\vec{d}$ appears on multiple retrieved lists, it will be assigned a high weight. The condition at line 5 will then compare the weight to the specified threshold y. If the weight is greater than the threshold y, the weakness triple $(id_{\vec{d}}, v_{\vec{d}}, \vec{d})$ will be collected into recommended results $D_{recommended}$.

Fig. 3.7 demonstrates the usage of Algorithms 2 and 3 by providing a running example of the SWIF weakness recommendation. Assume three SSI functional requirements $(r_1, r_2, r_3 \in R)$ are used as queries and represent the SSI management system's functionality. Algorithm 2 is executed to retrieve three collections of retrieved results by setting the threshold $x = 4$, and Algorithm 3 is executed to identify recommended weaknesses related to multiple SSI functional requirements by setting the threshold $y = 0.5$. Four weaknesses will have weights greater than 0.5, and SWIF will recommend them as SSI-specific weaknesses.

3.5 Weakness Mitigation Procedure

After recommending SSI-specific weaknesses based on language correlations, it is possible to determine whether the target SSI management system also

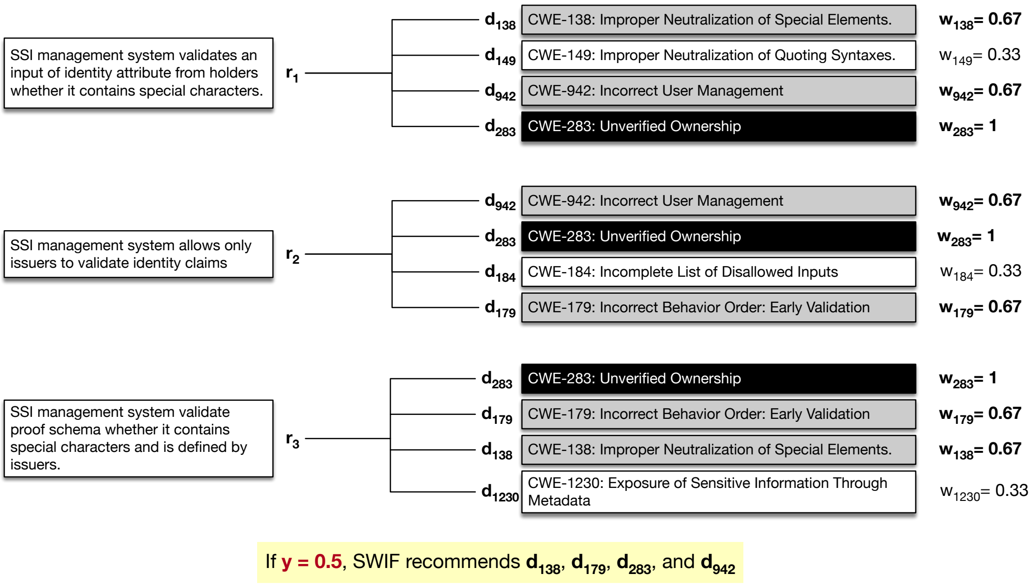

SSI management system validates an input of identity attribute from holders whether it contains special characters.
r1
d138
CWE-138: Improper Neutralization of Special Elements.
w138= 0.67
d149
CWE-149: Improper Neutralization of Quoting Syntaxes.
w149= 0.33
d942
CWE-942: Incorrect User Management
w942= 0.67
d283
CWE-283: Unverified Ownership
w283= 1
SSI management system allows only issuers to validate identity claims
r2
d942
CWE-942: Incorrect User Management
w942= 0.67
d283
CWE-283: Unverified Ownership
w283= 1
d184
CWE-184: Incomplete List of Disallowed Inputs
w184= 0.33
d179
CWE-179: Incorrect Behavior Order: Early Validation
w179= 0.67
SSI management system validate proof schema whether it contains special characters and is defined by issuers.
r3
d283
CWE-283: Unverified Ownership
w283= 1
d179
CWE-179: Incorrect Behavior Order: Early Validation
w179= 0.67
d138
CWE-138: Improper Neutralization of Special Elements.
w138= 0.67
d1230
CWE-1230: Exposure of Sensitive Information Through Metadata
w1230= 0.33
If y = 0.5, SWIF recommends d138, d179, d283, and d942

Figure 3.7: Running example of Algorithms 2 and 3 indicating how SSI-specific weaknesses are recommended.

contains these weaknesses. SWIF does not provide a method for automatically mitigating the recommended weaknesses, but it does provide criteria and procedures for mitigating them manually.

This dissertation realized that the SSI-specific weaknesses are impacted by three types of mistakes in the functionality of the target SSI management system: incorrect, incomplete, and missing. A weakness caused by *incorrect* functionality indicates that the target SSI management system does provide the corresponding functionality, but it has been incorrectly implemented. A weakness caused by *incomplete* functionality indicates that the target SSI management system implements a security function, but is missing some necessary steps. Lastly, a weakness caused by *missing* functionality indicates that the target SSI management system lacks the required security function.

Table 3.3: Criteria and procedures to mitigate SSI-specific weaknesses.

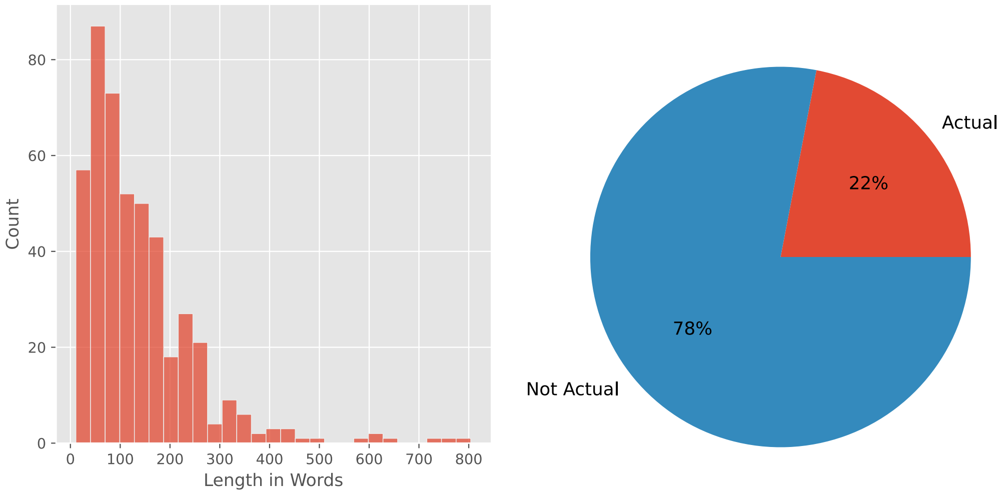
Criteria	Mitigation Procedure
Incorrect	1) Identify the location of the corresponding security function in the SSI management system. 2) Identify the correct security function suggested by the weakness. 3) Update the security function in alignment with the suggestion.
Incomplete	1) Identify the location of the corresponding security function in the SSI management system. 2) Compare the implemented and necessary steps to determine deficiencies. 3) Include the deficient necessary steps into the implemented steps as new SSI functional requirements.
Missing	1) Identify the missing security function suggested by the weakness. 2) Determine the functionality of the target SSI management system that is relevant to the missing security function. 3) Include the missing security function in the corresponding functionality as new SSI functional requirements.

As a criterion for determining the procedure to mitigate these weaknesses, the types of mistakes resulting from the identified weaknesses are determined. Table 3.3 outlines the criteria and procedures for mitigating each mistake type. In brief, the necessary steps suggested by the weakness must be compared to the existing SSI functional requirements and those that are lacking must be added. For instance, the CWE-306 weakness is recommended and determined to be *missing* because the blockchain's write mechanisms do not authenticate their users. This deficiency necessitates the introduction of a new SSI functional requirement.

The implementation of SWIF can ensure that SSI-specific weaknesses can be recommended and mitigated by system analysts, who conduct the design activity, when SSI functional requirements are gathered. However, it may be impossible to resolve all security issues by eliminating only security weaknesses. Consequently, additional analysis of the security and privacy of the SSI model or management system is needed for system analysts.

3.6 Implementation of SWIF

This section provides an example of the SWIF implementation as a command-line executable. This section also contains vital information that must be acknowledged when implementing SWIF, such as the dataset for security weaknesses and technical architecture.



(a) Weakness entry length in words

(b) Percentage of actual weaknesses

Figure 3.8: Analysis of the scoped dataset of common security weaknesses.

3.6.1 Analysis of Dataset for Security Weaknesses

As a main source for common security weaknesses, the CWE database includes almost a thousand of weakness entries. According to the purpose of SWIF, textual descriptions of common security weaknesses are compared to SSI functional requirements, which are parts of the design of the SSI management system. It is recognized that the CWE database provides metadata indicating the modes of introduction, such as requirements, architecture and design, and development phases.

For SSI-specific weakness identification, this dissertation assumes that only common security weaknesses with requirements, policy, architecture, and design phases in their modes of introduction are suitable. As a result, the dataset of common security weaknesses should be limited to them and this criterion results in 464 out of 927 weakness entries collected. Then, the scoped dataset's fundamental characteristics are analyzed in terms of word count. The majority of weakness entries in the dataset are approximately 50 to 100 words long, as depicted in Fig. 3.8a.

In addition to the fundamental characteristic, it can be recognized that the weakness identification requires a ground truth to evaluate if the identified SSI-specific weaknesses are accurate compared to subjective judgment. Therefore, three system analysts are invited to determine which common security weaknesses have language correlations with the SSI functional requirements provided. Fig. 3.8b demonstrates that, based on the security

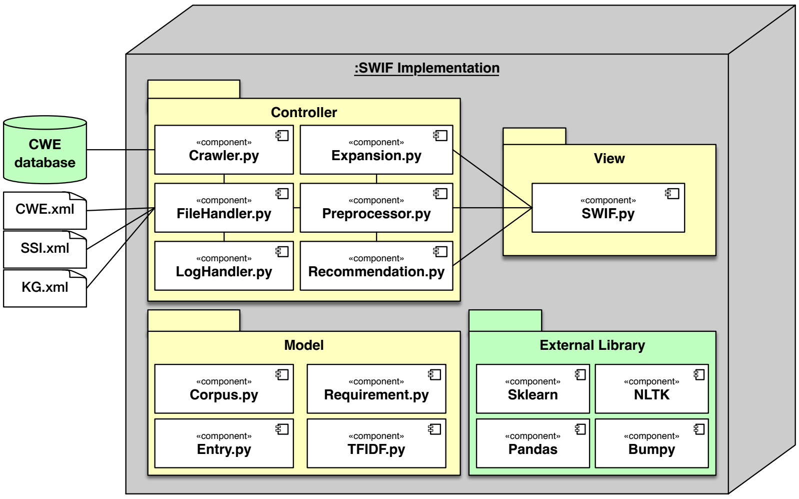


Figure 3.9: Architectural design of the SWIF implementation as a command-line Python program.

analysts' examination, 102 out of 464 scoped security weaknesses have language correlations to the provided set of SSI functional requirements. In conclusion, the scoped dataset will be the main source for common security weaknesses utilized in the implementation and evaluation of SWIF and the ground truth[3] is prepared for the evaluation.

3.6.2 Technical Architecture of SWIF Implementation

With the support of the dataset, SWIF has been implemented as a command-line program in Python to prove the execution of each module. The utilized workbench is a computer with a 3.4GHz quad-core processor, 32GB of 2400MHz DDR4 memory, and 4GD of graphic memory. The architectural design of the SWIF implementation is illustrated as a deployment diagram in Fig. 3.9, containing used files, libraries, and technologies.

The SWIF implementation uses the Model-View-Controller (MVC) design template that defines three different packages: model, view, and controller packages. First, a set of model classes that represent SWIF data objects is developed, including weakness vector corpus (Corpus.py), SSI functional requirements (Requirement.py), weakness entry (Entry.py), and TF-IDF vector (TFIDF.py) classes. Then, a package of controller classes was created for processing data objects in various ways. For instance, the "Crawler.py" and "FileHandler.py" classes are created to retrieve common

[3] Ground truth of actual SSI-specific weaknesses: https://doi.org/10.5281/zenodo.7423741.

security weaknesses from the CWE database and manipulate XML data files, respectively. To implement the text preprocessing module, the "Processor.py" class is developed for performing all suggested text preprocessing tasks. In addition, the knowledge expansion module is developed as a separate class named "Expansion.py" that collaborates with the text preprocessing module to use Algorithm 1 to expand cross-domain knowledge between the SSI and CWE domains. Following Algorithms 2 and 3, a controller class named "Recommendation.py" is developed to implement the recommendation module. Furthermore, a view package contains the "SWIF.py" file is provided as an interface for user and controller class communication.

The architectural design also specifies which external libraries are used to support the implementation, such as the SKLearns library for vectorization, the NLTK library for text preprocessing support, and the Pandas and Numpy libraries for modeling data objects. Based on the above design and implementation, an executable Python program that follows SWIF and can identify SSI-specific weaknesses is developed successfully.

3.6.3 Executable Program of SWIF Implementation

This section will describe the installation and operation manual for the executable Python program of the SWIF implementation. The stable version of the SWIF implementation is published on Github[4] so that researchers and system analysts can adopt it to identify SSI-specific weaknesses in their desired SSI management system.

Installation of Python Packages

Before executing the weakness identification with the SWIF implementation, a Python environment must be prepared. The installing environment must first download and install the Python compiler version 3.9 from the official website. Typically, packages and dependencies are required for the Python compiler to comprehend source code. As shown in Fig. 3.10, the `requirements.txt` file is generated to compile required packages and dependencies.

The requirements represent the packages and dependencies that correspond to the external libraries that were utilized during architectural design. In addition, additional data for some corpora, such as the WordNet corpus for lemmatization, is required to be downloaded for text preprocessing tasks.

[4] Open Source Repository for the SWIF implementation: https://github.com/chnpat/SWIF-Implementation.

```
beautifulsoup4==4.11.1
en-core-web-sm @ https://github.com/explosion/spacy-models/releases/
        download/en_core_web_sm-3.4.0/en_core_web_sm-3.4.0-py3-none-any.whl
matplotlib==3.5.2
nltk==3.3
numpy==1.23.0
pandas==1.4.3
Pillow==9.2.0
regex==2022.6.2
requests==2.28.0
scikit-learn==1.1.1
scipy==1.8.1
seaborn==0.11.2
sklearn==0.0
spacy==3.4.1
spacy-wordnet==0.0.5
stopwords==1.0.0
```

Figure 3.10: Requirements of packages and dependencies for the SWIF implementation.

This installation could take some time depending on the internet speed at the installing workbench.

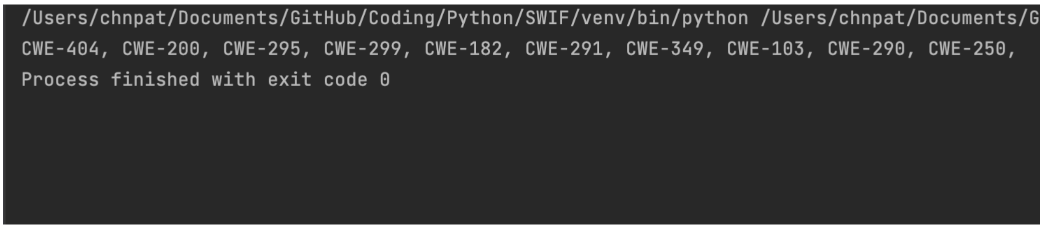
Operation Procedure of the SWIF Implementation

After the Python environment has been successfully prepared, the `SWIF.py` file will be the primary executable file. The `SWIF.py` file requires the input of an XML file containing the to-be-analyzed SSI functional requirements. The XML file must be named "queries.xml" and placed in the "Datasource" folder. Two levels are required for the SWIF implementation: queries and functions. The "queries" tags outline the query list of SSI functional requirements, which are denoted by the "function" tags. Fig. 3.11 illustrates an example of an XML input file. It is anticipated that sufficient SSI functional requirements will be provided for the SWIF implementation in order to define the target SSI management system. The number of input SSI functional requirements has no bearing on the SWIF implementation, but it will affect the execution time.

The SWIF implementation will operate automatically and report only a list of recommended SSI-specific weaknesses. In Fig. 3.12, an example of an output message recommending ten SSI-specific weaknesses is displayed. For instance, CWE-404 has the highest language correlations to the given list of SSI functional requirements (based on similarity scores) and has voted to be correlated with multiple SSI functional requirements. This output message can be used by security analysts as a resource for mitigating SSI-specific weaknesses by updating the SSI functional requirements and design.

```
<?xml version="1.0" ?>
<queries>
        <function>
                The identity wallet shall initiate a public key and
                a private key in the first use.
        </function>
        ...
        <function>
                The blockchain service shall create a data block
                containing the DID document.
        </function>
        ...
</queries>
```

Figure 3.11: Example of an XML file listing the input SSI functional requirements for the SWIF implementation.



```
/Users/chnpat/Documents/GitHub/Coding/Python/SWIF/venv/bin/python /Users/chnpat/Documents/G
CWE-404, CWE-200, CWE-295, CWE-299, CWE-182, CWE-291, CWE-349, CWE-103, CWE-290, CWE-250,
Process finished with exit code 0
```

Figure 3.12: Example of an output message indicating the identifiers of common weakness entries that are recommended to be SSI-specific weaknesses.

```
<?xml version="1.0"?>
<cdkg>
        <fact generic="application" relation="correlate_with">
                identity wallet
        </fact>
        ...
        <fact generic="software" relation="synonym_of">
                software program
        </fact>
        ...
</cdkg>
```

Figure 3.13: Example of an XML file listing domain knowledge from the fact of the SSI-CWE cross-domain transfer knowledge graph.

In addition, the SWIF implementation required the SSI-CWE cross-domain transfer knowledge graph as an input. The SWIF implementation initially provides a predefined SSI-CWE cross-domain transfer knowledge based on our domain knowledge, but it is possible for users to update this domain knowledge. The knowledge graph is provided in the "Datasource" folder's "cdkg.xml" XML file. In Fig. 3.13, a part of the content of the knowledge graph XML file is provided. The "cdkg" tags are used to define the scope of the SSI-CWE cross-domain transfer knowledge graph, and each fact is defined by two attributes and one value within the "fact" tags. The generic and relation attributes specify the generic term in the CWE weakness domain and the relation type for the fact, respectively. The value of the fact tag will be the SSI domain's specific terms. The SWIF implementation provides the capability to automatically identify SSI-specific weaknesses. However, the implementation cannot accommodate the manual mitigation requirements.

Chapter 4

Improvement of the Compliance of SSI Properties to Laws, Regulations, and Standards

This chapter will propose a compliance SSI system property based on laws, regulations, and technical standards in an effort to improve security and privacy control coverage. A consolidated SSI system property set that eliminates redundancy from multiple proposals is determined. A systematic review approach is then used to derive a collection of security and privacy controls shared among a wide variety of laws, regulations, and technical standards in order to identify a source of universal security and privacy safeguards. Finally, a comparative analysis is conducted to identify consistency among consolidated SSI system properties and shared controls in order to use them to improve SSI system property definitions. The subsequent sections will describe in detail the overview of the approach used, the consolidation of SSI system properties, the derivation of shared controls, and the analysis and improvement procedure, respectively.

4.1 An Approach for Improving SSI System Properties

It is discovered that the current SSI guiding principles and system properties are compatible with security and privacy controls to some extent. The consent property [15] and the purpose limitation control from GDPR [13] are

Table 4.1: Comparison between a current SSI system property and a control from GDPR.

Current SSI System Property	Privacy Control from GDPR
Consent. Every single piece of identity data must be released to a third party only after the corresponding user has consented to do so [15].	*Purpose Limitation.* Article 5.1.(b) Personal data shall be collected for specified, explicit, and legitimate purposes and not further processed in a manner that is incompatible with those purposes [13].

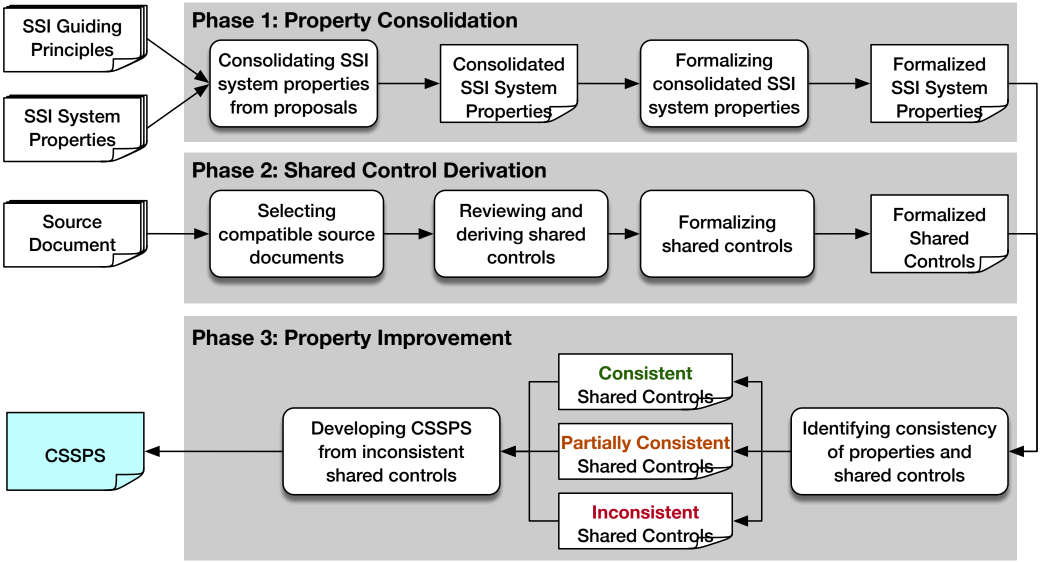

SSI Guiding Principles
SSI System Properties
Source Document
CSSPS
Phase 1: Property Consolidation
Consolidating SSI system properties from proposals
Consolidated SSI System Properties
Formalizing consolidated SSI system properties
Formalized SSI System Properties
Phase 2: Shared Control Derivation
Selecting compatible source documents
Reviewing and deriving shared controls
Formalizing shared controls
Formalized Shared Controls
Phase 3: Property Improvement
Developing CSSPS from inconsistent shared controls
Consistent Shared Controls
Partially Consistent Shared Controls
Inconsistent Shared Controls
Identifying consistency of properties and shared controls

Figure 4.1: Overview of the proposed approach for improving security and privacy through SSI system properties.

compared in Table 4.1. The consent property instructs the SSI management system to release identity data for any purpose following user consent. As suggested by the purpose limitation control, the property does not cover the protection of further processing that is incompatible with the initial purposes.

This section describes an approach for preparing a single list of SSI system properties, a list of shared controls, and how their consistency is compared. Fig. 4.1 provides an overview of the proposed approach, which consists of three major phases: *property consolidation*, *shared control derivation*, and *property improvement.*

The aim of the *property consolidation* phase is to create the most up-to-date and comprehensive view of the SSI system properties from the various proposals. On the basis of four proposals [4, 7, 14, 15], similar or duplicated

SSI principles and system properties with slightly different definitions are identified. It is redundant to make improvements based on each proposal individually. The proposed approach assumes that all four proposals for SSI guiding principles and system properties can be consolidated into a single list. Current SSI principles and system properties will be incorporated into the consolidated list of SSI system properties. Additionally, it can be recognized that the consolidated SSI system properties are written in plain text and may be difficult to compare systematically with shared controls. Consequently, a formal definition of the consolidated SSI system properties is given to formulate the comparable form. Section 4.2 will explain how the SSI system properties were consolidated and formalized.

The aim of the *shared control derivation* phase is to identify representative security and privacy controls shared in multiple source documents (i.e., laws, regulations, and standards). It is recognized that source documents provide security or privacy controls for use in compliance assessment checklists, and that some controls from different documents are similar. For example, the GDPR [13] describes the *accuracy* control similarly to the *accuracy and quality* control in ISO/IEC 29100:2011 [54]. To make security and privacy safeguards that are universally applicable to multiple source documents, this dissertation assumes that a list of shared controls that are located in source documents can be derived systematically. However, not all source documents are compatible with use as property improvement sources. As a result, a systematic review is conducted to sift through a large number of source documents and select those that are suitable for property improvement. For example, source documents that regulate organizational processes are incompatible and should not be included. In order to compare shared controls with the consolidated SSI system properties, a formal definition for shared controls is also given to formulate a comparable form. Section 4.3 will describe in detail how shared controls are derived and formalized.

The aim of the *property improvement* phase is to compare the current SSI system properties to shared controls in order to determine their consistency and to use the results to improve property definitions. Some SSI system properties, as shown in Table 4.1, are consistent with controls. They may not, however, be fully consistent. Consistency between an SSI system property and a shared control indicates that their respective written descriptions share the same semantics and can be subjectively evaluated for compliance. To structure analysis results, the following three types of shared controls are defined:

- A *fully consistent* shared control is one in which *all* endorsed tasks are consistent with any SSI system property constraint.

- A *partially consistent* shared control is one in which *some* endorsed tasks are consistent with any SSI system property constraint.
- An *inconsistent* shared control is one in which *none* of the endorsed tasks is consistent with any SSI property constraint.

To increase the likelihood of source document compliance, the partially consistent and inconsistent shared controls will be evaluated to revise SSI system properties with the missing endorsed tasks. In some instances, additional SSI system properties are required to be introduced if the missing endorsed tasks do not appear to be appropriate with the existing ones. The comparative analysis and improvement of SSI system properties are conducted through the use of automatic algorithms, which are described in detail in Section 4.4. As a result of the improvement, the compliance SSI system property set, or CSSPS, will contain SSI system properties that are more compatible with source documents.

4.2 Consolidated List of SSI Properties

This section will examine several existing proposals for SSI guiding principles and system properties, as well as how to eliminate duplication and consolidate SSI system properties in a formalized form.

4.2.1 Consolidating SSI System Properties

SSI principles and system properties are related in some ways, and one may derive from the other. This relation causes duplication in various proposals. For instance, the transparency principle can be found in one proposal [4] and the transparency property in another [15]. Table 4.2 provides a comparison between the definitions of the transparency principle and property. It can be seen that both principle and system property impose a constraint on the system by mandating that it be open source. Such duplications may necessitate additional effort if they are not consolidated prior to the improvement.

On the other hand, certain SSI guiding principles constraints may not be evaluable by the SSI management system and its functionality. For instance, the *existence* principle [4] states, "Any SSI is ultimately founded on the ineffable 'I' at the heart of identity." It is difficult for security analysts to determine this constraint based on the SSI management system's functionality. A key differentiator between SSI guiding principles and system properties is that system properties focus solely on system-centric characteristics or capabilities. For the purposes of security and privacy analysis

Table 4.2: Comparison between the transparency guiding principle and the transparency property.

SSI Guiding Principle	SSI System Property
Transparency. Systems and algorithms must be transparent. The systems used to administer and operate a network of identities must be open, both in how they function and in how they are managed and updated. *The algorithms should be free, open-source, well-known, and as independent as possible of any particular architecture; anyone should be able to examine how they work* [4].	**Transparency**. An SSI and its system must be transparent enough for every involved entity. Users should be well aware about all their partial identities and their corresponding interactions. The system and the corresponding algorithm must allow an easy retrieval of such interaction to ensure transparency. *Another way to achieve it is to ensure that the system is fully open source, allowing anyone to examine its internal mechanism and algorithms.* This will enable finding bugs within the system and ensure to be sustainable with a wider participation of members from the open source communities [15].

of the SSI management system, evaluable system properties should be obtained by excluding non-evaluable constraints. The following text will refer to all SSI principles without non-evaluable constraints as SSI system properties. To structure written data for comparison, *an SSI system property as a collection of constraints, achieving the property by satisfying all constraints.*

By excluding non-evaluable constraints, SSI principles can be transformed into SSI system properties and SSI system properties that are duplicated across different proposals can be consolidated. SSI principles constraints from three proposals [4, 7, 14] are evaluated to determine which constraints are system-centric evaluable based on the following conditions.

1. The constraint is subject to the SSI management system or its component system.
2. The constraint restricts SSI-specific functionalities or operations.
3. The constraint restricts the manipulation of data objects within the SSI management system.
4. The constraint can be interpreted as being directly evaluable on the SSI management system.

If at least one constraint in an SSI guiding principle meets more than two criteria above, the principle is considered to be an SSI system property that include the satisfying constraints. Table 4.3 presents a determination of two SSI guiding principles on whether their constraints are evaluable. Evidently, the interoperability principle [14] imposes a constraint on identity infrastructures and their functionality, satisfying that conditions 1 and 2. Consider this principle an SSI system property with an evaluable constraint. In contrast, the interoperability principle [4] imposes two evaluable constraints among its four constraints.

Table 4.3: Determination of two SSI guiding principles whose constraints are evaluable.

Principle	**Constraint**	**Evaluable?**
Interoperability [14]	Two different identity infrastructures should be capable of communicating with each other at any scale.	Yes
	This will enable enterprises and government organizations to communicate with each other irrespective of their employed identity infrastructures.	No
Interoperability [4]	Identities should be as widely usable as possible.	No
	Identities are of little value if they only work in limited niches.	No
	The goal of a 21st-century digital identity system is to make identity information widely available, crossing international boundaries to create global identities, without losing user control.	Yes
	Thanks to persistence and autonomy these widely available identities can then become continually available.	Yes

Then, SSI system properties from the converted SSI principles [4, 7, 14] and the system property proposal [15] are compared in order to identify and eliminate duplications. To determine if two or more SSI system properties can be consolidated, the following criteria have been established:

1. System properties should be consolidated if their title are identical.
2. System properties should be consolidated if they constrain on the same elements of the SSI management system.
3. System properties should be consolidated if they constrain the same system component with the same quality aspects.

If at least one of the aforementioned criteria is met, SSI system properties can be consolidated. Table 4.4 shows an example of a consolidated SSI system property, which consists of four constraints derived from various references of the property proposal. References are denoted by the identifiers A$x.y$, L$x.y$, N$x.y$, and F$x.y$, where A denotes the reference to Allen [4], L denotes the reference to Lopez [7], N denotes the reference to Naik & Jenkins [14], and F denotes the reference to Ferdous et al. [15], and x, y denotes the numbering scheme used to refer to the yth constraint of the xth system property. For instance, the first constraint of the *existence* property is derived from three references to the property proposal. Although the constraints are expressed in various different ways, the consolidation must collect all pertinent information.

Resultantly, 20 SSI system properties that represent the current property proposals are consolidated. Table 4.5 provides a summary of the titles and the complete definition of the consolidated SSI system properties is provided

Table 4.4: Example of consolidated SSI system property.

Title	Constraint	Reference
Existence	An SSI management system must enable users to represent their independent existence and characteristics to assert their selective information in the digital domain.	A1.1, F1.1, N2.1
	An SSI management system must make the independent existence and identities public and accessible.	A1.2, N2.3
	An SSI management system must link identities to users with their own unique identifiers	L1.1, N2.2, N5.3
	An SSI management system should not provide any mechanism to correlate confidential and biometric data with identities and credentials.	N11.2

Table 4.5: List of twenty consolidated SSI system properties.

Title of the Consolidated SSI System Property		
P_1 : Existence	P_8 : Consent	P_{15} : Cost Free
P_2 : Sovereignty	P_9 : Data Minimization	P_{16} : Decentralized
P_3 : Access	P_{10} : Protection	P_{17} : Verifiability
P_4 : Transparency	P_{11} : Recovery	P_{18} : Scalability
P_5 : Persistence	P_{12} : Single Source	P_{19} : Accessibility
P_6 : Portability	P_{13} : Availability	P_{20} : Sustainability
P_7 : Interoperability	P_{14} : Standard	

externally with the justification made[1].

4.2.2 Formalizing Consolidated SSI System Properties

The consolidated system properties obtained in the preceding section are described in plain text, preventing systematic comparisons. In other words, their comparability to other artifacts (i.e., shared controls) can only be justified subjectively. To address this difficulty, this section proposes a formal definition for an SSI system property in order to structure a comparable form for facilitating further analysis.

An analysis of SSI system property constraints is conducted and it can be determined that the constraints possess an implicit structure that is extractable. Typically, a constraint is a concise sentence controlling one or more functions of the SSI management system's component(s), such as the manipulation of a particular SSI-specific data object(s). For example, the first constraint of the existence property in Table 4.4 specifies that the SSI management system must permit users to represent their existence and char-

[1] Consolidated SSI system property definitions and analysis results: https://doi.org/10.5281/zenodo.7423887.

acteristics in their identity data. On the basis of this implicit structure, an SSI system property can be formally defined as follows.

Definition 4.2.1. *An SSI system property* is a collection of constraints denoted by a set $P \subseteq S \times A \times O = \{c : c = (S_c, A_c, O_c), S_c \subseteq S, A_c \subseteq A, O_c \subseteq O\}$, in which:

- c denotes a constraint triple (S_c, A_c, O_c),
- S denotes a set of terms or phrases indicating the SSI management system's components,
- $S_c \subseteq S$ denotes a subset of terms or phrases indicating the SSI management system's components that are related to the constraint c,
- A denotes a set of terms or phrases indicating actions or functions of the SSI management system,
- $A_c \subseteq A$ denotes a subset of terms or phrases indicating actions or functions that are restricted by the constraint c,
- O denotes a set of terms or phrases indicating data objects that are unique in the SSI management system, and
- $O_c \subseteq O$ denotes a subset of terms or phrases indicating data objects that are controlled by the constraint c.

A constraint triple (S_c, A_c, O_c) is implicitly structured in terms of how system components, actions, and data object manipulations are controlled, per the definition. However, the omission of information and the presence of excessively detailed information in constraints posed two additional challenges when structuring constraints in SSI system properties. For instance, the decentralized property has a constraint stating, "An SSI management system should register and manage through a decentralized infrastructure mostly run publicly." The constraint does not specify explicitly which data objects must be registered and managed by the SSI management system. This omission requires interpretation to determine what is missing. For instance, the *decentralized* property omits the registration and management of identity data. To satisfy the formalized property, the phrase "(identity)" should be added to O_c, indicating our interpretation with parentheses. In addition, the phrase "mostly run publicly" is excessively detailed and is unnecessary for evaluating compliance with this constraint. To simplify the formalized SSI system properties, the excessively detailed phrases can be omitted justifably. Table 4.6 displays a formalized example of decentralized property.

The formalization could be applied to the 20 consolidated SSI system properties. As shown in Table 4.5, each SSI system property is denoted by the abbreviation P_x, which represents the xth consolidated SSI system

Table 4.6: Example of the formalized SSI system property.

Plaintext SSI System Property	*Decentralized.* [constraint 1] An SSI management system should not register and manage identities centrally by *any proprietary organization.* [constraint 2] An SSI system should register and manage through a decentralized infrastructure *mostly run publicly.*
Formalized SSI System Property	P_{CP16} = { ({"SSI management system"}, {"not register", "not manage"}, {"identity"}), ({"SSI management system"}, {"register", "manage through a decentralized infrastructure"}, {(*identity*)}) }

The complete formalization of the consolidated SSI system properties are also provided in https://doi.org/10.5281/zenodo.7423887.

property. At this stage, a single list of consolidated SSI system properties that could be used for property improvement is compiled.

4.3 Shared Controls of Information Security and Privacy

The controls from source documents are an additional important input for our security and privacy improvements. Across business domains, hundreds of laws, regulations, and standards can serve as security and privacy source documents. However, not every source document can be used to improve SSI system properties. Source documents that are suitable for property improvement must be chosen first. In addition, it can be realized that if the improvement corresponds with each source document individually, this would limit the universality of SSI system properties. It would be more advantageous to make SSI system properties more universally usable if shared controls that are frequently stated in multiple source documents can be derived and use them to improve the properties.

This section will conduct a systematic review for selecting applicable source documents from multiple online resources and analyzing them to derive which security and privacy controls are shared among them. To conduct the review systematically, a research question that is expected to be answered from the review is formulated, as follows.

RQ. What are shared controls that are frequently stated in multiple laws, regulations, and standards for information security and privacy?

The above research question is the major goal of the systematic review, but it is difficult to find the answer evidently. To approach to the finding of that question, three additional guiding questions are established, as follows.

Q1. Where can applicable laws, regulations, and standards be gathered?

Q2. Which source documents are appropriate for SSI system property improvement of security and privacy?

Q3. How can we determine which controls are shared among selected source documents?

Since laws, regulations, and standards can be collected from a variety of hard and soft copy resources, the first guiding question is to identify online resources that provide sufficient access to knowledge about source documents. The second guiding question directs how are applicable source documents evaluated for our property improvement. The third guiding question seeks to specify precisely which shared controls will be utilized in this dissertation. The systematic review will search for answers to the main research question and additional guiding questions.

4.3.1 Selecting Compatible Source Documents

To answer the first question (*Q1*), the search strategy is determined to collect as many source documents as possible that explain information security and privacy controls. Numerous laws, regulations, and standards are currently available online and can be accessed for free or for a fee. This research asssumed that accessing source documents from online resources is sufficient for gathering data for our property improvement. This section will be divided into two subsections: gathering source documents and filtering them.

Gathering Source Documents from Online Resources

This research selected *websites* and *survey papers* as two reliable online resources that provide access to source documents. Numerous websites currently provide information about laws, regulations, and standards that can be source documents about information security and privacy. Wikipedia is a website that allows communities to verify and publish various facts. Although the information on Wikipedia is continually updated, it provides this research with numerous lists of useful source documents. ISO.org [55], the official website of the International Organization for Standardization (ISO), is another website that lists hundreds of published standards. ISO is a reputable international organization for acquiring standards. On the other hand, survey papers are reliable academic records that examine source documents pertaining to information security and privacy, which are acquired and analyzed critically. Multiple online archives, including IEEE Xplore, Elsevier ScienceDirect, Google Scholar, and SpringerLink, is used for searching survey papers that suit to the needs.

Table 4.7: Summary of online resources, keywords used, search results, and the number of source document titles found.

Online Resource	**Keyword Used**	**Search Result**	**#Source**
Research Archives	Information Security Standards	**A:** [56] **B:** [57]	10 44
	Information Privacy Standard	-	-
	Information AND {Security OR Privacy} AND {Laws OR Regulations}	-	-
Wikipedia	Information Security Standards	**C:** [58]	20
	Information Privacy Standards	-	-
	Information AND {Security OR Privacy} AND {Laws OR Regulations}	**D:** [59] **F:** [60] **E:** [61] **G:** [62]	8 58 12 12
ISO.org	{Security OR Privacy}	**H:** [55]	122

Using the selected online resources, different keywords are submitted to the search engine on such websites and archives to find lists of source documents and survey papers. The goal of the search is to locate as many titles for information security and privacy source documents. the keywords used and the search results from each online resources are summarized in Table 4.7.

Online archives for survey papers are searched using three keywords: "Information Security Standards," "Information Privacy Standards," and "Information AND {Security OR Privacy} AND {Laws OR Regulations}." Only two survey papers [56, 57] exist that evaluate the current state of knowledge regarding information security standards, and they provide 10 and 44 titles of source documents, respectively. Other survey papers that were came across lacked a list of source documents and only cited one or a few. Due to the small number of source document titles provided by the identified survey papers, a manual search of information security and privacy source documents [63] conducted by us is included. The manual effort should supplement scholarly research by offering an alternative viewpoint on source documents.

On the other hand, a search for the same keywords on Wikipedia provides five web pages containing information about information technology security standards, cybersecurity regulations, information privacy laws, privacy laws, and privacy policies, with respective lists of 20, 8, 58, 12, and 12 source documents. Finally, the search term "{Security OR Privacy}" is used on the ISO.org website to retrieve information security and privacy-related ISO standards. The website returns hundreds of standards, but only 122 source documents whose titles contained "security" or "privacy" are collected.

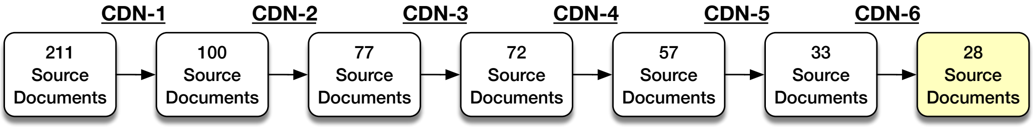


Figure 4.2: Summary of selection procedure and results, showing the number of source documents that meet each criterion.

A discriminating search for all source document titles is performed to obtain only those that were publicly accessible and sufficient for analysis. *211 distinctive source documents* are gathered, including *134 ISO standards*, *65 laws or regulations*, and *12 frameworks* for evaluating security and privacy. This research will be limited to the 211 gathered source documents. There may be additional source documents, but this dissertation assumes that the gathered source documents are sufficient for deriving shared controls.

Filtering Applicable Source Documents

The source documents discuss information security and privacy, but they may be incompatible with improving SSI system properties or applying to the SSI management system because they are collected indiscriminately. Several ISO standards, for instance, stipulate requirements for organizational data manipulation processes in order to safeguard information security and privacy. These source documents are unsuitable for our property improvements and should be excluded from further processing.

This section filters the gathered source documents by selecting only suitable source documents based on predefined criteria. Each source document is reviewed systematically according to six aspects: software-oriented, independence, availability, universality, evaluable, and suitability. These aspects are used to define the selection criteria as shown in Table 4.8.

The six preceding criteria outline the kinds of source documents that will be excluded. each of the 211 gathered source documents is compared to the selection criteria. This dissertation infers that source documents meet such criteria if we can provide evidence or an adequate justification to support them. Source documents that satisfy all of the criteria are intended to be chosen. Fig. 4.2 depicts the process as well as the outcomes for each criterion. The figure shows that 28 source documents are obtained after the analysis.

This section concludes by listing the 28 source document titles that satisfy all of the selection criteria in Table 4.9. $\texttt{SD}_i$ is the abbreviation for the ith source document, where i is the numbering scheme for indexing the 211 gathered source documents. The table also displays the number

Table 4.8: Selection criteria for filtering applicable source documents

Aspect	**Criteria**	**Justification and Example**
Software-Oriented	CDN-1: A source document must be established for regulating or evaluating software systems.	This criterion excludes source documents addressed to non-software-related targets, such as COBIT [64] and other organizational process standards.
Independence	CDN-2: A source document must be established for regulating or evaluating technological independent targets.	This criterion excludes source documents that are related to specific technologies, such as the payment card industry data security standard (PCI-DSS) [65].
Availability	CDN-3: A source document must be already published and available for use.	This criterion excludes source documents currently under development or are not operational. For instance, the EU Directive 95/46/EC [66] is no longer in force, despite being referenced in websites or survey papers.
Universality	CDN-4: A source document must not be domain-specific.	This criterion excludes domain-specific source documents. If source documents are domain-specific, our SSI system properties that are improved will be limited to a specific domain. For example, HIPAA [67] establishes requirements for the healthcare application domain.
Evaluable	CDN-5: A source document must provide at least one control evaluable by the functionality of the target software.	This criterion excludes source documents that did not provide an evaluable controls, which is required for comparing to SSI system properties. For example, ISO/IEC 29192-2:2019 [68] defines just algorithms and designs for lightweight cryptography.
Suitability	CDN-6: A source document must provide at least one control eligible to evaluate the SSI management system.	This criterion excludes source documents that provide incompatible controls with SSI management systems. For example, ISO/IEC 18032:2005 [69] provides requirements for prime number generators that are not usable.

Table 4.9: List of source document titles that met all selection criteria together with the number of controls and their applicable scope.

Source Document Title	#Control	Scope
$\mathtt{SD}_{005}$ = ISO/IEC 9796-2:2010 [71]	3	International
$\mathtt{SD}_{007}$ = ISO/IEC 9798-2:2008 [72]	6	International
$\mathtt{SD}_{009}$ = ISO/IEC 10118-1:2018 [73]	1	International
$\mathtt{SD}_{017}$ = ISO/IEC 11770-1:2010 [74]	5	International
$\mathtt{SD}_{022}$ = ISO/IEC 13888-1:2020 [75]	2	International
$\mathtt{SD}_{036}$ = ISO/IEC 18014-1:2008 [76]	2	International
$\mathtt{SD}_{037}$ = ISO/IEC 18031:2005 [77]	5	International
$\mathtt{SD}_{049}$ = ISO/IEC 19772:2020 [78]	2	International
$\mathtt{SD}_{063}$ = ISO/IEC 27001:2013 [52]	16	International
$\mathtt{SD}_{064}$ = ISO/IEC 27002:2013 [79]	16	International
$\mathtt{SD}_{095}$ = ISO/IEC 27701:2019 [80]	20	International
$\mathtt{SD}_{104}$ = ISO/IEC 29100:2011 [54]	11	International
$\mathtt{SD}_{123}$ = IEC 62443-3-3:2013 [81]	7	International
$\mathtt{SD}_{126}$ = NERC Cyber Security Standards [82]	6	International
$\mathtt{SD}_{136}$ = NIST Special Publication 800-12 Revision 1 [83]	20	International
$\mathtt{SD}_{141}$ = Cyber Essentials [84]	5	International
$\mathtt{SD}_{145}$ = OECD Privacy Framework [85]	8	International
$\mathtt{SD}_{146}$ = UN Personal Data Protection and Privacy Principles [86]	10	International
$\mathtt{SD}_{152}$ = General Data Protection Regulation [13]	7	International
$\mathtt{SD}_{160}$ = The Information Technology Act [87]	17	India
$\mathtt{SD}_{165}$ = The Personal Data Protection Code of Practice [88]	19	Malaysia
$\mathtt{SD}_{173}$ = Data Privacy Act [89]	12	Philippines
$\mathtt{SD}_{175}$ = Federal Law on Personal Data [90]	10	Russia
$\mathtt{SD}_{187}$ = Personal Data Act [70]	9	Sweden
$\mathtt{SD}_{189}$ = Personal Data Protection Act [91]	3	Taiwan
$\mathtt{SD}_{195}$ = Children's Online Privacy Protection Act [92]	3	United States
$\mathtt{SD}_{207}$ = Personal Information Protection Law [93]	13	China
$\mathtt{SD}_{211}$ = OWASP Application Security Verification Standard [94]	14	International

Note: full analysis results showing the checklist of selection criteria against each source document are provided in https://doi.org/10.5281/zenodo.6951404.

of controls present in the source document (#Control) and the source document's applicable scope (Scope). In Sweden, for instance, the Personal Data Act (PDA) [70] imposes nine controls applicable to software systems. The 28 source documents will be used to derive shared controls for confidently responding to the second guiding question (Q2).

4.3.2 Reviewing and Deriving Shared Controls

At this stage, a collection of source documents for improving the SSI system's properties is obtained sufficiently. Multiple controls are defined in these source documents for governing security and privacy in information processing. As it has been assumed that certain controls are shared among source documents, this section will detail the review of all controls in the 28 source documents and explain how shared controls are derived.

Table 4.10: Comparison of alternative terms and representative terms

Representative Term	Alternative Term in Source Document
system	application, data user, online service, PII collector, PII processor, system, service, website
personal data	data, information, message, personal data, personal information, sensitive data
entity	data controller, data owner, data subject, individual, protection authority, user

A control is a collection of endorsed tasks that should be evaluated to conclude the control's achievement. This dissertation make an assumption that two or more controls from different source documents as a shared control if they are defined in a similar manner. In order to merge the controls systematically, three conditions to determine whether two controls are similar are defined, as follows.

1. If two or more controls are named identically or has similar titles, we consider these controls similar.
2. If two or more controls are named differently or has different titles but they control the same aspect or have the same purpose, we consider these controls similar.
3. If at lease one pair of endorsed tasks in two or more controls is justifiable to be similar or identical, we consider these controls similar.

With the above-described conditions, similar controls are grouped to create a taxonomy based on their titles. In such similar controls, a number of endorsed tasks are comparable or identical, and can be combined in a way that incorporates all necessary information. However, it is discovered that different terms may be used in various source documents, making the merge prone to error. This issue is resolved by defining representative terms that can be substituted for some alternatives. In one source document, for instance, the term "PII processor" is used, while in another, the term "application" is used with the same meaning. Therefore, this dissertation predefines the term "system" as a representative term for these two terms. A summary of the predefined representative terms are provided in Table 4.10, which will be used to replace alternative terms with representative terms when merging similar controls.

An example of how the "data accuracy and quality" shared control are merged is illustrated in Table 4.11. Based on the first criterion, three source documents ($\texttt{SD}_{104}$, $\texttt{SD}_{153}$, and $\texttt{SD}_{207}$) contain comparable titles regarding data

Table 4.11: Example of the "data accuracy and quality" shared control derived from five source documents.

Control from Source Document	Shared Control
SD$_{104}$: *Accuracy and Quality* - The PII processed in the system is **accurate**, complete, **up-to-date**, adequate and relevant for the purpose of use.	**Data Accuracy and Quality**: (1) Personal data must be relevant to the purposes for which they are to be used, and, to the extent necessary for those purposes, should be accurate, complete and kept up-to-date. (2) If entities discover that their personal data is inaccurate or incomplete, they shall have the right to request the systems to correct, supplement, destroy, rectify, or restrict their further processing, without delay.
SD$_{152}$: *Accuracy* - Personal data shall be **accurate** and, where necessary, **kept up to date**; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay.	
SD$_{173}$: Personal information must, be **accurate**, relevant and, where necessary for purposes for which it is to be used the processing of personal information, **kept up to date**; inaccurate or incomplete data must be rectified, supplemented, destroyed or their further processing restricted.	
SD$_{187}$: The personal data that is processed is **correct** and, if it is necessary, **up-to-date**.	
SD$_{207}$: *Accuracy* - When processing personal information, the quality of the personal information shall be guaranteed in order to avoid any adverse impact on individuals' rights and interests caused by inaccurate or incomplete personal information.	

We provide full derivation results showing how we obtain shared controls in https://doi.org/10.5281/zenodo.6951404.

accuracy and quality. In addition, two controls in the remaining two source documents (SD$_{173}$ and SD$_{187}$) provide their endorsed tasks regarding the need for accurate, complete, and up-to-date data. These five controls are considered similar and can be combined into a shared control titled "data accuracy and quality," which defines its endorsed tasks based on the information contained in the five similar controls. It can also be seen that representative terms are used in the definition of the endorsed tasks for the shared control.

As a result of this section, 17 shared security controls and 14 shared privacy controls are derived from the 28 source documents. In Table 4.12, the titles of the derived shared controls are separated into two categories: security and privacy. The identifiers S.x and P.y, respectively, represent the xth security shared control and the yth privacy shared control.

4.3.3 Formalizing Shared Controls

Similar to the consolidated SSI system properties, shared controls are described in plaintext and must be interpreted subjectively. This section also gives a formal definition of derived shared controls in order to structure them in a comparable form.

Table 4.12: List of shared control titles categorized into security and privacy shared controls.

Security Shared Control		**Privacy Shared Control**	
S.1. Data Integrity S.2. Data Confidentiality S.3. Data Availability S.4. Authentication S.5. Authorization S.6. Accountability S.7. Non-Repudiation S.8. Validation and Sanitization S.9. Error Handling	S.10. Key Protection S.11. Malware Protection S.12. Communication Security S.13. Physical and Environmental Security S.14. Password Security S.15. Configuration Security S.16. Session Security S.17. Data Classification	P.1. Access Control P.2. Consent P.3. Fairness and Lawfulness P.4. Purpose Specification and Limitation P.5. Collection Limitation P.6. Use, Retention, and Disclosure Limitation	P.7. Data Accuracy and Quality P.8. Data Minimization P.9. Data Erasure and Rectification P.10. Data Recovery P.11. Notification P.12. Transparency P.13. Individual Participation P.14. Portability

Endorsed tasks in shared controls are analyzed to determine the implicit structure they possess. Typically, an endorsed task in a shared control is a concise statement controlling one or more information processing functions or operations of the target. For instance, the second endorsed task in the data accuracy and quality shared control regulates the system to permit entities to update their information's accuracy. This dissertation hypothesized that endorsed tasks correspond to SSI system property constraints. Using this hypothesis, a structure of shared controls in the same formal manner as the SSI system properties can be defined as follows.

Definition 4.3.1. *A shared control* is a collection of endorsed tasks denoted by a set $C \subseteq T \times F \times I = \{e : e = (T_e, F_e, I_e), T_e \subseteq T, F_e \subseteq F, I_e \subseteq I\}$, in which:

- e denotes an endorsed task triple (T_e, F_e, I_e),
- T denotes a set of terms or phrases indicating all the targets of shared controls,
- $T_e \subseteq T$ denotes a subset of terms or phrases indicating the targets that are controlled by the endorsed task e,
- F denotes a set of terms or phrases indicating all functions or operations within the target,
- $F_e \subseteq F$ denotes a subset of terms or phrases indicating functions or operations that are controlled by the endorsed task e,
- I denotes a set of terms or phrases indicating all information-related objects within the target, and
- $I_e \subseteq I$ denotes a subset of terms or phrases indicating information-related objects that are controlled by the endorsed task e.

Table 4.13: Example of formalized “data integrity” shared control.

Plaintext Shared Control	*S.1. Data Integrity* [task 1]: Personal data message must be signed and authenticated using digital signatures. [task 2]: Keys for message signing and verification must be shared in a secure way. [task 3]: Digital time-stamping should be used in a non-forgeable way. [task 4]: Systems must validate the integrity of the personal data. [task 5]: If the message authentication uses a hash function, it must be collision resistant.
Formalized Shared Control	C_{S1} ={ ({“*(system)*”}, {“sign”, “authenticate”}, {“personal data”}), ({“*(system)*”}, {“share in a secure way”}, {“key”}), ({“*(system)*”}, {“use”, “time-stamping in a non-forgeable way”}, {}), ({“system”}, {“validate integrity”}, {“personal data”}), ({“*(system)*”}, {“use”, “collision-resistant hash function”}, {“*(personal data)*”}) }

Similar to SSI system properties, an endorsed task triple (T_e, F_e, I_e) represents an implicit structure pertaining to the control of targets and their functions on information-related objects. Nevertheless, the formalization of shared controls faces the same problems of omission and excessive detail. For instance, the *data integrity* shared control includes an endorsed task that states, “Keys for message signing and verification must be shared in a secure manner.” This endorsed task does not specify explicitly which targets are being controlled, so the target system must be assumed. The formalization must include the phrase “(system)” ”utilizing parentheses to address this assumption. On the other hand, the phrase “for message signing and verification” is used to elaborate the term “keys,” but it is not required for evaluating this task of the data integrity shared control and may be deemed excessively detailed information. This phrase from the formalization should be excluded. A formalized example of the *data integrity* shared control is presented in Table 4.13.

All 17 security and 14 privacy shared controls are formalized in order to prepare knowledge for systematic comparison with SSI system properties. In addition, the shared controls could serve as an overview of multiple source documents and provide sufficient information for universally improving information security and privacy.

4.4 Compliance SSI System Property Set

Sections 4.2 and 4.3 perform an analysis of two knowledge sources for comparing and improving the information security and privacy of SSI system properties. This section uses the formalized knowledge sources to develop a Compliance SSI System Property Set, or CSSPS, that adheres to laws, regulations, and standards. This section will be divided into two sub-sections. First, the determination of consistency among SSI system properties and shared controls is defined, and then two systematic methods are used to identify consistency and develop CSSPS.

4.4.1 Formalizing Consistency Operator

As demonstrated in Table 4.1, SSI system properties are expressed in the same manner as controls in source documents. Using this semantic, the definition of how SSI system properties are consistent with shared controls can be given based on the following three conditions.

1. If at least one target controlled by the endorsed task is comparable to at least one subject in the constraint, the endorsed task is considered to be consistent with the constraint.
2. If at least one function controlled by the endorsed task is comparable to at least one action or operation in the constraint, the endorsed task is considered to be consistent with the constraint.
3. If at least one information-related object controlled by the endorsed task is comparable to at least one data object in the constraint, the endorsed task is considered to be consistent with the constraint.

The conditions help determine which pairs of an endorsed task and a constraint are consistent. To systematically and formally identify such consistency, a comparable operator can be formally defined based on the formal definitions of SSI system properties and shared controls as follows.

Definition 4.4.1. A comparable operator $\simeq$ denotes the consistency between two sets of terms or phrases.

The comparable operator signifies that two sets of terms or phrases are consistent. $X \simeq Y$, for instance, signifies that the set X is consistent with the set Y and that there exists $\exists x \in X$ and $\exists y \in Y$ such that x is justifiably comparable to y. Using the formal definitions of SSI system property and shared control (Definitions 4.2.1 and 4.3.1), their consistency can be formally defined as follows.

Definition 4.4.2. An SSI system property P and a shared control C are consistent if and only if $\exists c \in P$ and $\exists e \in C$ such that $S_c \simeq T_e$, $A_c \simeq F_e$, and $O_c \simeq I_e$.

According to the preceding definition, an SSI system property is consistent with a shared control if at least one of its constraints has a comparable relationship to at least one endorsed task from the shared control. All three elements of the constraint must be comparable, but it is negotiable if one element cannot be compared for a valid reason.

Further that, we realized that there can be three degrees of consistency that we can further utilize, including fully consistent, partially consistent, and inconsistent. First, a shared control is fully consistent with an SSI system properties if all endorsed tasks are comparable with any constraint. Then, a shared control is partially consistent if not all but some endorsed tasks are comparable. Lastly, a shared control is inconsistent if none of the endorsed tasks are comparable. Based on these aspects of the degree of consistency, we can formally define such degrees as follows.

Definition 4.4.3. A shared control C is consistent with an SSI system property P to the following degree:

- C is *fully consistent* with P if $S_c \simeq T_e, A_c \simeq F_e, O_c \simeq I_e, \forall e \in C$, and $\exists c \in P$,
- C is *partially consistent* with P if $S_c \simeq T_e, A_c \simeq F_e, O_c \simeq I_e, \exists e \in C$, and $\exists c \in P$,
- C is *inconsistent* with P if $S_c \not\simeq T_e, A_c \not\simeq F_e, O_c \not\simeq I_e, \forall e \in C, \exists c \in P$.

According to the preceding definitions, identifying consistency between SSI system properties and shared controls is sufficient. The following sections will present the automatic methods for identifying consistency and exploiting this information to develop CSSPS. The primary objective of identifying consistency is to determine how many shared controls are already consistent with the current SSI system properties. Fig. 4.3 depicts an overview of the used methodology.

As shown in the figure, the proposed method consists of two parts. First, the proposed method will classify shared controls as fully consistent, partially consistent, or inconsistent based on Definitions 4.4.2 and 4.4.3 for each pair of SSI system property and shared control. Then, the second part will improve the existing SSI system properties for each category of shared controls. In point of fact, the partially consistent shared controls will be utilized to enhance the existing properties by introducing new constraints for the missing endorsed tasks. If some missing endorsed tasks do not correspond to the existing SSI system properties, additional properties may be added. Inconsistent shared control, on the other hand, means that no SSI system property consists of endorsed tasks of the shared control. In this instance, additional SSI system properties will be introduced.

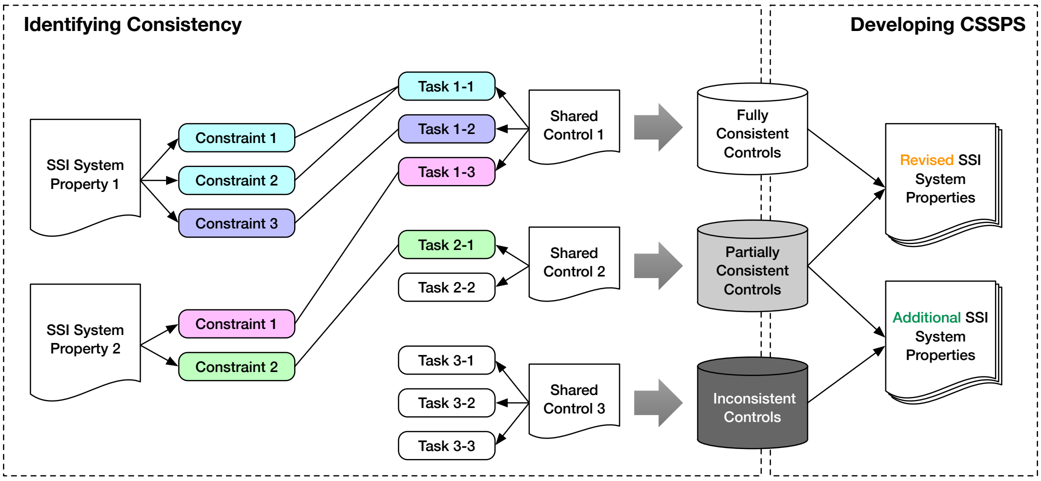


Figure 4.3: Overview of the proposed method for identifying consistency and developing CSSPS.

4.4.2 A Method to Identify Consistency Between SSI System Properties and Shared Controls

The first part of the proposed method to identify consistency is described in Algorithm 4, which defines the semi-automatic IDENTIFYCONSISTENCY() procedure. On the inputs of a shared control of analysis (C), a set of consolidated SSI system properties ($\mathcal{P}$), and a set of predefined domain knowledge (K), the procedure generates a quadruple $r = (C, type, R_{consist}, E_{missing})$ containing the corresponding control (C), the identified control type ($type$), a set of endorsed tasks with their consistent constraint ($R_{consist}$), and a set of missing endorsed tasks in the shared control ($E_{missing}$). On line 2 of the IDENTIFYCONSISTENCY() procedure, the resultant sets ($R_{consist}, E_{missing}$) and local variables ($flag, degree$) are initialized. The procedure then compares each endorsed task e_i within the specified shared control C to each constraint $c_{j,k}$ of each SSI system property $P_j \in \mathcal{P}$.

The condition at line 7 is intended to compare elements of the formalized structure of a shared control and an SSI functionality. The EVALUATE() function is designed to determine if two elements are comparable (i.e., $S_k \simeq T_i$, $O_k \simeq I_i$, and $A_k \simeq F_i$) based on predefined domain knowledge regarding term associations K. It can be acknowledged that terms or phrases describing both SSI system properties and shared controls are unique due to domain differences. As a result, the number of such terms or phrases is small, and the association between them can be manually predefined. As shown in Table 4.14, this dissertation therefore predefines term associations based on our

Algorithm 4 Procedure for identifying shared controls that are fully consistent, partially consistent, or inconsistent with SSI system properties.

```
Input: $C$ : A shared control that will be analyzed,
    $\mathcal{P}$ : A set of consolidated SSI system properties, and
    $K$ : Predefined domain knowledge indicating term associations.
Output: $r$ : A resultant quadruple indicating the corresponding control $C$,
    its category $T$, a set of pairs of consistent task and constraint $R_{consist}$,
    and a set of missing tasks $E_{missing}$.
 1: procedure identifyConsistency($C, \mathcal{P}, K$)
 2:     $R_{consist} \leftarrow \emptyset, E_{missing} \leftarrow \emptyset, degree \leftarrow 0, flag \leftarrow$ false
 3:     for $e_i = (T_i, F_i, I_i) \in C$ do
 4:         $flag \leftarrow$ false
 5:         for $P_j \in \mathcal{P}$ do
 6:             for $c_{j,k} = (S_k, A_k, O_k) \in P_j$ do
 7:                 if (evaluate($S_k, T_i, K$) or evaluate($O_k, I_i, K$)) and determineAssociation($A_k, F_i$) then
 8:                     $sign \leftarrow$ determineDirection($c_{j,k}, e_i$)
 9:                     $R_{consist} \leftarrow R_{consist} \cup \{c_{j,k}, e_i, sign\}$
10:                     $flag \leftarrow$ true
11:         if $flag =$ true then $degree \leftarrow degree + 1$
12:         else $E_{missing} \leftarrow E_{missing} \cup \{e_i\}$
13:     if $degree = |C|$ then $type \leftarrow$ "$FC$"
14:     else if $degree < |C|$ and $degree > 0$ then $type \leftarrow$ "$PC$"
15:     else $type \leftarrow$ "$IC$"
16:     return $r \leftarrow (C, type, R_{consist}, E_{missing})$
```

domain knowledge of both SSI management systems and source documents. For instance, the term "infrastructure" in a constraint corresponds to the term "system" in an endorsed task. The predefined domain knowledge K is a collection of term pairs that have an association to each other, such as $\{$("infrastructure", "system"), $\ldots\}$. This collection will be used with the evaluate() function. In contrast, the terms and phrases used to define functions or operations in constraints and endorsed tasks are diverse and difficult to predefine exhaustively. In this case, the determineAssociation() function is used to request a manual determination of the association between two input terms.

If the condition at line 7 is valid, the endorsed task (e_i) is considered to be consistent with the constraint ($c_{j,k}$). The determineDirection() function is designed to determine if an endorsed task differs from a constraint due to

Table 4.14: Predefined domain knowledge about term associations.

Term Association for Component		**Term Association for Data Object**	
Subject S	**Target** T	**Data Object** O	**Information-related Object** I
SSI management system	system	identity, attribute	personal data
infrastructure	system	identity claim	personal data
protocol, algorithm	system	SSI	personal data
user, holder, subject	entity, user	consent	personal data
issuer, verifier	system	input data, file	personal data
blockchain	system	public key, private key	key
		log information	log information

the fact that they may be consistent but provide different information. The function is defined as a partial map `determineDirection` : $(c_{j,k}, e_i) \rightarrow \{<, =, >\}$ where "$<$", "$=$", and "$>$" indicate that the constraint $c_{j,k}$ contains less, equal, or more information than the endorsed task e_i, respectively. The determineDirection() function returns the direction to the variable *sign*, which can be used to determine how to modify the associated SSI system property.

In lines 9 and 10, the procedure preserves the consistent triples of an endorsed task, a constraint, and a direction within the set $R_{consist}$ and sets the flag of consistency ($flag$) to `true`. Multiple constraints can be consistent with a single endorsed task, and all pairs should be maintained. Then, the condition at line 11 is used to examine the consistency flag ($flag$). If consistency is detected, the procedure will count the number of endorsed tasks with consistent SSI system properties and store them in the variable *degree*. If the endorsed task does not have a consistent constraint, then the missing endorsed task will be added to the set $E_{missing}$.

In lines 13, 14, and 15, the procedure uses the conditions to identify which category ($type \in \{``FC", ``PC", ``IC"\}$) the shared controls belong to. The shared control is fully consistent (FC) if the variable *degree* equals the number of endorsed tasks in the given shared control $|C|$, and partially consistent (PC) if the variable *degree* is greater than 0 but less than the number of endorsed tasks ($0 < degree < |C|$). If not, the shared control will be deemed inconsistent (IC).

At the conclusion of the procedure, a quadruple $r = (C,\ type,\ R_{consist},\ E_{missing})$ is created to collect all the resultant data required for further processing and returned by the procedure. For the purpose of elucidating Algorithm 4, a running example comparing the execution of the data erasure and rectification shared control with the persistence SSI system property is provided, as two formalized sets shown below.

c_1 = ({SSI management system}, {dispose if wish to, dispose if achieve objective}, {identity})
e_1 = ({system}, {erase if not necessary}, {personal data})

Based on Table 4.14, the procedure will use the EVALUATE() function at line 7 to determine that the terms “SSI management system” and “system” are associated. Identical situations exist for the terms “identity” and “data.” In addition, the DETERMINEASSOCIATION() function manually determines the relationship between the phrases “dispose if wish to” and ”erase if no necessary.” Even if they are stated differently, it is evident that they are associated. Because the endorsed task does not specify the accomplishment of objectives, the DETERMINEDIRECTION() function at line 8 reveals that the constraint c_1 contains more information than the endorsed task e_1. As a result, the function will assign “>” to the variable *sign*. Suppose the endorsed task e_1 is the only task consistent with any of the four SSI system properties. This shared control will be considered partially consistent (PC).

Algorithm 4 is applied to all derived shared controls to compare them with consolidated SSI system properties, and the IDENTIFYCONSISTENCY() procedure’s output quadruples is compiled in Table 4.15. The result demonstrates that shared controls are categorized differently. Non-repudiation shared control is classified as partially consistent, despite the fact that the only endorsed task is consistent. This result is justifiable because the “>” sign indicates that the shared control provides more information than the property, which is used to determine consistency. In conclusion, based on Algorithm 4, 4 fully consistent, 13 partially consistent, and 14 inconsistent shared controls are reported.

4.4.3 A Method to Develop CSSPS

The preceding section presented an algorithm for determining the degree of consistency between shared controls and SSI system properties across three categories. The outcome is sufficient for identifying the missing knowledge that the existing SSI system properties lacked. This section presents the second part of the proposed method for leveraging such knowledge to enhance SSI system properties and bring them into compliance with credible source documents. As shown in Algorithm 5, the proposed method is presented as a semi-automatic algorithm.

Table 4.15: Result of the identification of consistency among shared controls and SSI system properties.

Control C	**Category**	**Consistency Triple** $R_{\textbf{consist}}$	**Missing Task** $E_{\textbf{missing}}$
S.1.	PC	(S.1.3, CP.10.4, $>$)	S.1.1, S.1.2, S.1.4
S.2.	PC	(S.2.1, CP.10.4, $=$)	S.2.2, S.2.3
S.3.	FC	(S.3.1, CP.13.1, $=$)	-
S.4.	PC	(S.4.1, CP.10.5, $<$)	S.4.1, S.4.3, S.4.4
S.5.	IC	-	S.5.1, S.5.2
S.6.	IC	-	S.6.1, S.6.2
S.7.	PC	(S.7.1, CP.10.4, $>$)	-
S.8.	IC	-	S.8.1, S.8.2, S.8.3
S.9.	PC	(S.9.2, CP.11.1, $>$)	S.9.1
S.10.	PC	(S.10.3, CP.10.6, $>$)	S.10.1, S.10.2
S.11.	IC	-	S.11.1, S.11.2
S.12.	PC	(S.12.1, CP.10.6, $>$), (S.12.4, CP.10.6, $>$)	S.12.2, S.12.3
S.13.	IC	-	S.13.1, S.13.2
S.14.	IC	-	S.14.1
S.15.	IC	-	S.15.1, S.15.2, S.15.3, S.15.4
S.16.	IC	-	S.16.1, S.16.2, S.16.3
S.17.	IC	-	S.17.1
P.1.	PC	(P.1.1, CP.3.7, $=$), (P.1.2, CP.3.5, $<$)	P.1.3, P.1.4, P.1.5
P.2.	PC	(P.2.1, CP.8.1, $=$)	P.2.2, P.2.3
P.3.	IC	-	P.3.1
P.4.	IC	-	P.4.1, P.4.2
P.5.	PC	(P.5.1, CP.8.2, $>$)	P.5.2
P.6.	PC	(P.6.2, CP.5.1, $>$), (P.6.3, CP.5.3, $=$)	P.6.1, P.6.4
P.7.	IC	-	P.7.1, P.7.2
P.8.	FC	(P.8.1, CP.9.1, $=$)	-
P.9.	PC	(P.9.1, CP.5.2, $>$)	P.9.2, P.9.3, P.9.4, P.9.5
P.10.	PC	(P.10.1, CP.13.1, $>$)	P.10.2
P.11.	IC	-	P.11.1, P.11.2, P.11.3, P.11.4
P.12.	FC	(P.12.1, CP.4.1, $=$)	-
P.13.	IC	-	P.13.1, P.13.2, P.13.3, P.13.4
P.14.	FC	(P.14.1, CP.6.2, $=$)	-

Complete analysis results are provided in https://doi.org/10.5281/zenodo.6951404.

Algorithm 5 Procedure for developing CSSPS from the consistency.

```
Input: $R$ : A set of resultant quadruples indicating the consistency,
    $\mathcal{P}$ : A set of consolidated SSI system properties.
Output: $\mathcal{P}_{CSSPS}$ : A set of SSI system property that are improved to be
    compliant with source documents.
procedure developCSSPS($R, \mathcal{P}$)
    $\mathcal{P}_{CSSPS} \leftarrow \emptyset$
    for $(C, type, R_{consist}, E_{missing}) \in \mathcal{R}$ do
        if $type =$ "PC" then
            for $e \in E_{missing}$ do
                $\mathcal{P}_{CSSPS} \leftarrow$ determineFitOrCreate($C, e, \mathcal{P}, \mathcal{P}_{CSSPS}$)
            for $(c, e, sign) \in R_{consist}$ do
                if $sign = >$ then
                    $c_{merge} \leftarrow$ merge($c, e$)
                    $\mathcal{P}_{CSSPS} \leftarrow$ revise($c_{merge}, c, \mathcal{P}$)
        else if $type =$ "IC" then
            for $e \in E_{missing}$ do
                $\mathcal{P}_{CSSPS} \leftarrow$ DetermineFitOrCreate($C, e, \mathcal{P}, \mathcal{P}_{CSSPS}$)
    $\mathcal{P}_{CSSPS} \leftarrow$ reviseExisting($\mathcal{P}, \mathcal{P}_{CSSPS}$)
    return formulateText($\mathcal{P}_{CSSPS}$)
```

The developCSSPS() procedure is defined for revising and developing CSSPS using the identified consistency results. The procedure generates a set of improved SSI system properties that are compliant with the source documents ($\mathcal{P}_{CSSPS}$) on the inputs: a set of resultant quadruples indicating the identified consistency (R) and a set of consolidated SSI system properties ($\mathcal{P}$). Each resultant quadruple will be iterated and analyzed based on the category from the variable *type* at the beginning of the procedure.

If the shared control (C) is categorized as *type* = "PC" or partially consistent, the procedure will conduct the development on two conditions. First, each missing endorsed task e in the set $E_{missing}$ will be iterated, and the determineFitOrCreate() function will search the sets of consolidated SSI system properties ($\mathcal{P}$) and the most recent version of CSSPS ($\mathcal{P}_{CSSPS}$) for the fit SSI system property. If there is a fit property for the missing endorsed task, the function will add and tailor it as a new constraint in that property. The loop will then iterate over every consistent triple $(c, e, sign) \in R_{consist}$. This iteration will be used to determine if the consistent SSI system property contains less information (i.e., if the variable *sign* is assigned as $>$). If so, the merge() function must combine the different pieces of data in the endorsed task e with the corresponding SSI system property c to produce a merged constraint $c_{merge} = (S_{merge}, A_{merge}, O_{merge})$ where $S_{merge} = S_c \cup T_e$, $A_{merge} = A_c \cup F_e$, and $O_{merge} = O_c \cup I_e$. The revised or additional SSI system properties must be added to the set $\mathcal{P}_{CSSPS}$ at the conclusion of these conditions.

If the shared control (C) is classified as *type* = "IC" or inconsistent, the procedure will iterate through each missing endorsed task e in the set $E_{missing}$ and use the DETERMINEFITORCREATE() function to search for the fit property or create a new property for each missing endorsed task. At the conclusion of this procedure, a set of SSI system properties that are consistent with source documents $(\mathcal{P}_{CSSPS})$ will be generated, and the REVISEEXISTING() function will be used to gather all SSI system properties that have not been improved. The FORMULATETEXT() function will then be used to manually generate plaintext from the formalized structure. The purpose of this function is to meticulously compose texts using all the pertinent information from the formalized structure.

For the purpose of elucidating Algorithm 5, another running example is provided using the following consistent triple:

(({SSI management system}, {dispose if wish to, dispose if achieve objective}, {identity}), ({system}, {erase if not necessary}, {personal data}), >) $\in R_{consist}$

Assume the shared control corresponding to the above triple is partially consistent. As the $>$ direction is assigned, the MERGE() function will determine that the SSI system property contains more information than the endorsed task and will merge them. To elaborate the SSI system property constraint, however, it may be justifiable to include all functions from the endorsed task. The MERGE() function's resulting merged constraint is demonstrated below.

c_{merge} =({SSI management system}, {dispose if wish to or not necessary, dispose if achieve objective}, {identity})

Then, the REVISEEXISTING() function will modify the merged constraint to the existing persistence SSI system property. In addition, a manual formulation is requested by the FORMULATETEXT() function to convert the formalized structure to plaintext. Following is an illustration of the revised persistence SSI system property.

> **Persistence:** ... An *SSI management system* must *dispose identity data* if the corresponding user *wishes to* after it is *no longer necessary*, or after it *fulfilled its objectives* successfully. ...

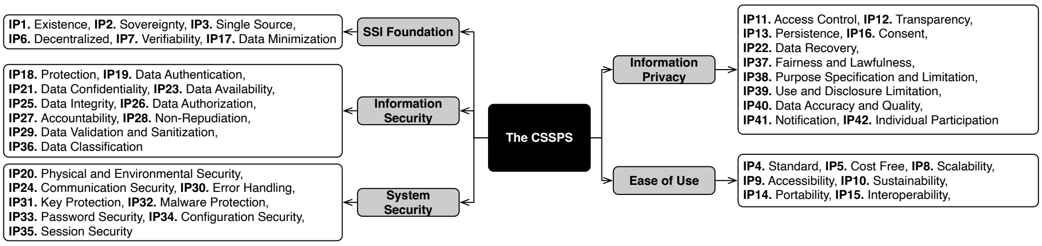

IP1. Existence, IP2. Sovereignty, IP3. Single Source,
IP6. Decentralized, IP7. Verifiability, IP17. Data Minimization
SSI Foundation
IP18. Protection, IP19. Data Authentication,
IP21. Data Confidentiality, IP23. Data Availability,
IP25. Data Integrity, IP26. Data Authorization,
IP27. Accountability, IP28. Non-Repudiation,
IP29. Data Validation and Sanitization,
IP36. Data Classification
Information Security
IP20. Physical and Environmental Security,
IP24. Communication Security, IP30. Error Handling,
IP31. Key Protection, IP32. Malware Protection,
IP33. Password Security, IP34. Configuration Security,
IP35. Session Security
System Security
The CSSPS
Information Privacy
IP11. Access Control, IP12. Transparency,
IP13. Persistence, IP16. Consent,
IP22. Data Recovery,
IP37. Fairness and Lawfulness,
IP38. Purpose Specification and Limitation,
IP39. Use and Disclosure Limitation,
IP40. Data Accuracy and Quality,
IP41. Notification, IP42. Individual Participation
Ease of Use
IP4. Standard, IP5. Cost Free, IP8. Scalability,
IP9. Accessibility, IP10. Sustainability,
IP14. Portability, IP15. Interoperability,

Figure 4.4: Overview of improved SSI system properties in CSSPS separated into five groups.

When Algorithm 5 is applied to all shared controls, CSSPS is generated, which contains 42 SSI system properties that are compliant with the controls of the source documents. The complete definitions of the improved SSI system properties in CSSPS is provided in Appendix A. To facilitate their adoption, SSI system properties are classified into the five categories: SSI foundation, information security, system security, information privacy, and ease of use. Fig. 4.4 provides an overview of SSI system property titles in CSSPS. Each SSI system property is identified by **IP**x, where x represents the xth SSI system property in CSSPS.

Using the proposed method for improving SSI system properties, the set of SSI system properties can be regularly updated based on the source document collection. Complying with CSSPS enables the SSI management system to safeguard information security and privacy in accordance with a credible source document.

Chapter 5

Modeling Secure and Privacy-Preserving SSI Data Sharing Events

This chapter will propose an approach for modeling data sharing events in SSI management systems in order to ensure their security and privacy specification. A high-level overview of the proposed approach is illustrated in Section 5.1 and other subsequent sections will detail the modeling method, the specification and the encoding in Alloy, respectively.

5.1 A Modeling Approach of the SSI Data Sharing Events

In this section, a comprehensive overview of the proposed modeling approach for SSI data sharing events is provided. The proposed method is intended for system analysts to convert the component diagrams representing the design of the target SSI management system into the formal model of the state transition system. The formal model is intended to be encoded in the Alloy specification language to enable the Alloy Analyzer to *automatically locate a system model that satisfies security and privacy specifications within a specified scope*. Figure 4.1 depicts the proposed approach, which consists of three modules: *modeling*, *specification*, and *encoding*.

The aim of the *modeling* module is to model SSI data sharing events derived from the design of the intended SSI management system in order to develop a model for analyzing security and privacy. This module assumes that the security and privacy constraints of the SSI management system, as outlined in the SSI principles and system properties, can be represented by

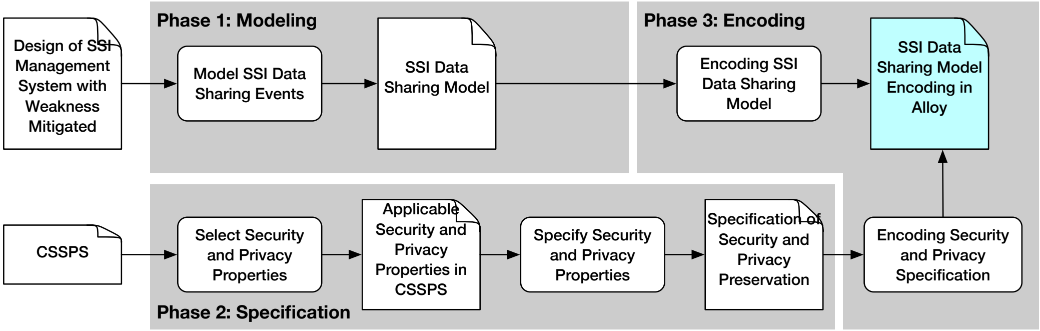

Design of SSI Management System with Weakness Mitigated
Phase 1: Modeling
Model SSI Data Sharing Events
SSI Data Sharing Model
Phase 3: Encoding
Encoding SSI Data Sharing Model
SSI Data Sharing Model Encoding in Alloy
CSSPS
Select Security and Privacy Properties
Applicable Security and Privacy Properties in CSSPS
Specify Security and Privacy Properties
Specification of Security and Privacy Preservation
Encoding Security and Privacy Specification
Phase 2: Specification

Figure 5.1: Overview of the modeling approach for secure and privacy-preserved SSI data sharing events.

the state of authorized access and data sharing among components. However, the sharing of data within the SSI management system is subject to specific constraints. For instance, the implicit sharing events reveal identity data in the form of identity claims, which should be constrained to disclose as little information as possible. It is difficult to use existing methods, such as [32, 33], to accommodate this type of unique constraint. This module provides a formal definition that extends the typical state transition system to assume the coverage of such constraints, as described in Section 5.2, to facilitate the modeling of SSI data sharing events. First, in Sections 5.2.1 and 5.2.2, the abstractions of components and data models, as well as the protocol-based data sharing events in the SSI management system, are analyzed to determine the model's completeness requirements. Then, on the basis of these abstractions and protocols, a formal definition of the custom-tailored state transition system is provided, as described in Section 5.2.3, and the procedure for developing the formal model is demonstrated in Section 5.2.4.

The aim of the *specification* module is to select pertinent information for specifying security and privacy in terms of the SSI data sharing model. As a result of this dissertation, it has been determined that the CSSPS provides multiple security and privacy SSI system properties. This module assumes that the security and privacy SSI system properties are compatible and can be implemented as constraints in the SSI data sharing model. For instance, the single source SSI system property specifies that identity data can only come from holders or users. This constraint can be expressed in terms of SSI data sharing events, since the holder is the only component that can share identity attribute objects. The procedure for defining security and privacy specifications is described in detail in Section 5.3. Sections 5.3.1 and 5.3.2 describe the selection criteria and method for transforming compatible SSI system properties into the context of the SSI data sharing model.

The aim of the *encoding* module is to provide a capability for ensuring security and privacy specifications in the SSI data sharing model across a variety of model scopes. It is possible that privacy and security issues may not arise in the initial few authorized access states. This module proposes the use of the Alloy specification language to encode the SSI data sharing model and makes use of its programmable predicates to automatically determine whether the encoding provides adequate security and privacy safeguards using the Alloy Analyzer tool. This is done to ensure the aforementioned constraints and to broaden the scope of the analysis. In Section 5.4, the encoding procedures for the formalized SSI data sharing model and its specification are described in detail.

5.2 SSI Data Sharing Model

As a system that implements the SSI model for manipulating personal data and digital identities, an SSI management system involves numerous data sharing events between its components. According to the SSI principles [4], data sharing events in the SSI management system must be authorized, with the owner's consent, and with minimal and selective disclosure. This dissertation hypothesizes that the above-mentioned notions should be incorporated in the analysis of the security and privacy. In this section, data model and component abstractions that are specific to the SSI management system are defined structurally and systematically. Then, the data model and component abstractions is proposed to be formalized as a state transition system from the design of the target SSI management system.

5.2.1 Data Model and Component Abstraction in SSI Management Systems

Although there is no standard that defines the data models and components of SSI management systems, the community has reached a consensus. In general, the SSI management system should be constructed according to standards for verifiable credentials [5]. Ferdous et al. [15] rigorously examine the concept of the SSI model and propose data models, including SSI objects and partial identities. Similarly, Lopéz [7] meticulously analyzed the technological foundation and fundamental components of the SSI management system by conducting extensive domain research. Mühle et al. [95] conducted a survey regarding the essential components of an SSI. These mutually agreed-upon works demonstrated that SSI management systems are likely to share components and data models. In order to thoroughly comprehend and model

SSI data sharing events, a domain analysis on existing work is conducted and derives abstractions of common data models and components as follows.

- A *holder* (also known as a user [7, 95], a subject [4], or an entity [15]) represents the owner of the identity data, which may be either the data subject or the agent. The holder is the most essential component for operating the SSI management system, including the input and utilization of identity data objects.
- An *identity wallet* (also known as a personal identity provider [15]) is a technical component that stores and manages identity data under the control of its holder.
- An *issuer* (also known as a claim-issuer [95] or validator) identifies the human, organizational, or system issuer components responsible for validating identity claims.
- A *verifier* (also known as a service provider [4, 7, 15] or claim-verifier [95]) represents identity consumption components, which may be humans, organizations, or systems, that require identity data for service authentication and authorization.
- A *distributed ledger*, also known as a blockchain [4, 7, 95], refers to the technical components that maintain decentralized and distributed data registries across peer-to-peer networks.

These components are mentioned and analyzed in numerous research studies, and they can be seen that there is consensus regarding them. Although other components, like functional components (such as authentication or registration), were emphasized in some works [95], this research determined that they were not explicitly relevant because they are not widely adopted.

Alternatively, the data models of SSI management systems from existing research can also be derived. In general, the primary data models adhered to the verifiable credential [5] and DID [6] open standards, but certain works cited the existence of unique data objects. Only common data models that can abstract the SSI management system were considered. Following is an abstract of a common data model in SSI management systems.

- An *identity attribute* (also known as a partial identity [15] or an attribute [7]) is a key-value pair of identity data describing a digital characteristic of one or more subjects.
- A *self-sovereign identity* describes a set of identity attributes of one or more subjects.
- An *identity claim* (also known as a claim [15]) is a statement that attests to the possession of an identity attribute. The claim may not reveal the actual attribute, but it serves as proof without knowledge.

- A *verifiable claim*, or *VC* (also known as a verifiable credential [5]) specifies a data model by which an issuer converts an identity claim into a form that can be validated by a proof schema.
- An *identity presentation* (also known as a profile [15] or credential presentation [5]) is a subset of identity attributes and verifiable claims used for service authentication and authorization to verifiers.
- A *private key* (also known as a secret key) is a cryptographic key used for message verification or decryption in accordance with public-private key infrastructure.
- A *public key* identifies a cryptographic key for message signing or encryption in accordance with the public-private key infrastructure.
- A *decentralized identifier*, or *DID*, is a unique address for an SSI management system component.
- A *DID document* (also known as a DID registry [7]) identifies a data registry associated with a DID by publishing a public key on the distributed ledger. It is resolvable by a DID.
- A *proof schema* (also known as a cryptographic proof [7]) identifies a data object that can be used to cryptographically demonstrate the validity of an identity claim. The proof schema must be made publicly available on the distributed ledger.
- A *service token* denotes a proof key that a verifier provides to a holder in order to authorize the holder's access to the service after identity presentation authentication.

After identifying the data model and component abstraction, a comprehensive structure for representing the SSI management system can be obtained. Figures 5.2 and 5.3 present a class diagram to illustrate the structural relationship between data objects and components. Many associations exist between the classes of the SSI management system's components and data objects. For instance, the holder class has a "share with" relationship with the identity wallet class to denote the holder's ownership of the identity wallet and they will share date to each other. These two class diagrams will be referenced in subsequent definitions of the SSI data sharing model.

5.2.2 Protocol-based Data Sharing Event

The data model and component abstractions provide a common understanding of the structure within the SSI management system. However, the notion of the SSI model also provides certain protocols to define the functionality of claim validation. These protocols are necessary to achieve the essential functionality. To generalize these protocols, this dissertation analyzes nu-

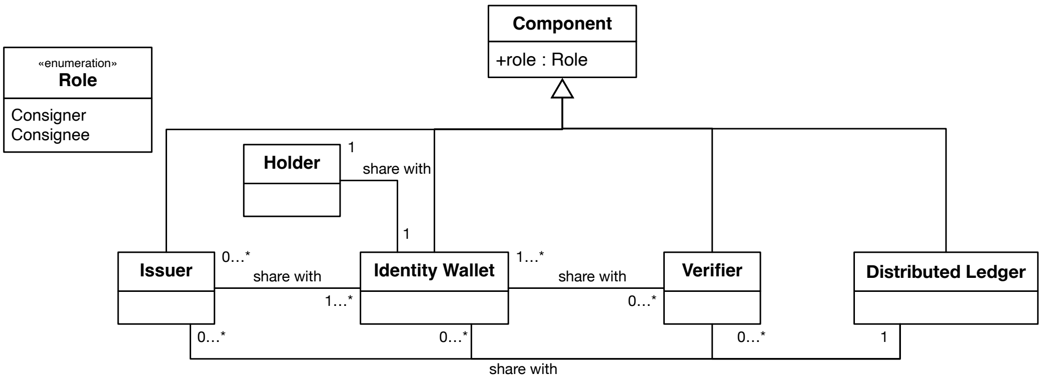


Figure 5.2: Class diagram for component abstraction.

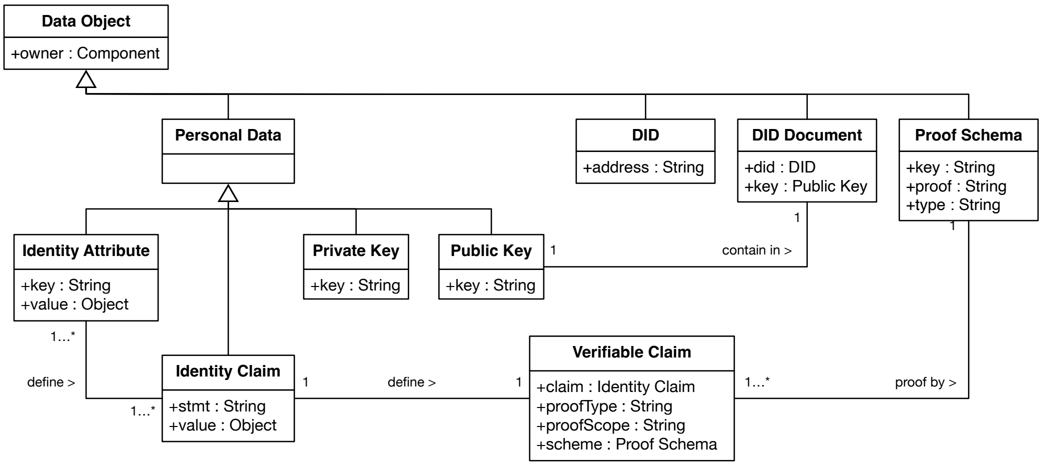


Figure 5.3: Class diagram for data model.

Table 5.1: Protocol-based SSI data sharing events among components.

Consigner	**Shared Data Object**	**Consignee**
Holder	Identity Attribute, SSI, Identity Claim	Identity Wallet
Identity Wallet	DID Document, Public Key	Distributed Ledger
Identity Wallet	Identity Claim, DID	Issuer
Issuer	DID Document, Public Key, Proof Schema	Distributed Ledger
Issuer	Verifiable Claim, DID	Identity Wallet
Identity Wallet	Identity Presentation, DID	Verifier
Verifier	DID Document, Public Key, Identity Presentation	Distributed Ledger
Verifier	DID, Service Token	Identity Wallet

merous research works on the design of the SSI model (e.g., [11, 15, 96]) and summarizes the common protocol-based data sharing events, as illustrated in Table 5.1. The table shows three groups of information on a sharing event: consigners, data objects, and consignees. A row in the table can be interpreted as consigners share data objects to consignees. For instance, a holder may share an identity claim with an issuer. Based on these protocols, it can be recognized that they share four different forms that are specific to SSI data sharing events, including:

- The *raw sharing form* represents a sharing of plaintext or readable data that exposes actual personal information. This form requires the consigner's permission before disclosing their information.
- The *implicit sharing form* represents the sharing of concealed data objects that do not expose the actual personal information. Sharing zero-knowledge proofs, for example, is not described in the actual data.
- The *broadcast sharing form* indicates the sharing of public data to all subscribers. This form is specific to the sharing of immutable data via peer-to-peer networks using distributed ledger technology.
- The *revocation sharing form* represents the opposite of data sharing. This form requires consignees to release and rectify shared data.

These forms will define how events involving protocol-based data sharing occur. For instance, the sharing of DID documents between an identity wallet and a distributed ledger will use the broadcast sharing form to make the DID document public to the network. In the modeling of SSI data sharing events, all four forms and protocol-based data sharing events should be taken into account to ensure that the SSI data sharing model cover the uniqueness of the SSI management system.

5.2.3 Formalization of SSI Data Sharing Model

Formalizing the SSI data sharing model facilitates model analysis and finding, particularly when the SSI management system's data sharing states are complex. For instance, ensuring information security and privacy when multiple issuers participate in claim validation is complex and difficult.

In contrast to previous research [33], in which instantaneous snapshots of privacy policies are defined and analyzed as system states, it can be contended that SSI management systems are more concerned with actual data handling than with policy enforcement. This section proposes a method for modeling protocol-based SSI data sharing events that takes into account the peculiarities of SSI management systems.

A system state is typically used to represent how the current configuration of a system operates or performs. This dissertation proposes that a system state in the SSI data sharing model is the current state of each component's data handling. In the first state, a holder's identity wallet, for instance, manages an identity attribute. One system state can transition to another in response to a triggering event. SSI data sharing events can be used as triggers for transitioning between authorized data access states. For example, the identity wallet will be accessible to a verifiable claim in the second state after an issuer has shared it at the end of the first state.

This concept aids in formalizing these phenomena to facilitate model analysis. Assume two abstracted sets of component identifiers ($\mathcal{C}$) and data object identifiers ($\mathcal{D}$) are prepared from the target SSI management system. In the SSI data sharing model, the system state is described as follows:

Definition 5.2.1. A system state indicates a snapshot of authorized access in each component as a set of tuples $\mathcal{S}$, denoted by $\mathcal{S} \subseteq \mathcal{C} \times \mathcal{D} = \{s : s = (c, d), c \subseteq \mathcal{C}, d \subseteq \mathcal{D}\}$

A system state $s = \{(\{\text{“student\_wallet\_1”}\}, \{\text{“grade\_claim”}\}), \ldots\}$, for instance, indicates that the “student_wallet_1” component is handling the “grade_claim” data object at the given time. Consider an enumeration $\mathcal{F} = \{$“Raw”, “Implicit”, “Broadcast”, “Revoke”$\}$ that represents four data sharing forms. The following is a formal definition of the SSI data sharing event that is used as a trigger for transitioning system states.

Definition 5.2.2. An SSI data sharing event indicates a trigger of state transition when a consigner component shares one or more data objects to a consignee component as a set of quadruples $\mathcal{E}$, denoted by $\mathcal{E} \subseteq \mathcal{F} \times \mathcal{C} \times \mathcal{C} \times \mathcal{D} = \{e : e = (f, c_{consigner}, c_{consignee}, d), f \in \mathcal{F}, d \in \mathcal{D}, \text{and } c_{consigner}, c_{consignee} \in \mathcal{C}\}$

An SSI data sharing event $e = \{(\{\text{“Implicit”}\}, \{\text{“student\_wallet\_1”}\}, \{\text{“university\_issuer\_1”}\}, \{\text{“grade\_claim”}\})\}$, for instance, depicts the triggering event in which the “student_wallet_1” component shares the “grade_claim” data object with the “university_issuer_1” component. If the above event transitions a system state to another, an example of the change of authorized access can be illustrated as:

$$\begin{aligned}
s_1 \quad &= \{(\{\text{“student\_wallet\_1”}\}, \{\text{“grade\_claim”}\})\} \\
\downarrow e \quad &= \{(\{\text{“Implicit”}\}, \{\text{“student\_wallet\_1”}\}, \{\text{“university\_issuer\_1”}\}, \\
&\quad \{\text{“grade\_claim”}\})\} \\
s_2 \quad &= \{(\{\text{“student\_wallet\_1”}\}, \{\text{“grade\_claim”}\}), \\
&\quad (\{\text{“university\_issuer\_1”}\}, \{\text{“grade\_claim”}\})\}
\end{aligned}$$

The “university_issuer_1” component is added to the second state (s_2) and handles the “grade_claim” data object that has been shared by the event e_1. Using the preceding notation, the following model of SSI data sharing events using a state transition system can be formed.

Definition 5.2.3. An SSI data sharing model is a labelled state transition system denoted by a quadruple $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{E}, \Lambda \rangle$, in which:

- $\mathcal{S}$ indicates a set of system states according to Definition 5.2.1,
- $\mathcal{S}_0$ indicates a set of initial system states s.t. $\mathcal{S}_0 \subseteq \mathcal{S}$ and $\mathcal{S}_0 \neq \emptyset$,
- $\mathcal{E}$ indicates a set of SSI data sharing events according to Definition 5.2.2,
- $\Lambda \subseteq \mathcal{S} \times \mathcal{E} \times \mathcal{S}$ indicates a set of transitional relations.

For instance, a transitional relation $\lambda = (\{s_1\}, \{e\}, \{s_2\}) \in \Lambda$ represents the transition of system states from s_1 to s_2 triggered by the SSI data sharing event e. This formal definition for the SSI data sharing model is sufficient to cover data model and component abstractions, as well as the protocols. It should be able to be a source for analyzing security and privacy constraints.

5.2.4 Preparation of the SSI Data Sharing Model

In order to model the SSI data sharing events as a state transition system in real-world situation, knowledge about the design of the SSI management system should be prepared. As data sharing events can occur among system components in the high-level abstraction, this dissertation suggests that the design of the target SSI management system should be represented in a component diagram to show provided and required interfaces of data objects. The component diagram is sufficient to input the overview of data objects

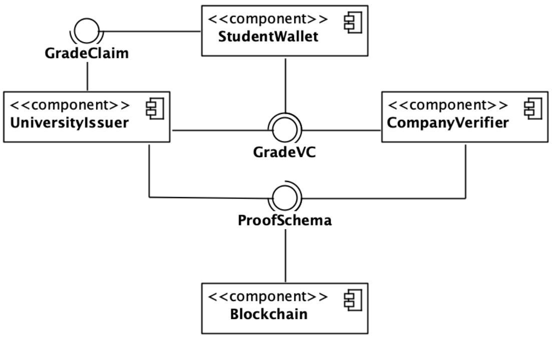


Figure 5.4: Example of a component diagram for the job application SSI management system that can be used to input the modeling approach.

and sharing events and can be directly converted into the proposed state transition system.

To make a clear understanding on how the design of real-world SSI management system can be modeled as an SSI data sharing model, this section provides an example of a component diagram showing an implementation of the SSI management system for the job application. Assume the component diagram shown in Fig. 5.4 is designed by a system analyst and attempt to be analyzed for secure and privacy-preserving SSI data sharing events. The diagram shows that four components, including a student wallet, a university issuer, a company verifier, and a blockchain, are defined and they share a grade claim, a grade verifiable claim, and a proof schema.

The above design explicitly shows multiple SSI data sharing events, but it cannot provide the state of data sharing. In order to get the knowledge of data sharing states, an analyst has to instantiate a use scenario based on the provided design. This scenario should represent expected execution of the design comprehensively. As shown in Fig. 5.5, the state transition system of the SSI data sharing model for the instantiated use scenario is depicted using a state diagram.

The state diagram depicts numerous transitions resulting from various SSI data sharing events. For instance, an event e_1 in the implicit sharing form transitions from the s_0 state to the s_1 state when a student wallet and a university share a grade claim. Another illustration is the self-transition, in which the event e_2 shares a grade-verifiable claim with the university itself. This could be indicative of the creation of a data object. Event e_4 on the revocation sharing form demonstrates that the student wallet requires the university to rectify the grade claim. It is evident that the university does not handle identity claim in the state s_4.

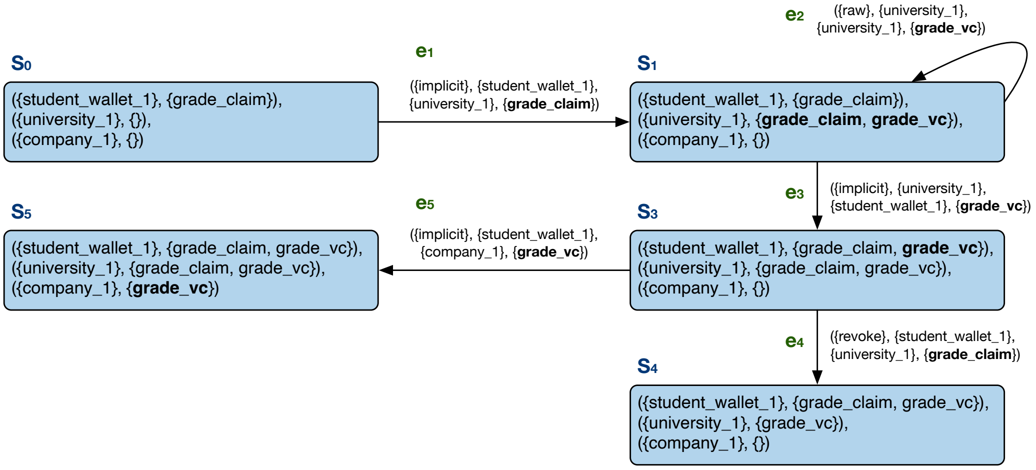

e2
({raw}, {university_1},
{university_1}, {grade_vc})
S0
e1
({implicit}, {student_wallet_1},
{university_1}, {grade_claim})
S1
({student_wallet_1}, {grade_claim}),
({university_1}, {}),
({company_1}, {})
({student_wallet_1}, {grade_claim}),
({university_1}, {grade_claim, grade_vc}),
({company_1}, {})
e3
({implicit}, {university_1},
{student_wallet_1}, {grade_vc})
S5
e5
({implicit}, {student_wallet_1},
{company_1}, {grade_vc})
S3
({student_wallet_1}, {grade_claim, grade_vc}),
({university_1}, {grade_claim, grade_vc}),
({company_1}, {grade_vc})
({student_wallet_1}, {grade_claim, grade_vc}),
({university_1}, {grade_claim, grade_vc}),
({company_1}, {})
e4
({revoke}, {student_wallet_1},
{university_1}, {grade_claim})
S4
({student_wallet_1}, {grade_claim, grade_vc}),
({university_1}, {grade_vc}),
({company_1}, {})

Figure 5.5: Example of the state transition system as a state diagram indicating an SSI data sharing model.

5.3 Specification of Security and Privacy in SSI Data Sharing Model

The formalization of SSI data sharing events aids in modeling the SSI management system in an analyzable manner. However, the number of states can be increased gradually over time and make the manual analysis more difficult. Furthermore, the formalization must possess some quality aspects to align with the SSI guiding principles and system properties. This section will describe how SSI system properties are selected for specifying the SSI data sharing model's information security and privacy requirements. The selected information security and privacy system properties are then defined in terms of the state transition system.

5.3.1 Selection of Security and Privacy Properties Related to SSI Data Sharing Model

The compliance of SSI system properties with respect to credible source documents was improved and proposed as the CSSPS in this work. Information security and privacy SSI system properties can be used to restrict SSI management systems, according to the results. Some of these system properties are also advantageous for bolstering the privacy and security of SSI data sharing events. For instance, the persistence system property that states: "An SSI management system must discard or permanently destroy identity data

Table 5.2: Analysis of SSI system properties in CSSPS for relating SSI data sharing events.

SSI System Property	Related?	SSI System Property	Related?
IP1. Existence	No	IP22. Data Recovery	No
IP2. Sovereignty	No	IP23. Data Availability	No
IP3. Single Source	Yes	IP24. Communication Security	No
IP4. Standard	No	IP25. Data Integrity	Yes
IP5. Cost Free	No	IP26. Data Authorization	No
IP6. Decentralized	Yes	IP27. Accountability	No
IP7. Verifiability	No	IP28. Non-Repudiation	No
IP8. Scalability	No	IP29. Data Validation & Sanitization	No
IP9. Accessibility	No	IP30. Error Handling	No
IP10. Sustainability	No	IP31. Key Protection	Yes
IP11. Access Control	Yes	IP32. Malware Protection	No
IP12. Transparency	No	IP33. Password Security	No
IP13. Persistence	Yes	IP34. Configuration Security	No
IP14. Portability	No	IP35. Session Security	No
IP15. Interoperability	No	IP36. Data Classification	No
IP16. Consent	No	IP37. Fairness and Lawfulness	No
IP17. Data Minimization	Yes	IP38. Purpose Specification & Limitation	No
IP18. Protection	No	IP39. Use & Disclosure Limitation	Yes
IP19. Data Authentication	No	IP40. Data Accuracy & Quality	No
IP20. Physical & Environmental Security	No	IP41. Notification	No
IP21. Data Confidentiality	No	IP42. Individual Participation	No

Note: Justification of how we decide SSI system properties are related to SSI data sharing events is provided in https://doi.org/10.5281/zenodo.6951404.

if the corresponding user wishes to modify, revoke, or delete them over time or after the purposes for which they were created have been fulfilled." can be checked to ensure that all subsequent system states are triggered after the event in the revocation sharing form. However, some system properties, such as "physical and environmental security," are difficult to reflect and analyze via SSI data sharing events.

On the basis of this justification, suitable security and privacy SSI system properties can be selected with respect to the SSI data sharing model and use them to constrain events. This dissertation assumes that SSI system properties are suitable to constrain SSI data sharing events if their target is related to data objects and the manipulation of them. An analysis on all SSI system properties in CSSPS is conducted and the results in Table 5.2 report which SSI system properties that relate to information security and privacy and are suitable for analyzing SSI data sharing events.

The results demonstrate that *eight* SSI system properties are related and can be reflected in the SSI data sharing model. Using the *single source* system property as an example, the holder must be the only component that shares an identity attribute with an identity wallet. Another example is

Table 5.3: Definitions of the selected SSI system properties in terms of SSI data sharing model.

SSI System Property	Definition in Terms of SSI Data Sharing Model
IP3. Single Source	Holders and identity wallets must always be consigners when identity attributes and identity claims are involved in an SSI data sharing event.
IP6. Decentralized	Only after a proof scheme has been shared on the distributed ledger may verifiable claims be created.
IP11. Access Control	Identity attributes and SSI must always be managed by the corresponding identity wallet and cannot be revoked.
IP13. Persistence	Data objects managed by a component must be persisted until the revocation sharing event is held by the component as the consignee.
IP17. Data Minimization	In all implicit sharing events, the sharing data must be either an identity claim or a verifiable claim.
IP25. Data Integrity	A DID sharing event must always precede other SSI data sharing events for a given pair of components.
IP31. Key Protection	All private keys must not be shared or involved in any SSI sharing event, unless the consigner and consignee are identical.
IP39. Use and Disclosure Limitation	Consignees must not share the received data objects to others (without the consent of their owners).

the *key protection* system property that prohibits components from sharing private keys. Only these eight system properties will be used to define the information security and privacy of the SSI data sharing model in this work.

5.3.2 Definition of Security and Privacy Properties

The preceding section lists the SSI system properties that can be reflected by the SSI data sharing model. These SSI system properties can be defined in accordance with the formalized state transition system, according to the findings. In this section, these SSI system properties are defined so that the SSI data sharing model can assert their presence.

Since the context of state transition system is applied to the SSI data sharing model, security and privacy properties must be reflected from the data structure and access control within the model. For example, all implicit-form events must involve with only an identity claim or verifiable claim. As a result, relational logic is sufficient to predicate such properties. To this end, the selected SSI system properties should be converted into relational logic-capable definitions in order to apply with the SSI data sharing model. This section proposes the definitions of the converted eight SSI system properties in Table 5.3. These properties are proposed under two conditions, which are: (1) the definition must explain only about how data objects are shared among components; and (2) the definition must be justifiable to align with the original SSI system properties.

All selected SSI system properties use the context of the SSI data sharing model to define how information security and privacy are protected. These

SSI system properties can be characterized as pre-conditions, post-conditions, or invariants. and serve as security and privacy specification of the SSI data sharing model.

5.4 Encoding of SSI Data Sharing Model and Properties in Alloy

It is possible to compare and contrast the SSI data sharing model and the specification of security and privacy. When the SSI data sharing model is in a complex system state, however, manual analysis becomes difficult to cover all of the cases. In order to address this issue, this dissertation proposes encoding both the SSI data sharing model and its specification of security and privacy in the Alloy specification language and using the encoding to find models that conform to the specification. It is required for system analysts to convert and encode the SSI data sharing model in Alloy. The following sub-sections will describe how the SSI data sharing model (Section 5.4.1) and its specification (Section 5.4.2) are encoded stepwisely.

5.4.1 Encoding of SSI Data Sharing Model

SSI data sharing model is the main part to be analyzed, which is defined in Definition 5.2.3. The unique characteristics of the SSI management system and the corresponding SSI data sharing model must be imitated in the encoding. However, it is difficult to incorporate full relationships between data objects and component abstractions because the encoding will be too complex and take much time to analyze by the tool. Consequently, this dissertation declares component and data types as enumerations in Alloy instead, as below.

```
enum ComponentType {
        Holder, IdentityWallet, Issuer, Verifier, DistributedLedger }
enum DataType { IdentityAttribute, SSI, IdentityClaim, VerifiableClaim,
        IdentityPresentation, PublicKey, PrivateKey, ProofSchema, DID,
        DIDDocument, ServiceToken }

abstract sig Component { compType: one ComponentType, own: set DataObject }
abstract sig DataObject { dataType: one DataType, ownedBy: one Component }
```

Component type and data type enumerations reveal the distinct components and data models utilized by the SSI management system. Two abstract component and data object signatures are declared. A component signature will have the “compType” relation with a component type item and the “own” relation with a collection of data objects that the component handles. A data object signature will also have the “dataType” relation with a data type item in the enumeration, as well as an “ownedBy” relation with a component signature to indicate the data object’s owner or creator. Since component and data object signatures are declared abstract, they can be extended to conform to the actual implementation of the SSI management system.

As a state transition system, SSI data sharing model can be defined as a quadruple, which can be encoded in Alloy directly. The encoding of the SSI data sharing model’s quadruple is shown as follows.

```
enum SharingForm { Raw, Implicit, Broadcast, Revoke }
sig State { component: set Component }
sig Event { form: one SharingForm, consigner: set Component,
    consignee: set Component, sharedData: set DataObject }

one sig SDSM {
        s0: set State,
        states: set State,
        events: set Event,
        transitions: State -> State }
```

Initially, the `SharingForm` enumeration for the SSI sharing form, which includes the `raw`, `implicit`, `broadcast`, and `revoke` items, is declared. The `State` signature for the system state is then declared based on Definition 5.2.1. The `State` signature includes a “`component`” relation that corresponds to a collection of components maintained by the system state. Additionally, the `Event` signature for SSI data sharing events (based on Definition 5.2.2), which consists of four relations: `form`, `consigner`, `consignee`, and `sharedData`, is declared. The `form` relation identifies a relation to a particular type of SSI sharing form as specified in the enumeration. The `consigner` and `consignee` relations represent relations to sets of components that are, respectively, consigners and recipients of the data sharing. The `sharedData` relation represents a relation to a collection of data objects corresponding to an SSI data sharing event.

A singleton `SDSM` signature for the SSI data sharing model is required to be declared in accordance with Definition 5.2.3. The `SDSM` signature includes the following four relations for the SSI data sharing model:

- the `s0` relation is a relation to a set of initial states,
- the `states` relation is a relation to a set of all system states,
- the `events` relation is a relation to a set of all SSI data sharing events that trigger state transitions, and
- the `transitions` relation is a relation to a set of all possible state transitions (`State -> State`).

The `SDSM` signature conforms to the definition with the exception of the transition relation definition. This dissertation decided to model a transition as a change from one state to another rather than as a combination of states and events (`State -> Event -> State`) because events can be used as predicates for pre- and post-conditions for such transitions.

Moreover, the protocols in the SSI management system must characterize the SSI data sharing events followed Table 5.1. To regulate these characteristics of the SSI data sharing model, the following predicates that define valid protocols are declared.

```
pred validProtocol [e: Event] {
        all a, b: Component, d: DataObject | a.compType = Holder and
                b.compType = IdentityWallet and a in e.consigner and
                b in e.consignee and d in e.sharedData
                implies d.dataType = IdentityAttribute + IdentityClaim
       ...}
```

The preceding predicate demonstrates a block constraint that restricts SSI sharing events between an identity wallet and a holder to only incorporating identity attributes or identity claims. In this predicate, all SSI data sharing events are controlled over system states.

In addition to the sharing protocols, different data sharing forms are restricted by different predicates. As four forms of sharing in Section 5.2.2 were defined, three predicates for each form of sharing should be included to specify their pre- and post-conditions. For example, the following code block demonstrates three predicates for the implicit sharing form.

```
pred implicit_sharing [a, b: State, e: Event] {
        e.form = Implicit
        pre_implicit [a, e]
        post_implicit [a, b, e] }
pred pre_implicit [a: State, e: Event] {
        e.consigner in a.component
        // The consigner owns the data object before sharing
        all c : Component, d: DataObject | c in e.consigner and
                d in e.sharedData implies c in a.component and d in c.own
        // The shared data must be identity claims or verifiable claims
        all d : DataObject | d in e.sharedData implies
                d.dataType = IdentityClaim + VerifiableClaim }
pred post_implicit [a, b: State, e: Event] {
        e.consigner in b.component
        e.consignee in b.component
        // All shared data must be persisted in the consignee after sharing
        all c : Component, d : DataObject | c in e.consignee and
                c in b.component and d in e.sharedData implies d in c.own }
```

Following is a description of each predicate regarding the implicit sharing form in the SSI data sharing model.

- The `implicit_sharing` predicate specifies that the form of the corresponding SSI data sharing event must be implicit. The `pre_implicit` and `post_implicit` predicates are then applied to system states and events.

- The pre_implicit predicate is a precondition for the SSI data sharing event in its implicit form. It declares three constraint blocks to limit: (1) that the consigner of the event must already be in the previous state; (2) that the consigner must have owned the sharing data object in the previous state; and (3) that the sharing data object must be the type of identity claims or verifiable claims that require the implicit sharing form.
- The post_implicit predicate is the post-condition for the SSI data sharing event in the form of implicit sharing. Additionally, it declares three constraint blocks to ensure that components in the consigner and consignee relations of the event are in the subsequent system state and that the shared data object is owned by the consignee.

These predicates are adequate to constrain each form of sharing as we expected. However, there are some constraints that are also necessary for both the unique characteristics of the SSI management system and the nature of state transition system. We decided to add these additional constraint blocks as a fact, predicates, and functions in the following encoding.

```
fun initialState : State { SDSM.s0 }
fun allStates : State { SDSM.states }
fun allEvents : Event { SDSM.events }
fun nextState : State -> State { SDSM.transitions }

pred init [s: State] { s.component.own = none }
pred transition [a, b: State, e: Event] {
        raw_sharing [a, b, e] or implicit_sharing [a, b, e] or
        broadcast_sharing [a, b, e] or revoke_sharing [a, b, e] }

fact {
        // Set the initial states
        all s: State | s in initialState iff init [s]
        // Set the possible transitions in the SSI data sharing model
        all a, b: State, e: Event | a->b in nextState iff transition [a,b,e]
        // States are always having transitions
        all a, b: State | a in allStates and b in allStates implies
                a->b in nextState
       ...
        // All sharing must be under a valid protocol
        all e: Event | validProtocol [e]
        // There must be only one distributed ledger in the system
        one c : Component | c.compType = DistributedLedger
        // Private key must not be shared to other components
        all e: Event, d: DataObject | d in e.sharedData and
                d.dataType != PrivateKey
}
```

The preceding code block demonstrates that functions to refer to all SSI data sharing model relations (init and transition) are declared. In addition, two predicates are declared to restrict the definition of initial states and expected transition types. These predicates are applied to facts that serve as model invariants. For instance, a fact guarantees that all transitions between two system states must be in raw, implicit, broadcast, or revoke form, and that all events must satisfy the `validProtocol` predicate.

With the above encoding, the formalized SSI data sharing model can be represented in Alloy and automatic analysis and model finding using the Alloy Analyzer are enabled. Explicit constraints are defined using facts, predicates, and functions. Additionally, it is necessary to define information security and privacy properties as implicit constraints in Alloy. In the next section, the encoding of the specification of information security and privacy will be elaborated.

5.4.2 Encoding of General Properties of Secure and Privacy-Preserving SSI Data Sharing Model

As the specification of information security and privacy is examined in Section 5.3, eight SSI system properties that the SSI data sharing model can reflect are obtained. In this section, these SSI system properties are shown to be converted into the Alloy syntax.

Security and privacy specification are considered as implicit constraints that the SSI data sharing model must or must not contain. On the basis of the specified definitions in Table 5.3, this dissertation assumes that the SSI data sharing model maintains security and privacy if the tool is unable to identify models that follows the negated SSI system properties. Using this premise, eight general SSI system properties can be declared in Alloy. The following general properties can be used directly or extend to the custom SSI data sharing model.

1) Single Source SSI System Property. This property specifies that "Holders and identity wallets must always be consigners when identity attributes and identity claims are involved in an SSI data sharing event." This property guarantees that the holder is the only source of identity data, including identity attributes and claims. This dissertation contends that this property is violated if a holder is not the one who provides the identity attributes or identity claims to an identity wallet. This property can be encoded as follows.

```
pred singleSourceProperty {
        all e: Event, a, b: Component, d: DataObject | d in e.sharedData and
                a in e.consigner and b in e.consignee implies
                a.compType != Holder and b.compType = IdentityWallet and
                d.dataType = IdentityAttribute + IdentityClaim }
```

2) Decentralized SSI System Property. This property states, “Only after a proof scheme has been shared on the distributed ledger may verifiable claims be created.” It requires the availability of a proof scheme prior to transforming identity claims into verifiable claims. This dissertation negated this property so that it will be violated if the proof schema is not owned by the distributed ledger prior to the subsequent event in which verifiable claims are shared. This property can be encoded in Alloy as follows:

```
pred decentralizedProperty {
        all e: Event, a, b: State, c: Component, d1, d2: DataObject |
                transition [a,b,e] and c in a.component and
                c.compType = DistributedLedger and d1.dataType = ProofSchema
                and d2.dataType = VerifiableClaim implies
                d1 not in c.own and d2 in e.sharedData }
```

3) Access Control SSI System Property. This property specifies that “Identity attributes and SSI must always be managed by the corresponding identity wallet and cannot be revoked.” It ensures that identity attributes and SSIs can only be managed by the identity wallet and can only be revoked by its owner. This dissertation negated this property by stating that it will be violated if an SSI sharing event results in the revocation of identity attributes and SSIs by non-holder components. This property can be encoded in Alloy as follows.

```
pred accessControlProperty {
        no e: Event, a, b: Component, d: DataObject | b in e.consigner and
                a in e.consignee and d in e.sharedData and
                d.dataType = IdentityAttribute + SSI
                implies a.compType = IdentityWallet and
                e.form = Revoke and b.compType != Holder }
```

4) Persistence SSI System Property. This property specifies that “Data objects managed by a component must be persisted until the revocation sharing event is held by the component as the consignee.” Until they are revoked by an event, it would ensure that data objects are always handled in a component corresponding to system states. This dissertation assumes a violation of this property if the consignees of the event in the revocation sharing form do not handle the data object in its previous state. This property can be encoded as follows.

```
pred persistenceProperty {
        all s1, s2: State, e: Event, c: Component, d: DataObject |
                c in s1.component and transition [s1,s2,e] and
                e.form = Revoke and c in e.consigner implies d not in c.own }
```

5) Data Minimization SSI System Property. This property specifies that "In all implicit sharing events, the sharing data must be either an identity claim or a verifiable claim." This property ensures that the implicit sharing of identity claims and verifiable claims is indicated. If the SSI data sharing event is marked as implicit but the shared data is not identity claims and verifiable claims, this property will be violated. This property can be encoded in Alloy as follows.

```
pred dataMinimizationProperty {
        all e: Event, d: DataObject | d in e.sharedData and
                d.dataType != IdentityClaim + VerifiableClaim implies
                e.form = Implicit }
```

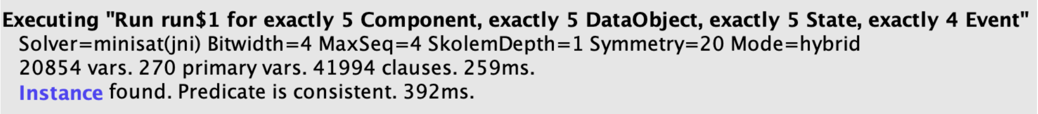
6) Data Integrity SSI System Property. This property states, "A DID sharing event must always precede other SSI data sharing events for a given pair of components." This property ensures that DID must be exchanged before other sharing events in order to prepare the public key for message authentication and encryption. Therefore, this dissertation assumes that this property is violated if the other component's DID is not handled prior to the sharing event. This property can be encoded in Alloy as follows.

```
pred dataIntegrityProperty {
        all s1, s2: State, e: Event, c1, c2: Component, d1, d2: DataObject |
                transition [s1, s2, e] and c1 in s1.component and
                c2 in s1.component and c1 in e.consigner and
                c2 in e.consignee implies d1 not in c1.own and
                d2 not in c2.own and d1.ownedBy = c2 and d2.ownedBy = c1 and
                d1.dataType = DID and d2.dataType = DID }
```

7) Key Protection SSI System Property. This property states, "All private keys must not be shared or involved in any SSI sharing event, unless the consigner and consignee are identical." This property merely satisfies the fundamental requirement of the public-private key infrastructure, which stipulates that private keys cannot be shared between components. If an event containing a private key as shared data occurs, this dissertation assumes this property will be violated. This property can be encoded in Alloy as follows.

```
pred keyProtectionProperty {
        all e: Event, d: DataObject, c: Component | c in e.consigner and
        c not in e.consignee and d in e.sharedData
        implies d.dataType = PrivateKey }
```

8) Use and Disclosure Limitation. This property states, "Consignees must not share the received data objects (without the owners' permission) with others." It ensures that components with which data objects are shared



```
Executing "Run run$1 for exactly 5 Component, exactly 5 DataObject, exactly 5 State, exactly 4 Event"
  Solver=minisat(jni) Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20 Mode=hybrid
  20854 vars. 270 primary vars. 41994 clauses. 259ms.
  Instance found. Predicate is consistent. 392ms.
```

Figure 5.6: Example of the output message after executing the run command.

will not share them without the owner's permission. This dissertation assumes that this property is violated if a subsequent event occurs in which the consignee of the initial event shares the same data object with other components. This property can be encoded in Alloy as follows.

```
pred useDisclosureLimitProperty {
        all s1, s2, s3: State, e1, e2: Event, c: Component, d: DataObject |
                transition [s1, s2, e1] and transition [s2, s3, e2] implies
                c in s2.component and c in e1.consignee and c in s3.component
                and d in e1.sharedData and c in e2.consigner and
                d in e2.sharedData and d.dataType != IdentityPresentation }
```

With the possession of these predicates defined by adhering to the chosen SSI system properties, the encoding of the SSI data sharing model[1] can be secured and privacy preserved. In the section that follows, the procedure to analyze the preceding encoding of the SSI data sharing model and the specification of security and privacy using the Alloy Analyzer will be explained in detail.

5.4.3 Model Finding with Alloy Analyzer

Since MIT developed Alloy Analyzer as an application interface for use as a model finder, Alloy Analyzer version 6 can be used as the model finding tool to analyze the SSI data sharing model. To initiate model instances based on the encoding, the following **run** command must be used.

```
run {} for exactly 5 Component, exactly 5 DataObject, exactly 5 State,
exactly 4 Event
```

After executing the Alloy Analyzer, an output message similar to the one shown in Fig. 5.6 is reported. The Alloy Analyzer is responsible for graph-based representations of model instances. However, it is advantageous for model analysis when the model scope and size are limited. In Fig. 5.7, for instance, a model instance visualized by the Alloy Analyzer is illustrated.

The model instance demonstrates that two system states transition into one another in response to an event. Five data objects are included within the SSI data sharing event. However, the preceding **run** command cannot

[1] Executable Alloy encoding is published on: https://github.com/chnpat/Alloy_SSI_Data_Sharing_Model.

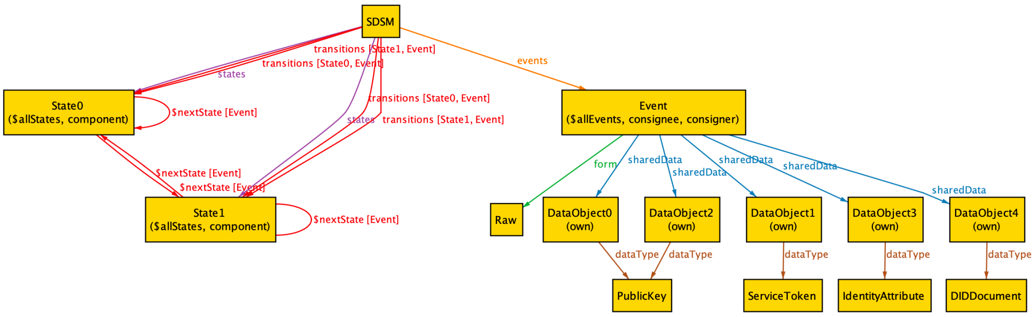
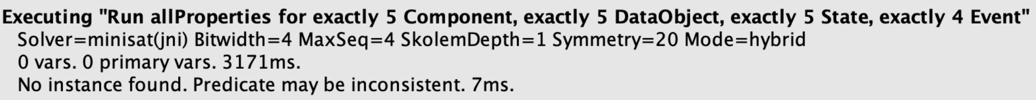


Figure 5.7: Example of a model instance of two system states and an event.



```
Executing "Run allProperties for exactly 5 Component, exactly 5 DataObject, exactly 5 State, exactly 4 Event"
   Solver=minisat(jni) Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20 Mode=hybrid
   0 vars. 0 primary vars. 3171ms.
   No instance found. Predicate may be inconsistent. 7ms.
```

Figure 5.8: Example of an output message showing that negated predicates are inconsistent.

guarantee that the model instance possesses the SSI system properties. As a result, an additional predicate is declared to compile all SSI system properties and use this predicate in the `run` command, as demonstrated below.

```
pred allProperties { singleSourceProperty and decentralizedProperty and
        accessControlProperty and persistenceProperty and
        dataMinimizationProperty and dataIntegrityProperty and
        keyProtectionProperty and useDisclosureLimitProperty }
run allProperties for exactly 5 Component, exactly 5 DataObject,
        exactly 5 State, exactly 4 Event
```

The Alloy Analyzer will generate a message (Fig. 5.8) indicating that no model instance can be created after the new `run` command is executed. Figure 5.7 depicts the output message. Since all selecteed SSI system properties are negated, the output message indicates that the SSI data sharing model complies with all predicates and no violation has occurred. On the other hand, it represents that the defined Alloy model of SSI data sharing model possessed adequate general SSI system properties.

As this dissertation proposes the SSI data sharing model, which is used to analyze for both security and privacy preservation, it can be discovered that it can automatically ensure information security and privacy as expected. The model is extendable or adaptable to the intended SSI management system implementation and uses the proposed encoding of the specification to protect the privacy and security of its data. The SSI data sharing model successfully imitates the protocol-based SSI data sharing events, and the SSI data sharing model successfully defines the information security and privacy specifications.

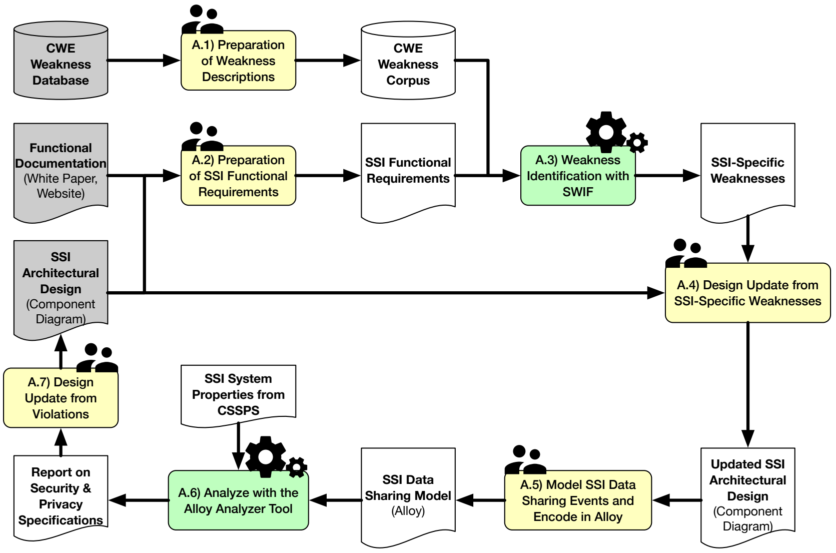

CWE Weakness Database
A.1) Preparation of Weakness Descriptions
CWE Weakness Corpus
Functional Documentation (White Paper, Website)
A.2) Preparation of SSI Functional Requirements
SSI Functional Requirements
A.3) Weakness Identification with SWIF
SSI-Specific Weaknesses
SSI Architectural Design (Component Diagram)
A.4) Design Update from SSI-Specific Weaknesses
A.7) Design Update from Violations
SSI System Properties from CSSPS
Report on Security & Privacy Specifications
A.6) Analyze with the Alloy Analyzer Tool
SSI Data Sharing Model (Alloy)
A.5) Model SSI Data Sharing Events and Encode in Alloy
Updated SSI Architectural Design (Component Diagram)

Figure 5.9: Overview of the Typical Integration Process.

In addition, the key advantage of the Alloy model is that it is lightweight and tractable, as you can comprehend how the SSI data sharing model is encoded and explore the graph visualization of the result manually.

5.5 Integration Process of Approach

As the modeling of SSI data sharing events is intended to utilize the output of the other parts, it must be interoperable and effectively integrated. This section will describe the typical integration process and variations that can be applied to the process in order to clarify how to integrate the proposed approach's parts.

5.5.1 Typical Integration Process

As described in Section 1.5, the proposed approach to analyzing security weaknesses and privacy preservation in the SSI management system consists of three main parts or solutions. The proposed approach is designated to be integrated as a batch process where the architectural design is updated by the recommended SSI-specific weaknesses and is used to model SSI data sharing events to analyze against the improved security and privacy SSI system properties. To clarify the means to use the proposed approach, this section provides a typical integration process, as shown in Fig. 5.9.

The figure illustrates that the typical process is designed to be iterative in order to continuously strengthen the security and privacy of the target SSI management system's design.

Input of the Typical Integration Process

To apply the typical integration process, it must be assumed that the target SSI management system's architectural design and feature documentation were created completely without regard for security or privacy. The architectural design and functional documentation are subject to modification based on the results of the analysis from the proposed approach. The inputs required to implement the proposed approach in the typical integration process consist of the three artifacts described below.

- *Common security weaknesses* - A compilation of the most recent version of common security weaknesses from the CWE database, including their natural language descriptions. This input should be gathered manually by downloading the files from the database's website, or automatically using any web crawling tool.
- *Functional documentation* - This input may consist of white papers or web-based documentation that describes the functionality of the target SSI management system. It is assumed that this document provides a comprehensive description of the provided functions, and that it matches the actual implementation.
- *Architectural design* - This input is intended to be a component diagram-representation of the complete design for component-level architecture. The architectural design that will serve as the process' input must contain the following information:
  - Technical components that will be necessary for the operation of the SSI management system should be included.
  - Data interfaces that indicate the provision and demand of data objects among technical components should be included.

  The component diagram must also be adequate to denote the occurrence of data sharing between technical components, according to the SSI model's functions.

The aforementioned inputs are necessary for the implementation of the proposed approach. This dissertation assumes that users of the proposed approach will manually prepare them in accordance with the target SSI management system. If the inputs are insufficient, the proposed approach may not be able to function at its optimal level.

Activities in the Typical Integration Process

To implement the proposed approach, the specified inputs and assumptions must be satisfied. Each activity's specifics are described as follows:

A.1) This activity is intended to collect descriptions of common security weaknesses from the CWE database and *manually* prepare a corpus of CWE weakness descriptions.

A.2) This activity is intended to analyze and convert the architectural design of the SSI management system, i.e. component diagrams, into a set of SSI functional requirements that must be *manually* entered into the SWIF implementation.

A.3) This activity is designed to enable the SWIF implementation as a tool for *automatically* identifying SSI-specific security weaknesses from common security weaknesses based on language correlations.

A.4) This activity is intended to analyze the identified SSI-specific weaknesses and use them as knowledge sources to update the architectural design in order to *manually* mitigate security weaknesses.

A.5) This activity is intended to model the SSI data sharing events derived from the updated architectural design as a state transition system of data sharing states and Alloy model in order to *manually* analyze them for security and privacy specifications.

A.6) This activity is designed to enable the Alloy Analyzer tool to *automatically* analyze security and privacy specifications violations (defined by the improved SSI system properties from CSSPS) in the Alloy model.

A.7) This activity is intended to *manually* update the architectural design of the target SSI management system by analyzing the violations reported by the Alloy Analyzer tool.

The typical integration process can be repeated by looping back from activity A.7 to activity A.2 in order to identify and mitigate any additional SSI-specific weaknesses that may arise after the update. A.3 and A.6 are designated as automatic for reducing the number of common security weaknesses to infer SSI-specific weaknesses and expanding the scope of SSI data sharing events to analyze security and privacy specifications that are challenging to manually analyze, respectively. The integration process described above is typical, but it can be modified based on user demands.

Output of the Typical Integration Process

There are multiple outputs following the execution of the integration process, including both intermediate and final results. These outputs are anticipated to yield benefits from the use of the proposed approach and serve as knowledge sources for improving the design of the SSI management system being investigated. Below is a summary of the outputs of a typical integration process and an explanation of how to use them.

- *A list of SSI-specific weaknesses* - Based on language correlations, this output identifies which common security weaknesses from the CWE database are SSI-specific. It can serve as sources for design analysis that aid system analysts in enhancing the security and privacy of the SSI management system under consideration.
- *A list of violated model instances* - This output is the result of the Alloy Analyzer tool, which provides graphical notations for model instances. System analysts can use the model instances to identify the SSI data sharing events associated with the violations and then modify the design to prevent these occurrences. This output should not, ideally, result from the analysis.
- *An updated SSI architectural design* - This output is an important outcome of the proposed approach that provides a secure and privacy-preserving architectural design to prevent security and privacy issues. System analysts are expected to utilize the previous two outputs to update the design accordingly.
- *Updated SSI functional requirements* - This output is an additional important outcome of the proposed approach, which provides secure and privacy-preserving functionality for the SSI management system. To prevent security and privacy issues from arising as a result of the analysis, it is expected that existing functional requirements will be revised or new functional requirements will be added.

The aforementioned outputs are the primary outcomes of the proposed approach based on the typical integration process, which are intended to aid system analysts in enhancing the design of the SSI management system.

5.5.2 Variation of Integration Process

As a combinatorial process, the proposed approach is applicable to a variety of process variants. In this section, potential variations of the proposed approach's integration procedure are described.

Manual security weakness analysis. It is possible that users of the proposed method will not be constrained by time and effort when identifying security weaknesses. Using the definition of language correlation, one can manually identify SSI-specific weaknesses. This variation applies to the variable activity A.3, which compares exhaustively common security weaknesses from the CWE database manually. This variation permits users to include subjective justifications and domain expertise to strengthen the validity of SSI-specific weaknesses. However, this variant's time-consuming and error-prone nature should be weighed against its benefits and accepted.

Another source of common security weakness. The reason the proposed approach targets common security weaknesses from the CWE database is because it provides sufficient resources for the SWIF implementation. Nonetheless, another database of common security weaknesses can also be used to expand the scope of weaknesses. To apply with a different weakness source, the variation must be applied to step A.1 in order to customize the weakness description based on the selected database. Another database, for instance, may not provide complete textual descriptions. Activity A.1 should manually prepare data to describe weaknesses using domain knowledge.

Solely analysis of security and privacy specifications. In some instances, the proposed approach may not be required in its entirety. Depending on the user's objectives, certain parts of the proposed approach can be omitted. A variation on the integration process is the use of the SSI data sharing modeling part to guarantee only the security and privacy specification. Therefore, the target SSI management system's architectural design can directly feed into the modeling activity and the typical integration process can begin with activity A.5. However, this variant may overlook security weaknesses that cannot be determined by SSI data sharing events.

Solely analysis of security weaknesses. Similar to the previous variant, the weakness identification part can be adopted independently. This variation is suitable for alternative designs of the SSI management system that have been created to ensure the security and privacy of SSI data sharing events. To apply this variation, only activities A.1 through A.4 of the typical integration process can be utilized to concentrate on the identification of weaknesses. If security and privacy are incorporated into the design, this variant reduces the time and effort required to model SSI data sharing events and analyze them using the Alloy Analyzer tool.

Using the variation described above, the integration process of the proposed approach can be chosen based on the users' needs. The variation can help simplify the process, but under the same assumptions, this dissertation recommends the typical integration process as the most suitable process.

Chapter 6

Evaluation and Discussion

This chapter will describe the used evaluation methods and results in order to demonstrate the advantages and limitations of the proposed approach. The internal parts of the proposed approach are evaluated from multiple perspectives, such as accuracy, performance, validity, and applicability. In addition, the overall applicability of the proposed approach to a case study is evaluated. Lastly, a dedicate section to discuss their advantages and limitations based on the evaluation results is provided.

6.1 Accuracy

6.1.1 Accuracy on Weakness Identification

The primary function of the weakness identification part of the proposed approach is to identify SSI-specific weaknesses. This feature must achieve an acceptable level of accuracy to ensure that the recommended SSI-specific weaknesses are similar to those identified manually. In addition, this dissertation expects that SWIF should be comparable to or superior to existing methods. Hariyanti et al. [27] is an alternative method to SWIF for identifying weaknesses. Therefore, this section will report the accuracy of SWIF relative to manual identification and compare it to an existing approach.

The evaluation is designed as an experiment to calculate an accuracy metric in comparison to the security analyst's ground truth as described in Section 3.6.1. The accuracy metric is also used as a comparative metric with the existing approach. To illustrate how SWIF differs from the existing approach, Table 6.1 provides a comparison of the approaches' various features. It can be realized that SWIF employed a distinct vectorization mechanism and introduced three additional recommendation features,

Table 6.1: Comparison in features between SWIF and the existing approach.

Approach	**Vectorization**	**Query**	**Recommendation Feature**			
			Similarity	Expansion	Synonym	Voting
Hariyanti et al.	Bag-of-Words	Task Label	Cosine	No	No	No
SWIF	TF-IDF	SSI FR	Cosine	Yes	Yes	Yes

namely cross-domain knowledge expansion, synonym expansion, and voting. In order to emphasize the benefits of the differences, this evaluation hypothesized that SWIF would be able to recommend SSI-specific weaknesses using the additional features with greater accuracy than the existing approach.

In the following subsections, the experimental settings and findings that test the hypothesis are described.

Experimental Settings

A) Dependent Variables. To measure the accuracy of SWIF in identifying SSI-specific weaknesses, four dependent variables are declared: precision, recall, F1-score, and accuracy. First, precision is a metric that indicates the proportion of correct recommendations in relation to the total number of recommendations. The precision metric can be computed as follows:

$$Precision = \frac{|W_{recommend,actual}|}{|W_{recommend}|} \tag{6.1}$$

Where $|W_{recommend,actual}|$ represents the cardinality of the set of recommended and actual weaknesses from SWIF, and $|W_{recommend}|$ represents the cardinality of the set of all recommended weaknesses from SWIF. When the precision score is high, the above equation indicates that SWIF is able to recommend the majority of actual weaknesses relative to all recommended weaknesses. It is ideal to achieve the greatest possible precision.

Second, recall is a metric that represents the proportion of correct recommendations relative to the entire corpus of actual data. The recall metric can be calculated as follows:

$$Recall = \frac{|W_{recommend,actual}|}{|W_{actual}|} \tag{6.2}$$

where $|W_{recommend,actual}|$ denotes the cardinality of the set of actual and recommended weakness from SWIF, and $|W_{actual}|$ denotes the cardinality of the set of actual weaknesses in the corpus. When the recall score is high, the above equation indicates that SWIF can recommend the majority of actual weaknesses in the corpus. It is ideal for SWIF to achieve the highest possible recall score, as this indicates a comprehensive corpus exploration.

The F1 score is the harmonic mean of the precision and recall metrics. This metric normalizes the difference between two metrics, which can be determined as follows:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.3}$$

If the F1 score metric is high, the above equation indicates that SWIF can achieve the high precision and recall metrics. It is ideal to achieve the highest possible F1 score, but in practice, it must be balanced.

The final metric is accuracy, which represents the proportion of all correct recommendations in the corpus. This metric measures the effectiveness of identifying and recommending SSI-specific weaknesses. This metric can be calculated as follows:

$$Accuracy = \frac{|W_{recommend,actual}| + |W_{not\_recommend,actual}|}{|W|} \tag{6.4}$$

where $|W_{recommend,actual}|$ represents the cardinality of the set of actual and recommended weaknesses from SWIF, $|W_{not\_recommend,actual}|$ represents the cardinality of the set of actual weaknesses that are not recommended from SWIF, and $|W|$ represents the cardinality of the set of all weaknesses in the corpus. Likewise, SWIF should ideally achieve the highest possible accuracy score to indicate the performance of weakness identification.

B) Independent Variables. Three controllable independent variables are declared to determine the comparison feature variant. According to Table 6.1, vectorization, cross-domain expansion, and synonym expansion are three distinguishable features that can be used as independent variables.

C) Experimental Procedure. Using the previously declared dependent and independent variables, recommendation systems that imitate the SWIF and existing approach-compliant settings can be implemented. In addition, the configurations are varied according to the combination of independent variables. For instance, the first configuration will implement a recommendation system based on the existing approach, while the second configuration will implement a recommendation system that employs TF-IDF vectorization but no other features.

52 functional requirements are prepared from a typical design of an SSI management system by us as an input for the SWIF implementation in order

Table 6.2: Comparison of experimental results of the recommendation system based on different configurations.

Configuration	**Independent Variable**			**Dependent Variable**			
	Vectorization	Expansion	Voting	Precision	Recall	F1 Score	Accuracy
1 Other [27]	Bag-of-Words	No	No	0.2291	**0.9118**	0.3661	0.3060
2	TF-IDF	No	No	0.2356	0.8824	0.3719	0.3448
3	TF-IDF	Yes	No	0.2618	0.8137	**0.3962**	0.4547
4 SWIF	**TF-IDF**	**Yes**	**Yes**	**0.8667**	0.1275	0.2222	**0.8039**

to identify SSI-specific weaknesses. These SSI functional requirements will serve as an additional controlled variable in the experiment. Different configurations of the recommendation system will be used to identify SSI-specific weaknesses based on the same set of SSI functional requirements.

In each execution of the recommendation system, the outcomes will be recorded and used to calculate precision, recall, F1 score, and accuracy metrics relative to the ground truth, as described in Section 3.6.1. Lastly, configurations are compared to each other using the metrics.

Experimental Results

After executing the recommendation system on various configurations, the experimental results can be obtained as shown in Table 6.2. Four configurations of the recommendation system are listed on the left side of the table, along with three distinguishing features: vectorization, cross-domain expansion, and voting.

The first row represents the configuration that adheres to the existing approach [27], employing the same vectorization and omitting the expansion and voting features. This configuration provides the highest recall score for the recommendation system. The second row represents the configuration in which bag-of-words vectorization is switched to TF-IDF vectorization. This configuration marginally improves precision, F1, and accuracy, while recall is marginally diminished. The third row is the configuration used to introduce the recommendation system's cross-domain expansion. This configuration yields the highest F1 score among the four available configurations. Besides the recall, this configuration also slightly improves the metrics. In the configuration that follows SWIF, cross-domain expansion and voting mechanisms are introduced in the fourth row. This configuration demonstrates the highest precision and accuracy, but it necessitates a trade-off between the recall and F1 scores and the other two. In Section 6.6, these experimental results will be discussed.

6.1.2 Configuration on Weakness Identification

The acceptable threshold of the number of top recommended results (x) and the acceptable threshold of the weight of weakness in voting (y) are two constant variables that must be configured prior to execution in the proposed algorithms in SWIF. Because these thresholds should be set up experimentally in accordance with the weakness dataset, the best or ideal value of these thresholds cannot be fixed in the content.

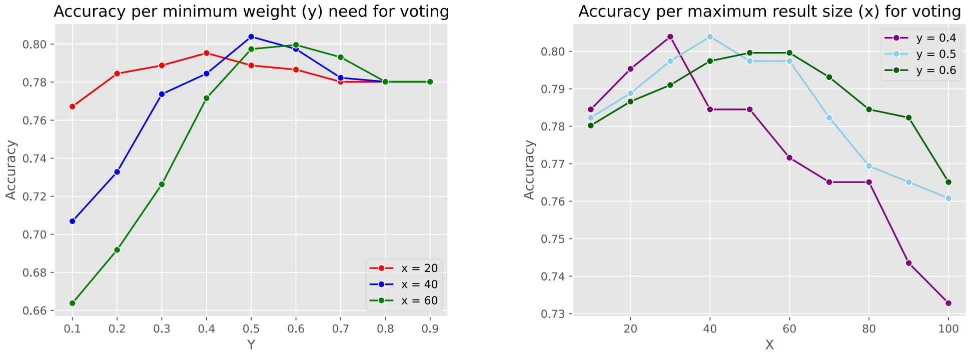
An experiment is conducted in this section to test the hypothesis that the SWIF implementation can recommend SSI-specific weaknesses with the best performance in accordance to the suitable configurations of the underlying thresholds. The experiment uses an accuracy metric (Equation 6.4) to measure how well the thresholds fit the dataset and is designed to test as many thresholds as it can. It can be seen from the semantic of the threshold values that the optimization goal is to maximize the acceptable weight of weakness and minimize of the threshold of the acceptable top recommended results. When the number of acceptable top results is low, it means that the actual SSI-specific weaknesses can be found in a few top recommendation results. On the other hand, when the acceptable weight of weakness is large, it means that the recommended SSI-specific weaknesses are severe and suitably recommended. The settings, procedure, and outcomes of the experiments are described in the following sections.

Experimental Settings

A) Dependent Variable. An accuracy metric is chosen as the the only dependent variable to illustrate how well the retrieval and ranking mechanism can identify SSI-specific weaknesses. The accuracy metric will record and evaluate various threshold configurations.

B) Independent Variable. The threshold of the maximum number of top recommended results and the threshold of the minimum acceptable weight of weakness are declared as the two controllable independent variables that will be used to determine the configuration variants. Each experiment run will include a different combination of these two variables.

C) Experimental Procedure. First, a list of 52 SSI functional requirements from a sample case study of the SSI management system (i.e., the same dataset as used in Section 6.1.1) are prepared. Using the predefined independent variables, the SWIF implementation is executed on the SSI functional requirements against the corpus of common security weaknesses. In each run, the two thresholds are logged and the recommendation's accuracy



(a) Accuracy at different thresholds of minimum weight when the number of top results is fixed to 20, 40, and 60.

(b) Accuracy at different thresholds of maximum top results when the weight of weakness is fixed to 0.4, 0.5, and 0.6.

Figure 6.1: Experimental result in finding the optimal configurations of the thresholds in SWIF.

is calculated by comparing it to the ground truth. The threshold for the maximum number of top-recommended results (x) is first determined with a fixed value, and then multiple thresholds for the minimum weight of weakness (y) are iterated through. The execution is then repeated for as many threshold configurations as possible, and the accuracy metrics are compared.

Experimental Results

After running the experiment on the recommendation system with various configurations, the calculated accuracy scores are collected and a line graph is plotted based on such scores, as depicted in Fig. 6.1.

In the first three rounds, the threshold for the maximum number of top-recommended results (x) is fixed at 20, 40, and 60, respectively, and the threshold for the minimum weight of weakness (y) is varied in increments of 0.1 from 0.1 to 0.9. When the line graph is plotted based on the recorded accuracy scores in Fig. 6.1a, the accuracy will peak at a certain point and then decline. When y is set to 0.4, 0.5, and 0.6 for fixed $x = 20$, 40, and 60, respectively, the peaks are observed. In the first three rounds, the highest possible accuracy is approximately 0.81.

In the next three rounds, the threshold for the minimum weight of weakness (y) will be fixed, while the threshold for the maximum number of top-recommended results (x) will be variable. As discovered in the first three rounds, the most accurate results are obtained when y is between 0.4 and 0.6. Consequently, y is set to 0.4, 0.5, and 0.6 in these following three rounds to achieve the highest level of accuracy. The x is varied for every configuration of y by 10 steps from 10 to 100. When the line graph of their accuracy is plotted in Fig. 6.1b, the same situation of peaking can be recognized. Peaks can be observed when x is set in between 30 to 60 and y is fixed at 0.4, 0.5, and 0.6, respectively. These rounds have a maximum accuracy of 85 percent.

From the graphs, it can be seen that, at some point, some threshold configurations will result in greater accuracy than others. In addition, the graph enables us to optimize threshold configuration. The graph helps summarize that the configuration of $x = 40$ and $y = 0.5$ is optimal because it yields the highest level of accuracy with the desired goals. Section 6.6 will discuss this optimal configuration.

6.1.3 Accuracy from the Knowledge Expansion with Knowledge Graph

The completeness of the SSI-CWE cross-domain transfer knowledge graph that is utilized in the knowledge expansion method also influences the accuracy of the weakness identification with SWIF. The purpose of this section's evaluation is to ensure that the proposed knowledge graph contains sufficient information regarding term interrelationships to support the identification. As a result, a hypothesis is formulated, given that a more comprehensive knowledge graph could support the increased accuracy of weakness identification. A quantitative experiment is conducted to test the hypothesis. In the following sections, experimental settings, procedures, and results are described, in that order.

Experimental Settings

A) Dependent Variable. Since the purpose of this experiment is to compare the various levels of information provided by the knowledge graph, each level should be measured using the same metric. As a result, the dependent variable in this experiment is the accuracy metric, as defined by Equation 6.4.

B) Independent Variable. The group of four distinct knowledge graphs used in the knowledge expansion method serves as an independent variable:

1. A knowledge graph that contains only generic terms from the weakness descriptions and their synonyms.
2. A knowledge graph that contains only 25 percent of facts that relate to generic terms, SSI-specific terms, and their synonyms.
3. A knowledge graph that contains only 25 percent of facts that relate

Table 6.3: Experimental result on the accuracy of weakness identification influenced by the completeness of the knowledge graph.

Knowledge Graph Variation	Graph Component Information			Precision	Accuracy
	#SSI-Specific	#Generic	#Fact		
1. No specific	0	7	20	0.5153	0.6267
2. 25% irrelevant facts	9	17	57	0.5517	0.7198
3. 25% relevant facts	19	12	59	0.6598	0.7866
3. Full knowledge graph	134	84	244	0.6842	0.7913

to generic terms, SSI-specific terms, and their synonyms, but are not crucial for identifying SSI-specific weaknesses.

4. The proposed knowledge graph contains all generic terms from the weakness description, all SSI-specific terms, and their synonyms.

In the experiment, the SWIF implementation used for weakness identification will be fixed to the same configuration, i.e., knowledge expansion and voting mechanisms will be included.

C) Experimental Procedure. In this experiment, 52 SSI functional requirements are applied to the SWIF implementation (identical to the requirements used in Section 6.1.1). The experiment consists of four rounds: First, the generic knowledge graph is utilized in the implementation of SWIF's knowledge expansion method. Second, a knowledge graph containing approximately 25 percent of facts related to SSI-specific terms is utilized. Third, the knowledge graph containing approximately 25 percent of irrelevant facts related to SSI-specific terms is utilized. Finally, the complete SSI-CWE cross-domain transfer knowledge graph is implemented. In each round of the experiment, the accuracy metric is evaluated against the truth. The accuracy metrics are then compared to those of previous rounds.

Experimental Results

The experimental results are collected in terms of the accuracy metric, as shown in Table 6.3.

The experimental results indicate that by incorporating more SSI-specific terms into the proposed knowledge graph, the knowledge expansion could improve the accuracy of weakness identification. In other words, when 25% of facts that relate to SSI-specific terms are utilized, the accuracy increases from 62% to 78% (i.e., first and second variations). In addition, when the proposed knowledge graph is used in its entirety, accuracy increases significantly from 78% to 79%. It is evident that the completeness of the proposed knowledge graph has little impact on the identification of weaknesses. This phenomenon is analyzed, and it is determined that the 19 SSI-specific terms included in the 25 percent version of the knowledge graph could cover terms that relate to the weakness description and have already supported the expansion of knowledge. To prove the relevancy of the second variation, the third variation shows that, with the similar number of facts (25 percent) and irrelevant facts, the knowledge graph does not influence the accuracy in weakness identification reporting approximately 72% accuracy. In fact, the 9 SSI-specific terms in the third variation are more influent in improving the accuracy than the 19 SSI-specific terms in the second variation.

6.2 Performance

6.2.1 Performance on Model Finding

Based on the number of components, data objects, system states, and event signatures, the SSI data sharing model proposed in Chapter 5 should be scalable. In reality, the SSI data sharing model necessitates a significant amount of time for model finding due to the increasing number of the aforementioned signatures. This dissertation assumes that the number of components and data objects will not increase drastically over the model's scope, as they may be fixed at some point in the SSI management system. On the other hand, the number of system states and events increases significantly over time. With these behaviors, the number of such signatures could impact the Alloy Analyzer's execution time for model finding and analysis, and the SSI data sharing model's overall performance.

In this section, an experiment will be conducted to quantitatively evaluate the performance of the SSI data sharing model in terms of execution time per model scope. In the following sections, experimental settings, procedures, and results are described, respectively.

Experimental Settings

A) Dependent Variable. The execution time reported by the Alloy Analyzer is selected as the only expected dependent variable. The execution time is readily discernible from the Alloy Analyzer's output message. The execution times of various signature configurations will be recorded and compared.

B) Independent Variable. In the experiment, there are four controllable independent variables: (1) the number of component signatures, (2) the number of data object signatures, (3) the number of system state signatures, and

(4) the number of event signatures. As stated previously, these variables impact the performance of the model findings derived from the SSI data sharing model in Alloy Analyzer. Three fixed configurations for (1) and (2) are established, while introducing a gradual range for (3) and (4) to accommodate the SSI data sharing model's characteristics.

Moreover, the performance reported from the experiment is scoped based on our workbench, which is a computer with a 3.4GHz quad-core processor, 32GB of 2400MHz DDR4 memory, and 4GD of graphic memory.

C) Experimental Procedure. The typical (non-custom) SSI data sharing model in Alloy is established, as described in Section 5.4. Then, the following `run` command is declared to execute the SSI data sharing model in the Alloy Analyzer as:

```
run {} for exactly X Component, exactly Y DataObject, exactly A State,
        exactly B Event
```

where `X`, `Y`, `A`, and `B` denote parameters indicating the number of the component signatures, the data object signatures, the system state signatures, and the event signatures, respectively. These parameters are adjustable based on the configuration. In this experiment, the checking of information security and privacy properties is omitted because it could take more time and does not show the performance of the SSI data sharing model. The above command is then executed in the Alloy Analyzer and the execution time reported by the tool is recorded.

Experimental Results

After executing the SSI data sharing model in the Alloy Analyzer with various configurations, the execution times can be recorded and used to generate the line graph shown in Fig. 6.2.

On the y-axis of the graph are the execution times in seconds, and on the x-axis are seven configurations of the number of system states (A) and sharing events (B). Additionally, the graph depicts three execution time trends based on the number of components (X) and data objects (Y). The trend indicates that the SSI data sharing model's execution time is increasing as the number of signatures rises. It can be discovered that execution time increases linearly with a small number of components and data objects. On the other hand, the large number of components and data objects exponentially increases the execution time. However, the execution time is acceptable given that the largest configuration is executed in less than seven minutes (or 400 seconds). Section 6.6 will discuss on the performance of the SSI data sharing model.

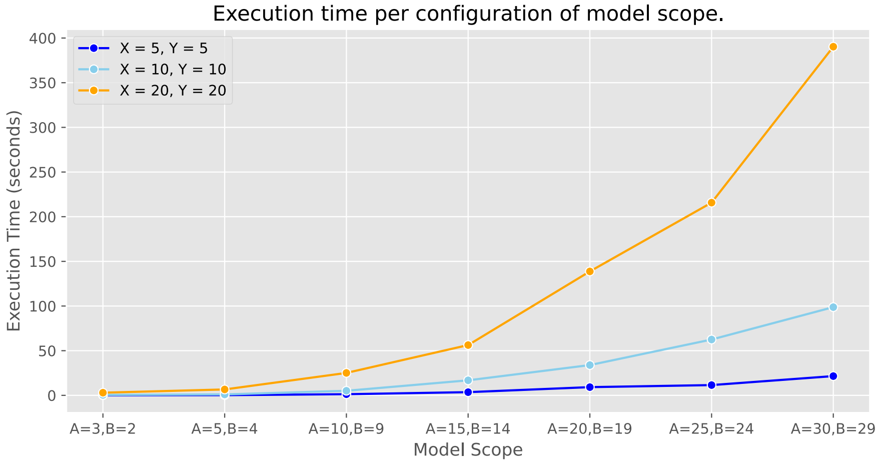


Figure 6.2: Execution time of the Alloy Analyzer per configuration of the components, data objects, system states, and sharing events.

6.3 Validity

6.3.1 Validity of Recommended SSI-Specific Weaknesses

The validity of the recommended SSI-specific weaknesses is another quality aspects of SWIF in which the recommendation provides security weaknesses that are actually based on the functionality of SSI management systems. However, with information retrieval technique proposed in this dissertation, it is difficult to conclude that the recommended security weaknesses are actual based on the concept of SSI management systems.

In this section, an experiment is conducted to qualitatively evaluate the validity of the recommended SSI-specific weaknesses. This dissertation assumes that SSI-specific weaknesses recommended by SWIF are valid if they are direct associated to the core functionality defined by the SSI model. The association can be determined by comparing the text description of the weakness to the core functionality of the SSI model. To facilitate the determination, three conditions are established to determine and justify the association between the description of the recommended weaknesses and the core functionality of the SSI model as follows:

1. A weakness is associated to the SSI model if the subject or target component mentioned in the description of weakness can be comparable to or replaceable with the typical component in the SSI model.

2. A weakness is associated to the SSI model if the functionality or operation constrained by the weakness can be comparable to or replaceable with specific functions or operations in the SSI model.
3. A weakness is associated to the SSI model if the data object related to the constraint in the weakness can be comparable to or replaceable with specific data objects or identity data in the SSI model.

These conditions may help to justify the recommended SSI-specific weaknesses. Nevertheless, this dissertation contends that these conditions differ from the justification in the ground truth because domain knowledge of the SSI model or SSI management systems is considered. Based on the conditions, each of the recommended results can be evaluated manually.

The SWIF implementation is first executed on the same set of 52 functional requirements (as used in Section 6.1.1) to recommend SSI-specific weaknesses. Out of 464 weaknesses from the corpus, the SWIF implementation recommends 15 SSI-specific weaknesses that passes the retrieval and ranking mechanisms. Then, the authors act as an evaluator to justify whether the description of the recommended weaknesses possesses any direct association to the core functionality of the SSI model. In order to avoid bias, the determination of valid SSI-specific weaknesses is performed with a clear justification. The evaluation results of the 15 SSI-specific weaknesses is summarized in Table 6.4.

Direction associations is concluded if all the three conditions have been satisfied by the recommended SSI-specific weaknesses. The table reports that 10 of the 15 recommended SSI-specific weaknesses ($\sim$ 67 percent) have direct associations with the core functionality of the SSI model. To support the determination on the direct association, justifications on which parts are associated with each other and why they are concluded as associated are provided. For instance, the "CWE-200: Exposure of sensitive information to an unauthorized actor" is concluded as having a direct association because "As a system that manipulates personally identifiable information, SSI management system may expose such information to unauthorized actors due to various reasons, including the combination of other weaknesses. For example, a SSI-specific weakness is occurred when: the dishonest issuer participates in the SSI management system can access to and make use of a holder's PII while they validate the claim if the zero-knowledge proving mechanism are not correctly implemented. The holder is required to trust the issuer when the claim validation occurs." Another prominent example of the weakness with the direct association is the "'CWE-1249: Application-level admin tool with inconsistent view of underlying operating system", which is concluded as not having a direct association because "SSI management systems are des-

Table 6.4: Evaluation results of the SSI-specific weaknesses against the predefined conditions.

Recommended Weakness	Satisfied Condition	Recommended Weakness	Satisfied Condition
CWE-200: Exposure of sensitive information to an unauthorized actor.	**1, 2, 3**	CWE-640: Weak password recovery mechanism for forgotten password	**1, 2, 3**
CWE-202: Exposure of sensitive information through data queries	3	CWE-645: Overly restrictive account lockout mechanism	**1, 2, 3**
CWE-213: Exposure of sensitive information due to incompatible policies	3	CWE-651: Exposure of WSDL file containing sensitive information	1, 3
CWE-214: Invocation of process using visible sensitive information	1, 3	CWE-862: Missing authorization	**1, 2, 3**
CWE-226: Sensitive information in resource not removed before reuse	**1, 2, 3**	CWE-863: Incorrect authorization	**1, 2, 3**
CWE-285: Improper authorization	**1, 2, 3**	CWE-922: Insecure storage of sensitive information	**1, 2, 3**
CWE-521: Weak password requirements	**1, 2, 3**	CWE-1249: Application-level admin tool with inconsistent view of underlying operating system	-
CWE-639: Authorization bypass through user-controlled key	**1, 2, 3**		

Note: Justification of each evaluation results against the conditions are provided in https://doi.org/10.5281/zenodo.6951404.

ignated to be distributed systems that avoid the control of central authority. In SSI management systems, admin tools are less likely to exist. Especially for the identity wallet components installed on the holder's device to manage identity data, admin tool cannot be used. Therefore, all the three conditions do not satisfy in the core functionality of the SSI model." The validity of the recommended SSI-specific weakness will be further discussed in Section 6.6.

6.3.2 Validity of the SSI-CWE Cross-Domain Transfer Knowledge Graph

The utilization of the SSI-CWE cross-domain transfer knowledge graph to influence the identification of weaknesses for the SSI management system is an additional important outcome of this dissertation. This knowledge graph was developed manually based on the knowledge of two domains: the SSI model and the CWE common weakness database. The development should result in a valid knowledge graph that meets quality requirements.

Chen et al. [97] conducted a survey and derived 18 quality requirements from the literature in order to determine the quality perspectives for the knowledge graph. This dissertation assumes that the proposed knowledge graph is valid if it satisfies the quality requirements [97].

In this section, an experiment is conducted with experts to evaluate the quality of the proposed knowledge graph. The experiment enlisted a group of knowledgeable experts to evaluate the proposed knowledge graph based on quality requirements. The experiment tests the hypothesis that the proposed knowledge graph can meet the maximum number of quality requirements. The subsequent subsections will describe the experimental settings, procedures, participant qualifications, and results, respectively.

Experimental Settings

A) Dependent Variable. This experiment establishes a dependent variable for determining the satisfaction of quality requirements. Each quality requirement is manually categorized as one of the following:

- *Satisfied* indicates the quality requirement is explicitly satisfied by the proposed knowledge graph.
- *Unsatisfied* indicates the quality requirement is explicitly not satisfied by the proposed knowledge graph.
- *Indecisive* indicates the participant cannot decide whether the quality requirement is satisfied.

The summary of the participants' responses will be expressed as the proportion of participants who selected the category of quality requirements.

B) Independent Variables. Participants and the 18 quality requirements for knowledge graph determined by Chen et al. [97] are treated as independent variables in this experiment. Each requirement will vary to ask the decisions of the participants. The chosen quality requirements are appropriate for evaluating the proposed knowledge because they were derived from numerous research works that proposed and evaluated the knowledge graph. In addition, the chosen quality requirements encompass multiple aspects, such as completeness, relevance, and trustworthiness. Fig. 6.3 depicts the covered quality aspects of the selected requirements.

C) Participant Qualification. In this experiment, human subjects, referred to as participants, are expected to be able to evaluate and justify whether the proposed knowledge graph satisfies the requirements. Consequently, this experiment assumes that participants possess a graduate degree and a fundamental comprehension of the graph-based data model. Participants with a graduate degree are qualified to conduct research and evaluate the research outcomes due to their critical thinking skills. In contrast, the participants' understanding of the graph-based data model qualifies them to grasp the

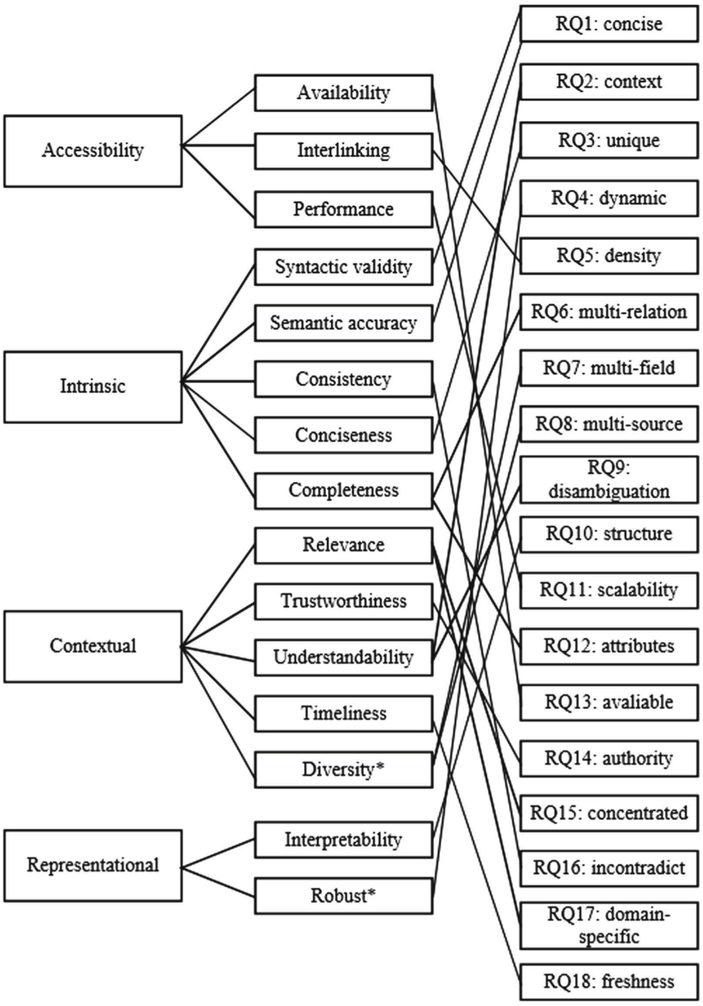


Figure 6.3: Mapping the requirements to quality aspects [97].

development context and comprehend the proposed knowledge graph's content. This dissertation concludes that these requirements are both minimal and sufficient for participants to provide substantial feedback on the proposed knowledge graph.

D) Experimental Procedure. The experiment consisted of six steps and was conducted through an online platform:

1. A web-based questionnaire form[1] is created to provide knowledge on the terminologies used in the SSI management system, the CWE weakness description, and the basic definition of the proposed knowledge.
2. Six qualified participants have been invited to participate in the questionnaire response.
3. Each participant is expected to read the SSI management system's basic terminology to familiarize themselves with the SSI-specific terms used in the proposed knowledge graph.
4. Each participant is then expected to read the basic definition and terminology used to describe common security weaknesses in the CWE database in order to familiarize themselves with the generic terminology.
5. Each participant is then required to read the knowledge graph's introduction, how nodes and relations are constructed, and an example of the proposed knowledge graph. Participants are encouraged to ask the authors as many questions as necessary to ensure they comprehend the evaluation objective's context. e evaluation target.
6. On the basis of the three categories of satisfied, unsatisfied, and indecisive, each participant must determine whether the proposed knowledge graph can meet the 18 quality requirements. The participants may provide additional comments or feedback if they so choose.

The percentage of the participants' decisions is collected and summarized in order to determine the level of requirement satisfaction.

Experimental Results

From the six distributions of the online questionnaire, five responses have been collected (a response rate of 83.3%). Table 6.5 provides an overview of the response results.

The table demonstrates that the majority of participants believe the proposed knowledge graph satisfies 16 out of 18 quality requirements (approximately 89 percent). There is unanimity (100 percent of participants agreed)

[1] The online Google forms: https://forms.gle/SeusGWfHX6mYB1vz5.

Table 6.5: Summary of the decision on the quality requirements from the participants.

Quality Requirement	**Participant Decision**		
	Satisfied	**Indecisive**	**Unsatisfied**
RQ1. Knowledge triples are concise.	**100%**	-	-
RQ2. Contextual information of entities can be explicitly captured.	**100%**	-	-
RQ3. Knowledge graph does not contain redundant triples.	**40%**	**40%**	20%
RQ4. Knowledge graph can be updated dynamically.	**100%**	-	-
RQ5. Entities should be densely connected.	**60%**	40%	-
RQ6. Relations among different types of entities should be included.	20%	**80%**	-
RQ7. Data source are multi-field.	**60%**	40%	-
RQ8. Data for constructing a knowledge graph should in different types and from different resources.	**60%**	40%	-
RQ9. Synonyms should be mapped and ambiguities should be eliminated to ensure reconcilable expressions.	**100%**	-	-
RQ10. Knowledge graph should be organized in structured triples for easily processed by machine.	**80%**	20%	-
RQ11. The scalability with respect to the knowledge graph size.	**80%**	20%	-
RQ12. The attributes of the entities should not be missed.	**60%**	40%	-
RQ13. Knowledge graph should be publicly available and proprietary.	**60%**	40%	-
RQ14. Knowledge graph should be authority,	**60%**	40%	-
RQ15. Knowledge graph should be concentrated.	**80%**	20%	-
RQ16. The triples should not contradict with each other.	40%	**60%**	-
RQ17. For domain specific tasks, the knowledge graph should be related to that field.	**60%**	20%	20%
RQ18. Knowledge graph should contain the latest resources to guarantee freshness.	**60%**	20%	20%

on the satisfaction of three quality requirements. First, participants agreed that the proposed knowledge graph adequately represented the knowledge of both domains. Second, participants agreed that the knowledge graph could successfully capture contextual information. Participants agreed that the inclusion of synonyms in the knowledge graph was successful.

The experiment also revealed that the participants are indecisive about two quality requirements: the majority of participants (80%) believe that relations between different types of entities may not be included, and the majority of participants (60%) also believe that quality requirements may contradict each other. After an oral discussion, the participants justify that they observe terms in nodes with the same textual type and that some facts appear contradictory, e.g., the fact ("software program", synonym_of, "software") indicates that the term "software" is contained within the term "software program" makes the two terms identical and the relation of synonym is contradictory. This dissertation concludes, based on its justifications, that these indecisive satisfactions are not crucial.

6.3.3 Degree of the Consistency of SSI System Properties in CSSPS

In Chapter 4, SSI system properties are improved by complying the existing ones to shared controls from credible source documents and by incorporating the missing endorsed tasks into SSI system property definitions. The improvement conducted is valid if it can achieve the higher degree of consistency in comparison to the existing SSI system properties. Consequently, the quality aspect of this improvement is the degree to which it can be more consistent with the credible source documents. A quantitative evaluation of the degree of consistency between the improved SSI system properties and shared controls is conducted. This evaluation hypothesized that SSI system properties in CSSPS are more compliant than SSI system properties in their original proposal. An experiment is conducted to quantitatively compare the number of controls that are consistent with the particular set of SSI system properties and test the hypothesis. In the following sections, the experimental settings, procedures, and results are described in detail.

Experimental Settings

A) Dependent Variable. Three dependent variables are established: the number of controls in a given source document that are consistent with any SSI system properties, the percentage of consistency, and the difference in consistency between two sets of given SSI system properties. First, the number of security or privacy controls from the given source document that are consistent with a specific set of SSI system properties will be manually justified. These justifications can be completed manually by a qualified analyst, as they consist solely of direct textual comparisons. The number of consistent controls can be expressed as follows:

$$XC_{\mathcal{P},\mathcal{C}} = \text{the number of controls in } \mathcal{C} \text{ that are consistent with } \mathcal{P}. \tag{6.5}$$

where $\mathcal{P}$ denotes an identifier of the given set of SSI system properties, $\mathcal{C}$ denotes an identifier of the given source document indicating a collection of controls, and $XC_{\mathcal{P},\mathcal{C}}$ represents the number of controls in $\mathcal{C}$ with all endorsed tasks that are consistent with the given set of SSI system properties ($\mathcal{P}$).

Second, the percentage of consistency calculates a normalized degree of consistency using the number of consistent controls ($XC_{\mathcal{P},\mathcal{C}}$). The percentage of consistency can be calculated as follows:

$$PC_{\mathcal{P},\mathcal{C}} = \frac{XC_{\mathcal{P},\mathcal{C}}}{|\mathcal{C}|} \tag{6.6}$$

where $PC_{\mathcal{P},\mathcal{C}}$ denotes the percentage of consistency between the given source document ($\mathcal{C}$) and the particular set of SSI system properties ($\mathcal{P}$), $XC_{\mathcal{P},\mathcal{C}}$ denotes the number of consistent controls as in Equation 6.5, and $|\mathcal{C}|$ denotes the cardinality of the set of controls in the given source document ($\mathcal{C}$). It is anticipated that the percentage of consistency will be as high as possible.

Finally, the difference of consistency serves as a comparative indicator to demonstrate the difference between two percentages of consistency. The difference of consistency can be calculated as follows:

$$\Delta_{\mathcal{P}_1,\mathcal{P}_2} = PC_{\mathcal{P}_1,\mathcal{C}} - PC_{\mathcal{P}_2,\mathcal{C}} \tag{6.7}$$

where $\Delta_{\mathcal{P}_1,\mathcal{P}_2}$ denotes the difference of consistency between two sets of SSI system properties ($\mathcal{P}_1$, and $\mathcal{P}_2$), and $PC_{\mathcal{P},\mathcal{C}}$ denotes the percentage of consistency according to Equation 6.6. The range of the difference is between -100 and +100. Positive and negative values indicate that the first set of SSI system properties is more or less compliant with the given source document ($\mathcal{C}$).

B) Independent Variable. In the experiment, two independent variables are established, which are: the set of SSI system properties, and the set of source documents.

Initially, this experiment intended to evaluate CSSPS and report on its improvement to the existing SSI system properties. Therefore, the variable $\mathcal{P}_1$ is established to denote the existing set of SSI system properties consolidated in Section 4.2.1, and the variable $\mathcal{P}_2$ is established to denote CSSPS.

In terms of the SSI system property definition, the consolidated set of SSI system properties should be an excellent reflection of the current state.

Second, the variable for the source documents are limited to two groups: (1) source documents that were used in the improvements (such as GDPR [13] and OWASP Application Security Verification Standard [94]) and (2) source documents that were excluded from the systematic review due to unfulfilled conditions. The source documents utilized in the improvement will reveal the degree to which the improvement can increase consistency. In contrast, the excluded source documents will emphasize the extent to which CSSPS can be applied universally. HIPAA [67] was selected as one of the excluded source documents because it is domain-specific and should apply to SSI management systems implemented in healthcare applications. The Personal Data Protection Act (PDPA) of Thailand [98] was also selected as a source document that was excluded because it was suspended at the time of the improvement, but will resume in June 2023. After this act is enacted, it will meet all of the selection criteria. These two source documents are excellent examples of source documents outside the scope of the improvement.

With these two independent variables, a qualified analyst will be able to evaluate each dependent variable appropriately.

C) Participant Qualification. As previously stated, this experiment involves a qualified analyst who can provide input on the degree of consistency to prevent author bias. The task that we expect the analyst to support is determining whether the controls in the given collection have any consistent SSI system properties. Ability to read and compare technical statements and provide substantial justification is required. Therefore, the analyst must be a system or software analyst with a minimum of three years of experience in software development and security awareness training. This research believes that the analyst's ability to analyze system or software requirements could be sufficient for comparing SSI system properties to controls, and security awareness training can assist the analyst in becoming familiar with the security and privacy controls.

D) Experimental Procedure. During the experiment session with the qualified analyst, $XC_{\mathcal{P},\mathcal{C}}$ should be obtained for all four pairs of source documents and two sets of SSI system properties. Prior to the session, the analyst is requested to read and become familiar with the fundamental concepts and functions of the SSI management system, and the analyst is encouraged to ask as many questions as possible to gain a solid understanding of the SSI system properties.

Table 6.6: Experimental results on the degree of consistency between the existing SSI system property set and CSSPS.

Source Document	$\lvert\mathcal{C}\rvert$	**Existing Property $\mathcal{P}_1$**		**CSSPS $\mathcal{P}_2$**		$\Delta_{\mathcal{P}_2,\mathcal{P}_1}$
		$XC_{\mathcal{P}_1}$	$PC_{\mathcal{P}_1}$	$XC_{\mathcal{P}_2}$	$PC_{\mathcal{P}_2}$	
GDPR [13]	7	4	57.14%	6	85.71%	+28.57%
OWASP Application Security Verification Standard [94]	14	8	57.14%	13	92.86%	+35.72%
HIPAA [67]	17	10	58.82%	12	70.59%	+11.77%
PDPA of Thailand [98]	14	11	78.57%	13	92.86%	+14.29%

Note: The justifications from the analyst is published at https://doi.org/10.5281/zenodo.6951404.

The experiment was conducted in two rounds. First, the analyst is provided with four lists of controls extracted from the four selected source documents, as well as a list of SSI system property definitions extracted from the consolidated list ($\mathcal{P}_1$). The analyst is inquired as to how many controls in each list have SSI system properties that can be defended as consistent. All of the analyst's decisions, justifications, and comments are documented. Then, in the second round, the analyst is requested to repeat the same tasks using SSI system properties in CSSPS ($\mathcal{P}_2$). In this round, all decisions, justifications, and comments are also documented. After the experimental session, all dependent variables (Equations 6.5, 6.6, and 6.7) are calculated based on the analyst's decisions.

Experimental Results

After conducting the experiment with a qualified analyst, the decisions are documented and all corresponding variables are calculated. Table 6.6 compares and reports recorded dependent variables to two groups of independent variables.

The first column indicates the source document against which each set of SSI system properties is evaluated. The second column displays the number of security or privacy controls present in the source document. The third and fourth columns represent the number of consistent controls and the percentage of consistency with respect to the existing set of SSI system properties, respectively. Similar to the third and fourth columns, the fifth and sixth columns are evaluated using CSSPS. The final column indicates the difference in consistency between the existing set and CSSPS.

The results demonstrate that the improvement to CSSPS increases the consistency of the existing SSI system properties. For example, the existing set of SSI system properties is consistent with 57.14 percent of GDPR controls, but CSSPS could increase consistency by 28.57 percent, resulting in

85.71 percent. It can also be observed that the overall trend for both used and excluded source documents is improving. In Section 6.6, the benefits and limitations of the increased degree of consistency will then be discussed.

6.4 Applicability

6.4.1 Applicability of SSI System Properties in CSSPS

The applicability of the SSI system properties in CSSPS to real-world situations or actual SSI management system implementation is an essential quality aspect. The applicability of SSI system properties can generally be expressed in two dimensions: applicability from existing SSI management systems and applicability to constrain the development of new SSI management systems. Nonetheless, it is difficult to find a new SSI management system at the time of this research. this dissertation choses to assess the applicability of SSI system properties in CSSPS as an asset in existing SSI management systems.

To evaluate the presence of SSI system properties in existing SSI management systems, it is necessary to have a firm grasp of the system's functionality and underlying concept. As a result, an experiment is conducted to evaluate the applicability qualitatively. This experiment hypothesized that the applicability of SSI system properties in CSSPS would be high if they were applicable to and possessed by as many existing SSI management systems as possible. To test the hypothesis, the experimental settings and procedures are devised as described in the following sections.

Experimental Settings

A) Dependent Variable. As this experiment anticipated to determine the possession of SSI system properties, a dependent variable is established to indicate the possession flag. Typically, a flag as "yes" or "no" can be used to indicate possession, but in some SSI management systems, the lack of accessible information prevents the experiment from determining the possession of SSI system properties. As a result, the dependent variable will be assigned the following three possession flags.

- A "*yes*" flag indicates that whether the target SSI management system possesses the specified SSI system property *can be justified.*
- An "*unclear*" flag indicates that whether the target SSI management system possesses the specified SSI system property *cannot be justified*, but it *may possible* based on the capabilities and internal mechanisms.

- A "*no*" flag indicates that whether the target SSI management system possesses the specified SSI system property *cannot be justified*.

This dependent variable will be assigned to every SSI system property for each real-world SSI management system.

B) Independent Variable. Since the dependent variable will vary depending on the specified SSI system property and the target SSI management system, we define two independent variables: the SSI system properties in CSSPS and the real-world SSI management systems.

The independent variable for SSI system properties varies across all 42 SSI system properties contained in CSSPS. In contrast, the independent variable for the real-world SSI management systems will vary between three industrial-level SSI management systems, namely uPort, Sovrin, and IBM Verify Credentials, in order to cover the vast real-world landscape. These SSI management systems were chosen based on their size, functionality, and marketing goals.

uPort is an SSI management system, which is a standalone product that does not adhere to the SSI model's concept, including the DID and verifiable credential standards. Numerous articles [99, 100] have referenced uPort as an alternative SSI management system design example. Currently, uPort is being phased out and replaced by another product called Veramo, but there are no significant differences between Veramo and other brands in terms of features. uPort is still chosen throughout this evaluation. The white paper [35] is used as a source of information to illustrate the functionality of the uPort.

Sovrin is an SSI governance network that offers reliable block validators who are well-known network stewards. Sovrin provides implementors with interfaces to connect their applications. In addition, Sovrin recommends an edge-cloud agent architecture that is compatible with the Sovrin network connection. Sovrin is selected as a representative of a large-scale identity network, which enables an SSI management system architecture with strict governance. The white paper [101] will be used as a source of information to illustrate the governance requirements and the suggested system architecture.

IBM Verify Credentials (IBM-VC) is a comprehensive solution for the IBM Corporation's SSI management system. IBM-VC utilizes Hyperledger Indy as a distributed ledger technology. This solution offers a comprehensive SSI management system that adheres to open standards and adopts the edge-cloud architecture proposed by Sovrin. We selected IBM-VC because it is an excellent example of full-featured SSI management systems and is widely used around the world. We will use the IBM-VC website's online documentation [36] as a source of information to illustrate its functionality and architecture.

C) Experimental Procedure. The key to evaluating the applicability of SSI system properties in the specified SSI management system is to comprehend the system's design and functionality in order to determine if the quality aspects defined in SSI system properties are controlled. First, the documentations for the three SSI management systems are carefully examined. Then, for each SSI system property in CSSPS, the constraint with the target SSI management system's design and functionality are compared by the authors to determine which flag of possession should be assigned to the property and justify the rationales behind the assignment.

Experimental Results

As shown in Table 6.7, a flag of possession is assigned for each individual SSI system property in CSSPS in correspondence with the target system. In Table 6.8, an example of the justification for assigning flags of possession of an SSI system property to the three SSI management systems is also provided.

According to the evaluation results, 11 SSI system properties in CSSPS have Yes flags assigned to them in every SSI management system. This indicates that these SSI system properties are interested and already possessed by existing SSI management systems. Following that, 23 SSI system properties that are flagged with a combination of Yes and Unclear are identified. This circumstance suggests that the existing SSI management systems are concerned with the SSI system properties and will be able to fully incorporate them in the future if they have not already been possessed. Six SSI system properties that have No flags assigned to them in some SSI management systems are other interesting results. It refers to the fact that the technology or design employed by the existing SSI management systems has resulted in certain limitations. This dissertation recognized that these limitations can be overcome by new SSI management system's design alternatives that are intended to be compliant with the SSI system properties from the beginning.

As demonstrated in Table 6.8, a justification for assigning possession flags to the "data recovery" SSI system properties is provided. This property merits attention because the three SSI management systems have assigned Yes, No, and Unclear flags to it. uPort is the only SSI management system that provides a clear mechanism for recovering the identity wallet via the recovery contact. Sovrin proposes the edge-cloud architecture, which is intended for locally storing identity data in an edge agent that may be lost, stolen, destroyed, or tampered with. However, it is the responsibility of implementers

Table 6.7: Evaluation results of the applicability of SSI system properties in real-world SSI management systems.

Property Name	**Flag of Possession**		
	uPort	**Sovrin**	**IBM-Vc**
IP1. Existence	Yes	Yes	Yes
IP2. Sovereignty	Yes	Yes	Yes
IP3. Single Source	No	Yes	Yes
IP4. Standard	Yes	Yes	Yes
IP5. Cost Free	Yes	Yes	Yes
IP6. Decentralized	Yes	Yes	Yes
IP7. Verifiability	Unclear	Yes	Yes
IP8. Scalability	Yes	Yes	Yes
IP9. Accessibility	Yes	Unclear	Unclear
IP10. Sustainability	No	Yes	Unclear
IP11. Access Control	Unclear	Unclear	Unclear
IP12. Transparency	Yes	Yes	Yes
IP13. Persistence	Unclear	Unclear	Unclear
IP14. Portability	Yes	Yes	Yes
IP15. Interoperability	Unclear	Yes	Yes
IP16. Consent	Unclear	Unclear	No
IP17. Data Minimization	No	Yes	No
IP18. Protection	Unclear	Unclear	Unclear
IP19. Data Authentication	Yes	Yes	Yes
IP20. Physical Authentication and Protection	Unclear	Unclear	Unclear
IP21. Data Confidentiality	Yes	Unclear	Unclear
IP22. Data Recovery	Yes	Unclear	No
IP23. Data Availability	Yes	Yes	Yes
IP24. Communication Security	Unclear	Yes	Yes
IP25. Data Integrity	Yes	Yes	Yes
IP26. Data Authorization	Unclear	Yes	Unclear
IP27. Accountability	Unclear	Unclear	Yes
IP28. Non-Repudiation	Unclear	Yes	Unclear
IP29. Data Validation and Sanitization	Unclear	Unclear	Unclear
IP30. Error Handling	Unclear	Unclear	Unclear
IP31. Key Protection	Yes	Unclear	Yes
IP32. Malware Protection	Unclear	Unclear	Unclear
IP33. Password Security	Unclear	Unclear	Unclear
IP34. Configuration Security	Unclear	Unclear	Unclear
IP35. Session Security	Unclear	Unclear	Unclear
IP36. Data Classification	No	No	No
IP37. Fairness and Lawfulness	Unclear	Yes	Yes
IP38. Purpose Specification and Limitation	No	Unclear	Unclear
IP39. Use and Disclosure Limitation	No	Unclear	Unclear
IP40. Data Accuracy and Quality	Unclear	Unclear	Unclear
IP41. Notification	Yes	Unclear	Unclear
IP42. Individual Participation	Unclear	Unclear	Unclear

Note: The full justifications on the decisions of the possession flags are provided publicly at https://doi.org/10.5281/zenodo.6951404.

Table 6.8: Example of the justification for assigning flags of possession of the data recovery SSI system property.

Justification for Flags of Possession		
uPort	**Sovrin**	**IBM-VC**
IP22. Data Recovery		
Yes - uPort provides a way for recovering a wallet account via friend contacts, and the identity data is stored on IPFS and can be restored to another wallet where the user can verify their integrity. We believed that this property is possessed by uPort.	*Unclear* - Sovrin suggests utilizing their edge agents as identity wallets to store identity data. However, based on the documentation, we cannot specify a mechanism to recover the identity data or store on an artifact that can be lost, stolen, destroyed, or falsified. We considered this property unclear to be possessed.	*No* - The documentation of IBM-VC does not mention any mechanism for recovering identity data. With its design, identity wallets are installed as mobile applications that can be lost, stolen, destroyed, or falsified. The design allows only the re-issuance of credentials and we cannot ensure that such mechanism can be implemented in the applications. We considered that this property is not possessed by IBM-VC.

to create an application that provides the data recovery mechanism and safeguards against unexpected events. The IBM-VC is designed to utilize the identity wallet solely for credential storage and only permits the re-issuance of credentials. It can be determined that the IBM-VC lacks the capability to recover the agent. The applicability of SSI system properties in CSSPS will be further discussed in Section 6.6.

6.5 Case Study

In the preceding activities, each component of the proposed approach is evaluated individually. To ensure applicability, the proposed approach must also be evaluated holistically. In this section, a case study will be used to test the applicability of the proposed approach to a real-world scenario and the interoperability of each part. The following sections provide an overview of the case study, the procedure for applying the proposed approach, and the results.

6.5.1 Overview of the Case Study

To simulate the development of an SSI management system in a domain application, the case study is designed to develop a subsystem of a driver's license system based on the SSI model's functionality. The subsystem will be known as the Driving Information Verification Application (DIVA) and

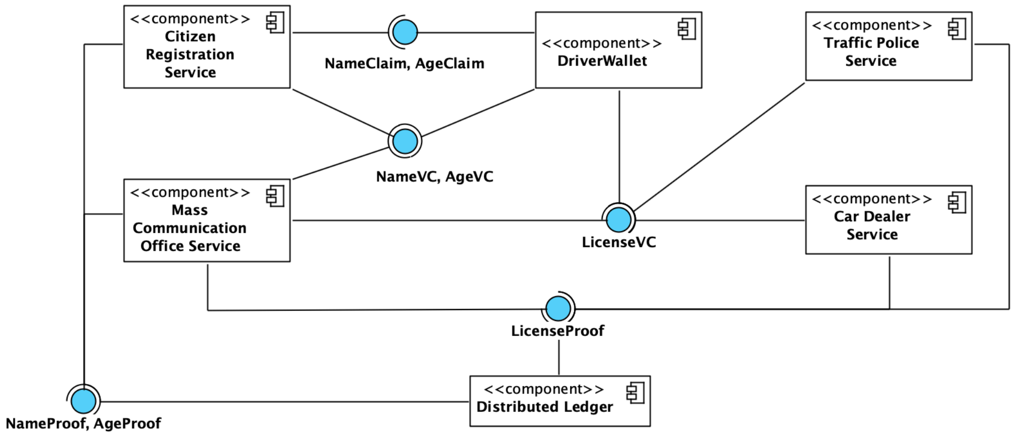


Figure 6.4: Component diagram showing the architectural design of the DIVA, especially the data sharing among components.

will have the same capabilities as a physical driving license for issuing and verifying driving licenses. The DIVA provides drivers with a mobile application that serves as an identity wallet for storing personal information and driving licenses. The driver must contact the office of mass communication to verify their information and driving test scores and to request a license. In addition, the driver must request their personal information from the office of citizen registration and provide proof when the license is issued. The issued license should be electronically stored in the driver's wallet. The DIVA must also permit traffic police officers to validate the validity of driver's licenses using the public schema on the blockchain. The case study will concentrate on this subsystem.

6.5.2 Procedure for the Case Study

In this section, the procedure to apply the proposed approach in order to analyze security weaknesses and privacy preservation in the DIVA is described in detail. Assume the authors are system analysts who are designing DIVA and wish to guarantee its security and privacy for widespread use. The procedure is designed to follow the typical integration process described in Section 5.5.1 and consists of the seven steps listed below.

1. The authors attempted to design the DIVA by preparing a component diagram of its architectural design (Fig. 6.4) manually.
2. The authors manually extract a set of functional requirements from the overview and architectural design. The objective of the elicitation

Table 6.9: Examples of SSI functional requirements for the DIVA.

ID	SSI Functional Requirement
R1.	Drivers shall register their wallet account in the DIVA mobile application using their name, age, home address, and email address.
...	...
R12.	Password must have at least eight characters, have at least one special character, and have at least one number.
R13.	DIVA wallet shall display a dashboard as a landing page after signing-in, providing the driver's profile picture, name, email address, the number of identity attributes, the number of identity claims, and the number of verifiable claims.
...	...
R30.	The verifier shall be able to verify the driving license verifiable claim by retrieving the corresponding license schema from the distributed ledger.

is to collect the necessary functions for the target DIVA. This step yields thirty SSI functional requirements, some of which are shown in Table 6.9.

3. The SWIF implementation receives the prepared collection of SSI functional requirements as an XML file (similar to Fig. 3.11). The SWIF implementation identified a list of SSI-specific weaknesses with language correlations to the specified requirements.
4. The authors analyzed the identified SSI-specific weaknesses and attempted to mitigate them by revising the architectural design and SSI's functional requirements.
5. The authors use the updated design and SSI functional requirements to manually model an instance of the SSI data sharing events as a state transition system and encode it in Alloy.
6. The Alloy encoding is provided to the Alloy Analyzer tool for automatic analysis and model finding. The tool is tasked with determining whether the SSI data sharing model meets the privacy and security specifications. If the violation is discovered, the authors will evaluate the design and make any necessary changes.

Using the procedure outlined above, this evaluation can be made from the case study's results.

6.5.3 Result of the Case Study

On the basis of the procedure outlined in the preceding section, a number of intermediate results are collected and analyzed. This section will describe the results and their analysis in detail.

With the preparation of the input, the first and second steps yield an architectural design (Fig. 6.4) and a collection of SSI functional require-

Table 6.10: Recommended SSI-specific weaknesses for the DIVA.

ID	Recommended SSI-Specific Weakness Title
103	Struts: Incomplete validate() Method Definition
182	Collapse of Data into Unsafe Value
200	Exposure of Sensitive Information to an Unauthorized Actor
250	Execution with Unnecessary Privileges
290	Authentication Bypass by Spoofing
291	Reliance on IP Address for Authentication
295	Improper Certificate Validation
299	Improper Check for Certificate Revocation
349	Acceptance of Extraneous Untrusted Data With Trusted Data
404	Improper Resource Shutdown or Release

ments (Table 6.9). After step 3, the SWIF implementation automatically identifies *ten* SSI-specific weaknesses in relation to the specified SSI functional requirements. Table 6.10 displays the identifiers and names of the identified SSI-specific weaknesses.

On the basis of the identified weaknesses, it is possible to conclude that certain SDLA capabilities may be vulnerable to attack. For example, CWE-103 recommends that the validated method of issuers be exhaustive. Due to the technology-specific nature of the weakness, it may not apply to the DIVA. The CWE-200 weakness is an additional example of this weakness. By manually analyzing the description, it is possible to determine whether the citizen registration service may disclose the name claim or age claim to other sub-services. This could be considered an unauthorized access to information. To address this weakness, the authors suggest that SSI functional requirements must include more information regarding the use of such claims in the citizen registration service, and that sub-services must be incorporated into the architectural design. Therefore, the authors decided to update the collection of SSI functional requirements and architectural design by adding sub-service component functionality. In this instance, new components are added to the design for the citizen information database. The same is true for the mass communication office service and the database of driving test scores. As shown in Fig. 6.5, the revised architectural design can be updated based on the findings of this analysis.

In the fifth step, the updated architectural design is utilized to further analyze the security and privacy specifications of the SSI data sharing events. In order to do this, a typical instance of the data sharing events defined by the design will be manually transformed into a state transition diagram, as depicted in Fig. 6.6. The transformation helps to ensure that the architectural design is able to be represented as the SSI data sharing model. Then,

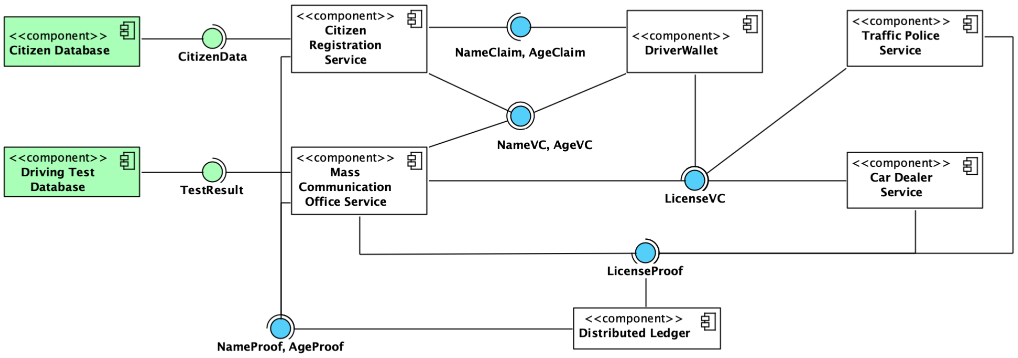
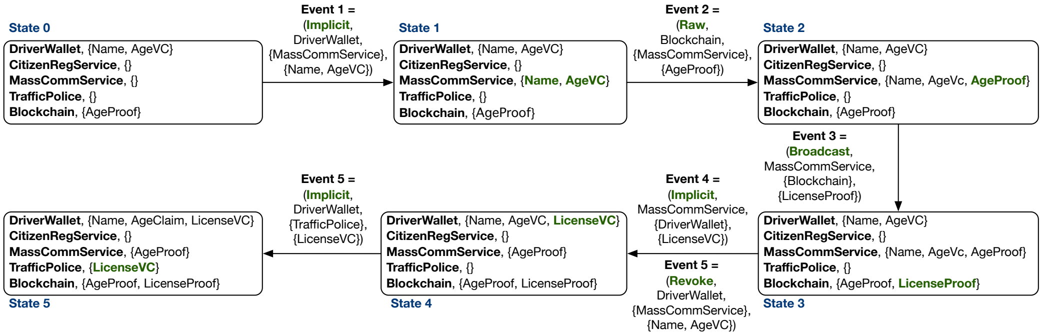


Figure 6.5: Example of the updated component diagram of the DIVA after mitigating SSI-specific weaknesses.



State 0
DriverWallet, {Name, AgeVC}
CitizenRegService, {}
MassCommService, {}
TrafficPolice, {}
Blockchain, {AgeProof}
Event 1 = (Implicit, DriverWallet, {MassCommService}, {Name, AgeVC})
State 1
DriverWallet, {Name, AgeVC}
CitizenRegService, {}
MassCommService, {Name, AgeVC}
TrafficPolice, {}
Blockchain, {AgeProof}
Event 2 = (Raw, Blockchain, {MassCommService}, {AgeProof})
State 2
DriverWallet, {Name, AgeVC}
CitizenRegService, {}
MassCommService, {Name, AgeVc, AgeProof}
TrafficPolice, {}
Blockchain, {AgeProof}
Event 3 = (Broadcast, MassCommService, {Blockchain}, {LicenseProof})
State 3
DriverWallet, {Name, AgeVC}
CitizenRegService, {}
MassCommService, {Name, AgeVc, AgeProof}
TrafficPolice, {}
Blockchain, {AgeProof, LicenseProof}
Event 4 = (Implicit, MassCommService, {DriverWallet}, {LicenseVC})
Event 5 = (Revoke, DriverWallet, {MassCommService}, {Name, AgeVC})
State 4
DriverWallet, {Name, AgeVC, LicenseVC}
CitizenRegService, {}
MassCommService, {AgeProof}
TrafficPolice, {}
Blockchain, {AgeProof, LicenseProof}
Event 5 = (Implicit, DriverWallet, {TrafficPolice}, {LicenseVC})
State 5
DriverWallet, {Name, AgeClaim, LicenseVC}
CitizenRegService, {}
MassCommService, {AgeProof}
TrafficPolice, {LicenseVC}
Blockchain, {AgeProof, LicenseProof}

Figure 6.6: Instance of the SSI data sharing events in the DIVA.

in the sixth step, the architectural design and the state transition system instance of the DIVA are encoded in Alloy. The components and data objects from the DIVA are represented by signature extensions, as follows:

```
abstract sig Component { compType: one ComponentType, own: set DataObject  }
sig DriverWallet, MassCommService, CitizenRegService, TrafficPolice,
        CitizenDatabase, DrivingTestDatabase extends Component {}
abstract sig DataObject { dataType: one DataType, ownedBy: one Component }
sig AgeClaim, AgeVC, LicenseVC, LicenseProof, AgeProof extends DataObject {}
```

Evidently, the SSI data sharing model is applicable and is easily adaptable to domain-specific applications. Then, particular types and protocols in the DIVA are defined as facts in Alloy to restrict how SSI data sharing events should occur. The following facts are some of the constraints that have been declared:

```
fact{
        all c: DriverWallet | c.compType = IdentityWallet
        all c: MassCommService | c.compType = Issuer or c.compType = Verifier
        all c: CitizenRegService | c.compType = Issuer
        all c: TrafficPolice | c.compType = Verifier
        all d: Name | d.dataType = IdentityAttribue
        all d: AgeClaim | d.dataType = IdentityClaim
        all d: AgeVC | d.dataType = VerifiableClaim
        all d: LicenseVC | d.dataType = VerifiableClaim
        all d: AgeProof | d.dataType = ProofSchema
        all d: LicenseProof | d.dataType = ProofSchema }
```

When the `run` command is used to execute the Alloy Analyzer tool to analyze the SSI data sharing model of the DIVA, it is evident from the output message that the model can be executed without error and that the tool successfully validates the security and privacy specification (general properties converted from the improved SSI system properties in CSSPS). The outcomes ensure that the DIVA architecture design can meet the specifications within the specified scope.

The results of the case study demonstrate that the output of the weakness identification component (i.e., SWIF) is applicable for updating the design to mitigate SSI-specific weaknesses, and that the updated design can be used to model a state transition system for analysis against security and privacy specifications (from CSSPS). A portion of the proposed method is interoperable as a semi-automatic batch procedure.

6.6 Discussion

In the preceding sections, the proposed approach and its parts are evaluated from multiple aspects. Advantages and limitations are discovered and merit discussion. This section will discuss the results for the proposed approach.

6.6.1 SSI-Specific Weakness Identification Using SWIF

For the identification of SSI-specific weaknesses, there are three important quality aspects that demonstrate the utility of SWIF: the performance in recommending weaknesses, the validity of the recommended results, and the validity of the knowledge graph used to extend current knowledge.

Several quality metrics, including precision, recall, and accuracy, are employed to evaluate the performance of recommending weaknesses. Regarding quantitative performance, three substantial advantages are identified. First, as described in the last row of Table 6.2, SWIF is able to recommend a list of SSI-specific weaknesses with a significantly higher precision score (86.67 percent) than the existing and comparable approach to weakness recommendation. The precision score reflects the observation that the majority of recommended SSI-specific weaknesses are observed to be actual weaknesses. When comparing the precision score to the existing approach, it can be discovered that the change in vectorization approach, the cross-domain expansion, and the voting mechanism are introduced and contribute to the approach's higher precision. Second, it can be realized that SWIF can also recommend SSI-specific weaknesses with a significantly higher metric of accuracy (80.39 percent). This accuracy metric indicates that SWIF can distinguish between actual and irrelevant security weaknesses in accordance with the SSI functional requirements. Based on the additional features, the accuracy of SWIF has increased by more than twofold (from 30.06 percent to 80.39 percent) in comparison to the existing method. Lastly, SWIF introduces two flexible threshold values that will be utilized in the weakness identification and recommendation process and will have an effect on the overall performance. As shown in the line graph of Fig. 6.1, the experiment demonstrates that the two thresholds must be modified to accommodate the dataset of security weakness descriptions. Depending on the configuration of the thresholds, it is able to report an accuracy peak. This flexibility of threshold values impacts the performance of SWIF and highlights the benefit that SWIF can be applied to any database that collects security weaknesses in a manner similar to the CWE database. These three benefits leverage the need for knowledge transfer and additional features to fine-tune the process of weakness identification and recommendation. There is also a performance limitation due to the low recall and f1-score metrics, as shown in Table 6.2. These metrics indicate that SWIF has difficulty exploring the entire corpus of weaknesses in order to recommend the maximum number of SSI-specific weaknesses. Comparing this phenomenon to the existing approach, which does not include cross-domain expansion and other additional features, the results indicates that the introduction of a voting mechanism may exclude certain security weaknesses, which may be actual weaknesses. This limitation can be addressed by adjusting the minimum acceptable weight of weakness threshold in Algorithm 3.

On the other hand, the validity of the recommended SSI-specific weaknesses is evaluated in Section 6.3.1. Since the three conditions were defined, the evaluation results show that SWIF can identify SSI-specific weaknesses that are directly associated with the core functionality of the SSI model. As in the evaluation, SWIF can identify approximately 67 percent of the directly-associated results reported in Table 6.4. However, the identification of directly-associated SSI-specific weaknesses can only be performed manually because it requires domain expertise. SWIF alone cannot guarantee the validity of the recommended SSI-specific weaknesses.

An advancement on the knowledge of weakness identification in this dissertation comes from the proposal of the SSI-CWE cross-domain transfer knowledge graph. The experiment in Section 6.3.2 reported that the proposed knowledge graph is evaluated against the quality requirements by a group of participants. In contrast, the experiment described in Section 6.1.3 revealed that the level of knowledge provided by the proposed knowledge graph could expedite the identification of weaknesses. Two advantages are discovered from the evaluations. Firstly, the result demonstrates that the proposed knowledge graph possesses most of the quality requirements as shown in Table 6.5. The proposed knowledge graph is concise with relevance information from the SSI model and the CWE weakness domains. Secondly, the proposed knowledge graph can provide cross-domain knowledge that can influence weakness identification. It can be seen that the accuracy increases proportionally after the incorporation of SSI management system-specific facts (as reported in Table 6.3). However, it cannot be guaranteed that the more pertinent data provided by the knowledge graph will result in more precise identification. As the results of the 25 percent of facts and complete knowledge graph are provided, it is possible that the accuracy will not improve if the essential information is already included. With the incorporation of the synonyms and the domain-specific terms extracted from the white papers, a limitation on the term redundancy is realized by the participants. This research argues that this limitation can be accepted because the coverage on terms that will be influenced the identification of language correlations is necessary.

6.6.2 Improved SSI System Properties in CSSPS

As part of the improvement of SSI system properties, a new compliance set of SSI system properties (i.e., CSSPS) was developed. Various aspects can be analyzed to illustrate the advantages and limitations of CSSPS. This section highlights and discusses the increasing degree of consistency and the applicability of CSSPS in actual SSI management systems.

As a primary objective for the improvement of SSI system properties, a greater degree of consistency between SSI system properties and credible source documents is intended to be ensured. As reported in Table 6.6, two substantial advantages to the increasing degree of consistency are discovered. First, the property improvement could explicitly increase consistency between SSI system properties and source documents in all four cases, shown in Table 6.6. The differences (Δ) in all four cases of the evaluation result in positive values, indicating that the SSI system properties in CSSPS are more consistent than the existing set. In addition, the results of the evaluation indicate that SSI system properties in CSSPS can achieve greater than 85 percent consistency in each of the four cases (according to the sixth column of Table 6.6). CSSPS provides implementers of SSI management systems with a source of knowledge for developing instances of SSI management systems that can maintain a certain level of information security and privacy. Second, the comparative analysis of shared controls from multiple source documents could contribute to the universality of SSI system properties in CSSPS. Incorporating missing endorsed tasks in shared controls into SSI system properties enhances the level of consistency for both included and excluded source documents in our improvement. Even if source documents are not included in the improvement, the results of the evaluation in the third and fourth rows of Table 6.6 indicate that SSI system properties in CSSPS are approximately 10 percent more consistent than the current set due to the positive value of $\Delta_{\mathcal{P}_2,\mathcal{P}_1}$. These results demonstrate that the SSI system properties in CSSPS are highly advantageous for enhancing the security and privacy of sensitive data in accordance with the wide range of source documents. The specific definition of SSI system properties in CSSPS and the compliance with an individual source document are, however, two limitations with regard to the degree of consistency. First, it can be recognized from the comments of the analyst who participated in the experiment in Section 6.3.3 that it is difficult to compare the definition of SSI system properties to endorsed tasks in controls of source documents. It is possible for auditors to disagree with the compliance of SSI system properties in CSSPS and controls in the source document being audited. Second, the experiment in Section 6.3.3 demonstrates that the improved SSI system properties in CSSPS were unable to ensure complete source document compliance. This phenomenon occurs when shared controls are utilized as opposed to controls from the source document. Nonetheless, this limitation is tolerable because universal compliance may be more advantageous when the target SSI management system must be made compliant with multiple source documents.

In addition, three noteworthy advantages regarding the applicability of SSI system properties in CSSPS are discovered. First, it can be observed that the majority of SSI system properties in CSSPS are already influenced by real-world SSI management systems. According to the evaluation results in Table 6.7, 35 of 42 SSI system properties in CSSPS have Yes or Unclear flags, indicating that approximately 84% have the potential to be adopted by actual SSI management systems. This result highlights the necessity of such SSI system properties in real-world scenarios. Second, it can be observed that SSI system properties in CSSPS with Unclear flags, showing in Table 6.7, are still useful even if they are not explicitly adopted by actual SSI management systems. When evaluating the applicability of SSI system properties, the Unclear flag indicates SSI system properties that may be supported or have no conflicts with the underlying technologies and architecture. For future implementations of both existing and new SSI management systems, it is still advantageous to incorporate the properties with less effort. Last but not least, it can be found that SSI system properties in CSSPS could save time and reduce the effort required to comply with applicable laws, regulations, or standards. Implementers of the SSI management system must, as is customarily required, conduct a security and privacy analysis of the final product in comparison with the applicable source documents. With CSSPS, implementors can concentrate on implementing the system's compliance with SSI system properties in CSSPS, while the system is likely to comply with approximately 85% of contents in relevant source documents (as reflected in the sixth column of Table 6.2). In contrast, a limitation is discovered when applying SSI system properties in CSSPS to develop actual SSI management systems. When an SSI management system implementation contains application-specific terminology, extensive domain knowledge is still required to map or transform SSI system properties to functional requirements. Because SSI system property definitions should be sufficiently general to be adopted regardless of the application domain. In some instances, the CSSPS still presents adoption obstacles, and its users must weigh this trade-off.

6.6.3 SSI Data Sharing Model Finding with Alloy

As part of the SSI data sharing model discovery process using Alloy Analyzer, a system model encoding is proposed to ensure the security and privacy specification of SSI data sharing events. The performance of the model analysis and the applicability to real-world situations are evaluated and their advantages and limitations are noted.

Encoding system models with the Alloy specification language enables the capability to automatically find model instances. Nonetheless, Alloy and Alloy Analyzer will be useful if they can find model instances in a reasonable amount of time. Typically, the Alloy Analyzer's execution time is dependent on the model's size and scope. If the model encoding is too complicated, the execution time may increase dramatically. In Section 6.2.1, it is stated in the line graph in Fig. 6.2 that the proposed encoding of the SSI data sharing model exhibits linear behavior when the number of signatures increases. The only limitation occurs when a large number of components and data objects are configured. The evaluation results of performance in Fig. 2.10 indicates that the execution time increases exponentially with the number of system states and sharing events. However, even if the execution time is exponentially increased, it will not exceed a reasonable amount. The execution of the largest model in the evaluation still takes less than seven minutes.

Regarding applicability, it can be discovered that the proposed SSI data sharing model could provide a significant advantage to the customized SSI management system. For instance, the case study in Section 6.5 demonstrates that the proposed encoding makes use of abstract signatures to allow the SSI data sharing model to be extended or customized. For instance, the component signature can be extended to the driving test database component. The case study demonstrates that the proposed SSI data sharing model can be implemented with the domain-specific SSI management system. These advantages promote the applicability in real-world situation as shown in the Alloy code block in Section 6.5. However, the case study also includes a limitation, which is the difficulty of manual analysis. As demonstrated in Section 5.4.3, when the model's scope is expanded, the graph visualization becomes excessively complicated. It may restrict the analyst's ability to manually verify the correctness of the SSI data sharing model and increase the analyst's reliance on Alloy Analyzer reports and output messages.

6.6.4 Analysis of Security Weakness and Privacy Preservation

In the preceding sections, the advantages and limitations of each component of the proposed approach were discussed, but it is intended to be implemented in batches. Therefore, the advantages and limitations of the whole proposed approach must also be discussed. This section discusses two quality aspects of the overall approach, including the security and privacy coverage, the internal collaboration, and the supportability of domain knowledge.

In general, mistakes in the system's design and functionality can pose threats to information security and privacy. The proposed approach aims to account for both mistakes in order to maximize information security and privacy. This research discovered three substantial advantages in terms of information security and privacy coverage. First, the proposed approach analyzes information security and privacy properties in accordance with applicable laws, regulations, and standards by using the encoding of general properties. CSSPS is an input for specifying information security and privacy general properties in the SSI data sharing model, as described in Section 5.3. The data integrity and key protection properties, for instance, represent security safeguards, whereas the access control and data minimization properties represent privacy safeguards. Although only eight SSI system properties are included in the specification (Table 5.3), they are suitable for constraining the SSI data sharing model. Second, the proposed approach incorporates an early-stage analysis of information security and privacy. For instance, we use SSI functional requirements of the DIVA case study in Table 6.9 as an input for security weakness identification. In contrast to conventional security analysis, which is typically a retrospective activity, the proposed approach can be implemented as soon as the requirements have been elicited and the functional design is complete. Since requirements and design are required artifacts that define how the SSI management system is developed, it is highly advantageous to identify and mitigate information security and privacy risks as soon as possible. Lastly, the typical integration process of the proposed approach could help automatically reduce the number of common security weaknesses that must be mapped to the SSI management system's specifics with high precision and accuracy (Table 6.2). This automatic process is dependable and significantly reduces the scope of the manual process. However, a limitation on the scope of information security and privacy coverage must be noted. Since the proposed approach can guarantee that model instances that do not violate security and privacy SSI system properties can be found, it cannot guarantee that the SSI management system implementation is devoid of security weaknesses identified by SWIF. This limitation presents an opportunity for additional research.

Regarding internal collaboration, three advantages of the applicability of the proposed approach is identified. Due to the fact that the proposed approach combines three significant parts that work together to maximize information security and privacy coverage. First, the case study reveals that the proposed approach's parts can collaborate with others without difficulty. For instance, the case study presented in Section 6.5 demonstrates that the results of weakness identification suggest two additional components where SSI data sharing events may occur. The encoding will then display the additional components explicitly. Second, the typical integration process of the proposed approach could reduce the time and resources required to identify SSI-specific weaknesses. The case study demonstrates that the SWIF implementation can automatically reduce the number of correlated common security weaknesses from hundreds to ten (Table 6.10). This could significantly reduce the cost of analyzing the entire corpus of common security weaknesses in order to identify SSI-specific weaknesses. Lastly, the typical integration of the proposed approach could be effectively applied to enhance the design of the target SSI management system and use it to satisfy security and privacy specifications. According to Fig. 6.5, two additional components are added in order to mitigate the identified SSI-specific weaknesses. The application of the standard integration procedure can reduce the likelihood of missing instances of privacy and security issues. The manual collaboration is, however, a limitation of the proposed approach. It demonstrates in the case study that the results of weakness identification necessitate manual effort to update and mitigate the identified weaknesses. Consequently, the proposed approach cannot be executed in its entirety automatically.

A significant advantage, which is the supportability of domain knowledge in every part of the proposed approach, is discovered. Since SSI management systems or the SSI model concept are unique and contain complex mechanisms, it is advantageous to incorporate domain knowledge to improve the quality of analysis results. The SWIF explicitly incorporates domain knowledge into the cross-domain transferring knowledge graph, whereas the improvement of SSI system properties in CSSPS customizes property definitions to SSI-specific terminology. According to Table 6.2, the addition of cross-domain expansion improves the accuracy of an analysis in comparison to its absence. This dissertation ensures that the proposed approach takes domain knowledge of SSI management systems into account as an important factor and that each part effectively supports the incorporation of domain knowledge. However, there is also a limitation regarding the supportability of domain knowledge. It can be acknowledged that the domain knowledge incorporated into the proposed approach is based on the fundamental concept of the SSI model, and that it may not be optimally suitable for application-specific implementation of the SSI management system. For instance, the DIVA case study demonstrates that the recommended SSI-specific weaknesses may not be applicable to the intended implementation because the keyword used in its functional requirements may be distinct. However, the proposed approach is adaptable enough to accommodate this variation. For instance, the abstract signature is utilized in the encoding of the SSI data sharing model, which allows it to be tailored to application-specific terminology. As presented in Section 6.5, the recommended SSI-specific weaknesses can be used to reconsider the SSI functional requirements in SWIF.

6.6.5 Threats to Validity

Threats to the validity of this dissertation will be discussed in this section to justify the chosen methodology. This section will cover both internal and external threats.

Internal Threats

According to the contents of this dissertation, a number of internal threats to its validity were identified in various perspectives of the proposed approach.

During the research and evaluation, four internal threats to the validity of the SWIF proposals were identified. Each threat's mitigation can be explained and justified as follows:

1. Based on four documentation-defined functionalities of actual SSI management systems, the SSI-CWE cross-domain transfer knowledge graph was developed. Additionally, product-related terms are considered to be entity nodes. Nonetheless, this dissertation is confident that the selection of the four documents is accurate and provides coverage of differences in size, market target, and feature characteristics. On the other hand, the selected SSI management systems are all based on the fundamental concepts of the SSI model and are expected to share a common SSI-specific terminology. Therefore, the terminology employed in the development of the proposed knowledge graph is adequate and pertinent.
2. Five participants, who may not be experts in the SSI management system domains, conduct the evaluation of the proposed knowledge graph's validity. Since the purpose of the evaluation is to verify the quality requirements, it is straightforward to ascertain whether or not the participants comprehend the context and notation of the proposed knowledge graph. In order to support the evaluation, the online questionnaire requests that all participants review the terminology and basic concepts of the SSI management system and CWE weakness database. This dissertation is confident that participants can contribute valuable inputs to the experiment due to their qualifications and introductions.
3. In the evaluation of the SWIF's accuracy, the ground truth is compared to the identified SSI-specific weaknesses. The reliability of the ground truth directly reflects the validity of this evaluation. To ensure its reliability, three system analysts with a minimum of three years of experience in software design and requirement analysis are invited to review the document. This qualification is intended to demonstrate the ability to textually map the SSI functional requirements to the

weakness description, in a manner similar to how they analyze typical system requirements. In addition, analysts are introduced to the fundamental concept of the SSI management system in order to ensure that they comprehend the SSI functional requirements. In order to collect a single set of ground truth, the decisions of three analysts are normalized. This dissertation is confident that the collected ground truth is reliable and mitigates the threat.

4. The authors perform a subjective and manual evaluation of the validity of the identified SSI-specific weaknesses. As stated in Section 6.3.1, validity can only be defined using the SSI management system concept. It is extremely challenging to provide objective statistical data regarding the validity. To mitigate this threat, the authors intend to provide and publish a clear justification for their decision. These arguments may bolster the validity of the evaluation's findings.

Then, for the part of the property improvement and the CSSPS proposals, four additional internal validity threats are identified during the improvement procedure and evaluation. Each threat's mitigation will be explained and justified as follows:

1. One system analyst evaluates the degree of consistency between SSI system properties and controls from source documents, which appears to be a small sample size. Nonetheless, the experiment in the evaluation is intended to require the analyst to compare two collections of texts. This dissertation is confident that the analyst's decisions are straightforward because two sets of text are comparable and supported by empirical evidence. Therefore, additional participants are not required to validate the conclusions, which are unlikely to result in statistically significant differences.
2. As comparators for the evaluation of the degree of consistency, only four source documents were chosen. It is possible to argue that the improved SSI system properties in CSSPS are incompatible with other source documents. To mitigate this threat, the selection of source documents includes both those that are included as sources for property improvement and those that are not. This selection is anticipated to increase the validity's coverage scope. This dissertation is confident that the chosen source documents are appropriate for the evaluation.
3. In the evaluation of applicability, the authors assign the possession flags of the improved SSI system properties in CSSPS based solely on the documentation. At the time of this research, it was difficult to gain access to the actual implementers of SSI management system, and the actual technical design specifications were typically confidential, so the

documentation was the best available source. Despite the fact that the documentation cannot cover all of the functionality implemented in the SSI management system in comparison to the design specification, the documentation should be provided to inform users of the system's key features and functionality. As a result, such documentation must be trustworthy and should not deviate significantly from the actual implementation. Aside from that, the evaluation proposes using three possession flags: yes, no, and unclear. The unclear flag is provided to reduce the likelihood that the documentation is ambiguous. Additionally, the justification is provided and made public to demonstrate the basis for the decision. This dissertation is confident that the applicability evaluation is valid and can accurately reflect the present state of the improved SSI system properties in CSSPS.

The only threat to the validity of the SSI data sharing model is the fact that the security and privacy specifications are taken from the CSSPS proposal. Since the CSSPS is included in this dissertation, the legitimacy of defining the specification may be contested. The development of CSSPS is based on the consolidation of four existing proposals and the use of credible source documents. This dissertation is confident that the SSI data sharing model's specification is valid in light of current knowledge.

External Threats

In addition to internal threats, this dissertation identified external threats that may compromise the validity of the proposed approach when applied to the external environment. This section will summarize and discuss threats to validity from the following perspectives:

1. CWE is the only database of prevalent security weaknesses utilized in this dissertation. There may be limitations to the knowledge's reliability. However, because the CWE database is well-known and widely utilized, it should be a trustworthy source of information for research. Furthermore, the CWE database contains well-described content, particularly the weakness description, which the SWIF can use adequately and effectively. This dissertation is confident that the SWIF is applicable to any weakness database with a textual description similar to the CWE database.
2. The improvement of SSI system properties is carried out by comparing them to the shared controls from a particular source document version. If the updated version is frequently released, the improvement may be invalid. However, shared controls were derived globally, and updates

to source documents should not completely contradict their previous version. In addition, security and privacy controls are likely to remain unchanged. This dissertation is confident that the updated source documents is less likely to invalidate the improved SSI system properties in CSSPS.

3. Only three real-world SSI management system products were selected for the evaluation of the applicability of the improved SSI system properties in CSSPS. It is possible to argue that the improved SSI system properties are not applicable to real-world situations. Nonetheless, this threat is anticipated to be mitigated through the selection of SSI management system samples with varying sizes, scopes, features, and marketing objectives. The evaluation includes a smaller system (uPort) as well as two larger systems (Sovrin and IBM Verify Credentials). In addition, uPort is chosen because it is a system that does not strictly adhere to the SSI model concept. Different uPort features could reflect the expanding features. This dissertation is confident that the selection of actual SSI management systems can mitigate this threat effectively.

According to all discussed threats to validity and this dissertation's method for mitigating them, it is possible to conclude that this dissertation is concerned with validity and has made every effort to address these threats and enhance the research outcomes.

Chapter 7

Related Work

In this chapter, a literature review is conducted in order to compare the proposed approach, its outcomes, and its quality to previous research. Each component of the proposed approach is intended to be reviewed and compared to other techniques or approaches presented in the research literature, including approaches for identifying security weaknesses, enhancing and utilizing SSI system properties, and finding model instances achieving information security and privacy model properties. Lastly, the proposed approach is compared to other security and privacy analysis approaches of the SSI management system.

7.1 Weakness Identification Approaches

In this section, this dissertation intends to compare the SSI-specific weakness identification approach in SWIF to other existing vulnerability and weakness analysis approaches. Table 7.1 displays the landscape and comparable features. The table compares seven characteristics of each research paper, including the year of publication (Year), the analysis purpose (Goal), the evaluation target (Target), the analysis method (Method), the automatic capability of the analysis method (Automatic?), the scope of the analysis domain (Domain-Specific?), and their evaluation method and results (Evaluation).

Table 7.1: Comparison between SWIF and other weakness/vulnerability identification approaches

Reference	**Year**	**Goal**	**Target**	**Method**	**Automatic?**	**Domain-Specific?**	**Evaluation**
Gao et al. [102]	2013	Vulnerability	Design	Modeling	Manual	No	Case Study
Lu et al. [103]	2014	Weakness	Attack Patterns	Modeling	Manual	No	Case Study
Mokhov et al. [104]	2014	Weakness	Source Code	SPA	Automatic	No	N/D
Scandariato et al. [28]	2014	Vulnerability	Source Code	SPA	Automatic	Yes - Android App	P: >80%, R: >75%
Sun et al. [105]	2014	Vulnerability	Source Code	SPA	Automatic	Yes - E-commerce	Coverage
Nadeem et al. [29]	2015	Vulnerability	Source Code	SPA	Semi-Automatic	No	P: 85%
Son et al. [106]	2015	Weakness	Source Code	SPA	Automatic	No	N/D
Yu et al. [107]	2015	Weakness	Design	Modeling	Semi-Automatic	No	Case Study
Kim et al. [108]	2015	Vulnerability	Source Code	SPA	Semi-Automatic	Yes - Open Source Software	N/D
Stergiopoulos et al. [109]	2016	Vulnerability	Source Code	SPA	Automatic	No	TP: 100%
Wang et al. [110]	2018	Weakness	Source Code	SPA	Automatic	No	Case Study
Han et al. [111]	2019	Vulnerability	Design	Modeling	Automatic	No	Detection Rate: 30.9%
Bruzhuk [112]	2019	Vulnerability	Design	Modeling	Automatic	No	Case Study
Byun et al. [113]	2020	Weakness	Source Code	SPA	Automatic	No	R: 72%
Liu et al. [114]	2021	Vulnerability	Source Code	Modeling	Automatic	Yes - Smart Contract	ACC: 80~89%
Hariyanti et al. [27]	2021	Weakness	Design (Text)	ML	Automatic	Yes - Business Process	Reliability Metric
Skandylas et al. [115]	2021	Vulnerability	Design	Modeling	Semi-Automatic	No	N/D
Aidee et al. [116]	2021	Vulnerability	Source Code	SPA	Automatic	Yes - Smart Contract	FDR: 68%, FNR: 47%
Jeon & Kim [117]	2021	Vulnerability	Source Code	ML	Automatic	No	F1: 96.11%
Agarwal et al. [118]	2022	Vulnerability	Source Code	ML	Automatic	Yes - Smart Contract	Case Study
Zheng et al. [26]	2022	Weakness	Design (Text)	IR	Automatic	No	G-Measure: 71~87%
This work	**2022**	**Weakness**	**Design (Text)**	**IR**	**Semi-Automatic**	**Yes - SSI**	**P: >86%, ACC: >80%**

Note: P = Precision, R = Recall, N/D = Not clear determined, TP = True positive, ACC = Accuracy, FDR = False Detection Rate, FNR = False Negative Rate, SPA = Static Program Analysis, ML = Machine Learning, IR = Information retrieval.

Existing approaches attempt to analyze the security of a target system on multiple dimensions, such as vulnerabilities and weaknesses. In certain works, the terms vulnerability and weakness are combined or used interchangeably. We recognized that the target of analysis can serve as a distinguishing characteristic. Source codes, which are analyzed against vulnerable or weak code examples, are the most prevalent target of analysis (e.g., [28, 103]). Designs that demonstrate the operation of the target system are another common target of analysis (e.g., [102, 115]). Different analysis purposes also provide distinct benefits. The source code is an accurate and transparent representation of the system's functionality; however, it must be analyzed after the system has been developed. There may be such code examples in public databases of vulnerabilities and weaknesses, but not all of them. On the other hand, the design provides an implementation-ready high-level abstraction of the system's functionality as soon as the requirements are stable. However, there is no assurance that the implementation will strictly adhere to the design. Similar to other frameworks, SWIF establishes an analysis target for the design, as an early analysis of security weaknesses can be conducted and the design is more comparable to the textual description of weaknesses than source code. Also, the broader scope of security weaknesses is identified, which is not limited to the code examples not provided by all security weaknesses.

Comparing the analysis methods, it can be discovered that four distinct methods have been used in related research: static program analysis [28, 29, 104, 105, 106, 110, 113], security modeling [26, 27, 102, 103, 107, 111, 112], machine learning [117, 27, 118], and information retrieval [26]. In the early stages of security analysis, static program analysis and security modeling are prioritized, as shown in the table. Then, machine learning has been promoted alongside other techniques for analyzing security incidents when actual incident data is available. Using information retrieval techniques, textual components of security weaknesses have begun to be used in recent security analysis, which is an interesting advancement. SWIF joins this trend and employs information retrieval techniques to identify security weaknesses.

Another comparative feature is the inclusion of domain knowledge in the analysis. It is found that domain knowledge has been included in recent work to enhance the accuracy and comprehensiveness of the security analysis. For instance, knowledge of smart contracts is taken into account when analyzing blockchain applications [113, 116, 118]. We found that the inclusion of domain knowledge in the field of SSI management system has not been provided in existing work and the cross-domain knowledge transfer in sWIF is new to the weakness identification.

In terms of quality evaluation, numerous metrics, such as precision, recall, fault detection rate, or g-measure, have been used when evaluating the performance of security analysis. However, it is not fair to compare different metrics to each other. This dissertation recognized only few works that used comparable metrics, and reported that SWIF has a comparable accuracy and slightly higher precision scores than existing approaches [28, 29, 113].

7.2 Improvement and Utilization of SSI System Properties

In this section, the improvement of SSI system properties in CSSPS is compared to other proposals and how SSI system properties are utilized as evaluation criteria of SSI management systems.

First, CSSPS is compared to other proposals for SSI system properties and SSI principles, and then Table 7.2 is summarized the differences. As the SSI principles and SSI system properties are regarded as important for the development of SSI management systems, researchers have attempted to publish numerous proposals to improve the existing ones. This dissertation found five proposals for either SSI principles or SSI system properties [4, 7, 14, 15, 63] at the time of writing. Allen [4] was the first to propose ten fundamental SSI principles. His proposal focuses solely on the fundamental constraints of the SSI management system's functionality. Since Allen's proposal did not address information security and privacy, CSSPS has the advantage of incorporating a greater number of SSI system properties that comply with credible source documents for information security and privacy. In terms of the applicability of SSI system properties, Allen's proposal appears to have the same degree of applicability as CSSPS, as CSSPS incorporates SSI system properties that are consistent with Allen's SSI principles. In addition, subsequent work [7, 14, 63] made reference to Allen's proposal and sought to enhance security and privacy. The paper by Naik and Jenkins [14] was the first effort to improve the data privacy of Allen's SSI principles by referencing the GDPR [13]. However, they provide a few references demonstrating which GDPR sections are evidently adopted. Lopez [7] attempted to expand Allen's SSI principles in alignment with the laws of identity [12]. Ferdous et al. [15] transformed Allen's SSI principles to SSI system properties. It can be found that none of the existing proposals is enhanced in accordance with a variety of applicable laws, regulations, and standards. We can conclude that CSSPS preserves more information security and privacy than other existing proposals because it is more consistent to missing tasks.

Table 7.2: Comparison with other proposals of SSI principles and system properties of the SSI management system.

Reference	Year	Source of Principles /Properties	Improve?	Source of Improvement	Security?	Privacy?	Proposal
Allen [4]	2016	The SSI system concept	No	-	No	No	10 principles for the SSI management system.
Ferdous et al. [15]	2019	[4]	No	-	Yes	No	17 properties in five groups.
López [7]	2020	[4]	Yes	Digital identity model	No	No	16 principles for SSI digital identity.
Naik & Jenkins [14]	2020	[4]	Yes	GDPR [13]	No	Yes	20 privacy-preserved principles.
Pattiyanon & Aoki [63]	2022	[4, 15, 14]	Yes	5 standards and regulations	Yes	Yes	29 security and privacy SSI system properties.
This work	**2022**	[4, 15, 7, 14]	**Yes**	**28 sources (laws, regulations, and standards)**	**Yes**	**Yes**	**The CSSPS of 42 SSI system properties**.

Table 7.3: Comparison of the utilization of SSI principles and system properties to evaluate SSI management systems.

Reference	Year	Approach	Source of Principles /Properties	Evaluation Method
Ferdous et al. [15]	2019	In-depth analysis of SSI concepts and notations	Their proposed SSI system properties	Evaluate SSI management sysrtems against the proposed SSI system properties.
Haddouti & Kettani [8]	2019	Analysis of blockchain-based Identity management systems	Allen [4], Laws of Identity [12]	Evaluate SSI management systems qualitatively against principles.
Liu et al. [99]	2020	Review of blockchain-based identity management systems	Laws of Identity [12]	Evaluate SSI management systems against laws of identity.
Satybaldy et al. [119]	2020	Proposal of an evaluation framework for SSI systems	Allen [4], Laws of Identity [12]	Take principles to be requirements for evaluation.
Grüner et al. [10]	2021	Propose a design architecture of brokered SSI integration	Allen [4]	Evaluate the proposed design architecture against Allen's principles.
Soltani et al. [120]	2021	Survey on SSI ecosystems	GDPR [13], PSD2, Privacy-by-Design, eIDAS	Evaluate current SSI management systems against regulations.
Stockburger et al. [121]	2021	Design of SSI management systems in public transportation.	Allen [4]	Evaluate the design against Allen's principles.
Shuaib et al. [122]	2022	Review of identity model for land registry systems	Laws of Identity [12]	Evaluate different identity model against laws of identity.
Shuaib et al. [123]	2022	Review of SSI management systems for land registry systems	Allen [4], Laws of Identity [12]	Evaluate SSI management systems applicable to land registry systems against the principles.
This work	**2022**	**Compliance with laws, regulations, and standards**	**42 Proposed SSI System Properties**	**Evaluate SSI management systems against the proposed CSSPS.**

Table 7.4: Comparison of the SSI data sharing model to other approaches.

Reference	Year	Target	Specification	Encoding
Kumar and Shyamasundar [32]	2014	Any system model	Privacy Policy	Manual
Joshaghani et al. [33]	2019	Ubiquitous system model	Privacy Policy	SpEL & JavaSMT
Fareen et al. [124]	2020	Any system model	CTLFC	Alloy
This Work	2022	SSI data sharing model	security and privacy properties	Alloy

Notes: SpEL = Spring Expression Language, and CTLFC = Computational Tree Logic with Fairness Constraints.

Second, the CSSPS is defined similarly to existing proposals, but is able to adhere to applicable laws, regulations, and standards. Compared to how existing proposals have been utilized, demonstrating the utilization of CSSPS should be advantageous. Table 7.3 summarizes and compares the usage of SSI principles and SSI system properties in other works. Researchers have proposed secure, privacy-preserving SSI management system designs and architectures [10, 121]. Allen's SSI principles [4] and laws of identity [12] are the most commonly used standards for evaluating their results. The CSSPS is capable of evaluating future designs and architectures because it is based on Allen's principles and defined in the same way. The publication of a second group of review articles [8, 120, 123, 122] analyzing and evaluating identity models and real-world SSI solutions on multiple aspects. In Section 6.4.1, real-world SSI solutions were examined in the same vein as the review articles, and it can be recognized that the CSSPS can be used in future reviews. The comparison regarding the use of SSI system properties indicates that CSSPS is functionally equivalent to other proposals because it defines SSI system properties in the same way.

In conclusion, CSSPS could enhance the information security and privacy of SSI management systems in accordance with a variety of source documents. In addition, CSSPS can be utilized broadly because the SSI system properties contained within it have been used as criteria to evaluate the possession of actual SSI management systems, similar to the existing proposals.

7.3 SSI Data Sharing Model Finding

In this section, the way SSI data sharing events is modeled and encoded in Alloy for automatic model finding capability is compared to other approaches. Table 7.4 is created to compare the SSI data sharing model to other existing approaches to find model instances of secure and privacy-preserved information sharing events.

First, it can be determined that information flow analysis is a prominent technique for analyzing information security and privacy. Existing research attempts to use privacy policies as a criterion for regulating information-sharing events. Kumar and Shyamasundar [32] were the first to propose a technique for implementing privacy policies by manually analyzing the state transition system for data sharing events. A subsequent work [33] that adopts the concept of information flow analysis on a state transition system against privacy policies and extends it to an analysis with data read/write concepts and formalization for automatic analysis with JavaSMT is introduced. From these previous works, it can be determined that the proposed SSI data sharing model employs the same concept by defining a state transition system via the flow of data sharing. However, this dissertation argues that the SSI data sharing model's information security and privacy must go beyond the privacy policy. The proposed model for SSI data sharing expands the scope of analysis to include both information security and private SSI system properties.

Second, the proposed SSI data sharing model uses the Alloy specification language as its encoding mechanism. As the SSI data sharing model is defined based on a state transition diagram, we compare it to a previous work by Fareen et al. [124] that proposes a transitive-closure-based model checking strategy. It can be observed that the proposed SSI data sharing model employs the same concept to define the state transition system. However, they used the CTLFC as a specification, which is excessive in terms of information security and privacy properties. In actuality, the prior work does not appear to adequately support encoding the SSI data sharing model.

7.4 Security and Privacy Analysis of SSI Management Systems

Last but not least, the entire proposed approach for analyzing security weaknesses and preserving the privacy of the SSI management system must be compared to other works that analyze or improve information security and privacy. This section summarizes and creates Table 7.5 to compare various factors visually. The table contrasts the target of a security and privacy analysis or improvement (Target) with the actual solution implemented (Solution). To represent the scope of the work, how the proposed approach addresses security (Security) or privacy (Privacy) is emphasized.

In previous analysis and improvement, it is discovered that the design of SSI management systems and identity data are common target artifacts for

Table 7.5: Comparison of the proposed approach for security weakness and privacy preservation analysis to other related work.

Reference	Year	Target	Solution	Security	Privacy
Grüner et al. [96]	2018	Identity data	QM	Yes	No
Bernabe et al. [125]	2019	SSI design	SDM	Yes	Yes
Grüner et al. [126]	2019	SSI design	QM	Yes	No
Bhattacharya et al. [20]	2020	Identity data, SSI design	QM	Yes	Yes
Luecking et al. [127]	2020	SSI design	SDM	Yes	No
Naik & Jenkins [14]	2020	SSI principles	PR	No	Yes
Yang et al. [16]	2020	SSI design	SDM	Yes	Yes
Kim et al. [19]	2021	DID document	TI, PR	No	Yes
Rhie et al. [18]	2021	DID document	TI	Yes	No
Kirupanithi & Antonidoss [128]	2021	SSI design	SDM	Yes	No
This work	2022	SSI design	TI, PR, MF	Yes	Yes

Note: QM = Quality Modeling, SDM = System Design and Modeling, PR = SSI Principles and System Properties, TI = Threat Identification, and MF = Model Finding.

representing information security and privacy. In addition to focusing on the design of SSI management systems for analyzing security weaknesses and preserving privacy, the proposed approach is also centered on the analysis of security weaknesses and the preservation of identity data. At the earlier stage, researchers frequently focus on the quality modeling of data attribute trustworthiness within the SSI management system [96, 126, 20]. This results in a method for selecting suitable actions or discriminating against trust issuers. However, it is discovered that the data attribute trustworthiness are typically beyond the control of those implementing the SSI management system. This dissertation choses to disregard a concern about the data attribute trustworthiness in the proposed approach. Multiple design alternatives [16, 125, 127, 128] were subsequently proposed to enhance the security and privacy of the core SSI management system. This dissertation argues that the secure and privacy-preserving designs are contingent on the actual implementation, and that the variation in implementation necessitates additional investigation into information security and privacy. Therefore, the comparison concludes that the proposed approach for analysis is more advantageous than a design proposal at this stage. Threat identification is yet another method for determining the current level of information security and privacy. However, previous research has focused on identifying risks of data leaks [19] and vulnerabilities in the DID update process [18]. Numerous aspects of security remained unaddressed. The proposed method utilizes security weaknesses as an analysis target, where very few prior work has done. The above justifications establish the proposed approach as novel in the field of security and privacy analysis and SSI management system improvement.

Chapter 8

Conclusion

8.1 Concluding Remarks

Since information security and privacy in SSI management systems are essential to gaining users' trust, they must be thoroughly analyzed during the design phase. However, when analyzing security and privacy preservation in SSI management systems using existing knowledge, three technical problems were identified:

P1: Knowledge on SSI-specific security weaknesses has not been investigated yet and it is difficult to employ existing identification approaches to identify without domain knowledge;

P2: Existing SSI system properties are not universally consistent with a vast collection of credible laws, regulations, and standards to cover adequate governance of security and privacy;

P3: Existing approaches to analyze data sharing events are incompatible to ensure unique security and privacy specification of the SSI management system.

This dissertation recognizes the need to address the mentioned problems and proposes an approach for security weakness and privacy preservation analysis. The proposed approach is intended to be a combination of numerous techniques for addressing each problem. The outcomes of this dissertation in response to the aforementioned problem are as follows.

First, this dissertation proposes SWIF as a framework for identifying SSI management system-specific security weaknesses. By incorporating cross-domain knowledge transfer and information retrieval techniques, SWIF was able to identify language correlations between SSI functional requirements and text descriptions of security weaknesses. The identified language correlations can indicate the specificity of security weaknesses in relation to SSI management system functionality. This dissertation discovered that SWIF could identify weaknesses with greater precision and accuracy than existing methods in the early stages of development. This part of the proposed approach can overcome the first problem (**P1**).

Second, this dissertation employs a systematic review process to improve information security and privacy in SSI system properties in accordance with a vast array of applicable laws, regulations, and standards. CSSPS is a new compliance set of 42 SSI system properties resulting from this dissertation. 28 source documents whose controls are suitable for constraining SSI management systems are exhaustively selected. Comparing to the existing SSI system properties, missing endorsed tasks in the controls shared among the source documents are identified. The missing tasks is used as a source of information to revise and update the definitions of SSI system properties and introduce new SSI system properties to cover more information security and privacy controls. This dissertation found that, compared to the existing SSI principles and SSI system properties, the improved SSI system properties in CSSPS are more consistent with a wide variety of source documents (i.e., both source documents that were included in our improvement and ones that were excluded). The outcome of this part can provide an evidence to resolve the second problem (**P2**).

This dissertation concludes with a method for finding secure and privacy-preserving SSI data sharing models in Alloy. It is realized that SSI data sharing events are unique and governed by particular constraints that must be managed. These constraints are utilized to define a state transition system that represents the states of SSI data sharing events among SSI-specific technical components. Eight information security and privacy properties from CSSPS that are realizable through the SSI data sharing model are also specified. The SSI data sharing model and its specifications for information security and privacy are both encoded in the Alloy specification language to enable automatic analysis and model finding with the Alloy Analyzer tool. The encoding demonstrates that the SSI data sharing model successfully and without violation possesses the specified information security and privacy specification. The proposed model for SSI data sharing is able to abstract and analyze how SSI data sharing events maintain security and privacy. The proposed SSI data sharing model could address the third question (**P3**).

In conclusion, this dissertation provides a combination of methods that can function as a batch procedure. First, SWIF can be used to create a design of the target SSI management system that contains as few security weaknesses as possible, and then this design can be utilized to model SSI data sharing events in Alloy. This combination shall maximize information security and privacy in SSI management systems by boosting the comprehensiveness of security weakness and privacy preservation analysis.

8.2 Dissertation Contributions

This section outlines the two-fold contributions of this dissertation: academic contribution and practical contribution.

8.2.1 Academic Contribution

This dissertation strengthens existing approaches to security weakness and privacy preservation analysis by proposing expanded approaches with previously limited features. SWIF is the first method to incorporate cross-domain knowledge transfer into the identification of weaknesses based on textual functional requirements. This capability encourages researchers to identify security weaknesses in functional requirements and ensure that domain knowledge is adequately incorporated. The identified SSI-specific weaknesses can be used to advance the knowledge of security weaknesses that are specific to the SSI management system. Additionally, SWIF's performance is superior to that of comparable existing methods. This should aid in the future implementation of recommendation systems for security weaknesses.

CSSPS provides the most up-to-date SSI system properties pertaining to security and privacy that are broadly compliant with credible source documents. CSSPS contributes to the quality of SSI management systems as a whole, which is typically aligned with at least one set of SSI principles or SSI system properties. Further research on designing a secure and privacy-preserved SSI management system may use SSI system properties in CSSPS to elicit security and privacy requirements or as criteria for evaluating the protection of information security and privacy in the design. In addition, the consolidation of previous proposals conducted in this dissertation contributes to a collection of the most recent SSI principles and SSI system properties that can be cited as a single source. Shared controls derived from this dissertation are also advantageous in the information security and privacy domains because they provide a comprehensive landscape of security and privacy safeguards derived from a large number of laws, regulations, and standards. In addition to the SSI management system, researchers may also use these shared controls to limit or evaluate their research subjects.

The SSI data sharing model emphasizes the extensive capacity to analyze information security and privacy automatically. The SSI data sharing model defines information security and privacy in data sharing for the first time and is tailored to the SSI management system. The SSI data sharing model can serve as a jumping-off point for further research into information security and privacy in SSI management systems.

The academic contribution of the entire proposed approach is the combination of existing approaches to improve the ability and comprehensiveness of the security and privacy analysis in a domain-specific application. The idea to include domain knowledge to the analysis can be applied to other domains if the intermediate artifacts in the proposed approach are tailored to such domain applications.

8.2.2 Practical Contribution

This dissertation identifies key contributions to the implementation and management of actual SSI management systems. After analyzing and mitigating security and privacy issues, it enables SSI management system implementations to reach a higher level of information security and privacy.

SWIF is implemented as a Python recommendation system executable with any given set of SSI functional requirements. The stable version of the SWIF implementation will be made publicly available on Github. Since Python is a well-known programming language, the SWIF implementation will not be difficult to utilize. Real-world practitioners or implementers of SSI management systems can utilize the SWIF implementation to analyze the SSI functional requirements of their target SSI management system.

Similar to existing sets of SSI system properties, CSSPS can be used to constrain the operation of real-world SSI management systems. Sovrin [101], for instance, adopted Allen's SSI guiding principles [4] as an evaluation framework to guarantee the accuracy of their suggested architecture. The SSI system properties in CSSPS allow implementers of real-world SSI management systems to elicit both functional and non-functional requirements. In addition, SSI system properties in CSSPS can be used to generate test cases to ensure information security and privacy during implementation.

Lastly, the extensibility of the SSI data sharing model enables implementers of SSI management systems to extend and adapt it to their specific implementation. As demonstrated in the use case, the SSI data sharing model is available for analyzing information security and privacy and finding valid model instances applicable to real-world applications. Since it is based on the formalization of a state transition system, practitioners can also reimplement the SSI data sharing model in their preferred specification language.

8.3 Limitations and Future Works

The limitations of the proposed approach for security weakness and privacy preservation analysis and directions to overcome these limitations are summarized below.

1. The proposed SWIF in identifying SSI-specific weaknesses is limited in its ability to effectively explore the weakness corpus due to a low recall metric. This limitation stems from the implementation of a weakness voting mechanism, which prioritizes weaknesses related to multiple functions over others. To improve the suggested procedure within the SWIF, steps should be taken to balance or increase the exploration of the corpus while still preserving the ability to select crucial results.
2. The proposed approach aimed at enhancing the SSI system properties results in the creation of a new property set, namely CSSPS. While the CSSPS exhibit a lower degree of inconsistencies compared to source documents, it constitutes a large collection and may not be fully consistent with any single source document. This limitation is viewed as a trade-off between coverage and usability. To address this limitation, a solution that facilitates the ease of use and extends the CSSPS towards complete source document compliance should be investigated. This could be achieved by utilizing a checklist or a rule-based verification approach as evaluation criteria when using the improved SSI system properties in CSSPS.
3. The proposed modeling approach for analyzing SSI data sharing events aims to ensure the satisfaction of eight specific security and privacy specifications related to data sharing, derived from the improved SSI system properties. However, it is limited in scope due to the use of the SAT solver in the Alloy Analyzer tool and the specifications. To further extend the analysis of SSI data sharing events and guarantee a more comprehensive coverage of security and privacy aspects, an alternative approach, such as the use of an SMT solver, could be considered.
4. The proposed approach in this dissertation consists of three main solutions, but the integration between them requires manual involvement and domain expertise. Although the output of one solution can be utilized as input for the subsequent solution, it cannot be utilized directly without manual intervention. This limitation may lead to potential human error during the analysis. To address this limitation, it is suggested to optimize the process and consider implementing an automated pipeline to minimize the risk of human subjectivity.

Publications

International Conference Proceedings

- Charnon Pattiyanon, and Toshiaki Aoki. “Analysis and Enhancement of Self-sovereign Identity System Properties Compiling Standards and Regulations.” In: *Proceedings of the 8th International Conference on Information Systems Security and Privacy*, ICISSP 2022, pp. 133–144, 2022. doi: 10.5220/0010877300003120

- Charnon Pattiyanon, Toshiaki Aoki, and Daisuke Ishii. “A Method for Detecting Common Weaknesses in Self-Sovereign Identity Systems Using Domain-Specific Models and Knowledge Graph.” In: *Proceedings of the 10th International Conference on Model-Driven Engineering and Software Development*, MODELSWARD 2022, pp. 219–226, 2022. doi: 10.5220/0010824900003119.

- Charnon Pattiyanon, Takashi Tomita, and Toshiaki Aoki. Modeling and Analyzing Self-Sovereign Identity Specific Data Sharing Events in Alloy, 2023. (To be submitted).

International Journals

- Charnon Pattiyanon, and Toshiaki Aoki, “Compliance SSI System Property Set to Laws, Regulations, and Technical Standards,” *IEEE Access*, vol. 10, pp. 99370–99393, 2022. doi: 10.1109/ACCESS.2022.3204112.

- Charnon Pattiyanon, and Toshiaki Aoki, “Weakness Identification Framework for SSI Management Systems Using Cross-Domain Recommendations and a Knowledge Graph,” 2023. (Submitted, Under Review).

Bibliography

[1] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "Digital identity guidelines," Standard NIST Special Publication 800-63-3, National Institute of Standards and Technology (NIST), June 2017.

[2] Gartner, "Identity And Access Management (IAM) - Gartner Glossary." https://www.gartner.com/en/information-technology/glossary/identity-and-access-management-iam, 2022. Accessed: 2022-09-17.

[3] Gartner, "Identity Management - Gartner Glossary." https://www.gartner.com/en/information-technology/glossary/identity-management, 2022. Accessed: 2022-09-17.

[4] C. Allen, "The path to self-sovereign identity." http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html, 2016. Accessed: 2022-05-17.

[5] M. Sporny, D. Longley, and D. Chadwick, "Verifiable Credential Data Model v.1.1 - Expressing verifiable information on the Web." https://www.w3.org/TR/vc-data-model/, 2019. Accessed: 2022-05-18.

[6] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen, "Decentralized Identifiers (DIDs) v.1.0. - Core architecture, data model, and representations." https://www.w3.org/TR/did-core/, 2021. Accessed: 2022-05-18.

[7] M. A. López, "Self-Sovereign Identity - The Future of Identity: Self-Sovereignity, Digital Wallets and Blockchain," tech. rep., Inter-American Development Bank, 2020.

[8] S. E. Haddouti and M. D. E.-C. E. Kettani, "Analysis of Identity Management Systems Using Blockchain Technology," in *Proceedings of the 2019 International Conference on Advanced Communication Technologies and Networking*, CommNet 2019, pp. 1–7, 2019.

[9] J. Lee, J. Y. Hwang, J. Choi, H. Oh, and J. Kim, "SIMS : Self Sovereign Identity Management System with Preserving Privacy in Blockchain," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 1241, 2019.

[10] A. Grüner, A. Mühle, and C. Meinel, "ATIB: Design and Evaluation of an Architecture for Brokered Self-Sovereign Identity Integration and Trust-Enhancing Attribute Aggregation for Service Provider," *IEEE Access*, vol. 9, pp. 138553–138570, 2021.

[11] E. Samir, H. Wu, M. Azab, C. Xin, and Q. Zhang, "DT-SSIM: A Decentralized Trustworthy Self-Sovereign Identity Management Framework," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 7972–7988, 2022.

[12] K. Cameron, "The Laws of Identity." https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf, 2005. Accessed on: 2022-09-19.

[13] European Parliament and the Council of the European Union, "The General Data Protection Regulation (GDPR)." https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016. Accessed: 2022-08-19.

[14] N. Naik and P. Jenkins, "Governing principles of self-sovereign identity applied to blockchain enabled privacy preserving identity management systems," in *Proceedings of the 2020 IEEE International Symposium on Systems Engineering*, ISSE 2020, pp. 1–6, 2020.

[15] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In Search of Self-Sovereign Identity Leveraging Blockchain Technology," *IEEE Access*, vol. 7, pp. 103059–103079, 2019.

[16] X. Yang and W. Li, "A zero-knowledge-proof-based digital identity management scheme in blockchain," *Computers & Security*, vol. 99, p. 102050, 2020.

[17] Q. Stokkink, G. Ishmaev, D. Epema, and J. Pouwelse, "A Truly Self-Sovereign Identity System," in *Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN)*, LCN 2021, (Los Alamitos, CA, USA), pp. 1–8, IEEE Computer Society, oct 2021.

[18] M.-H. Rhie, K.-H. Kim, D. Hwang, and K.-H. Kim, "Vulnerability Analysis of DID Document's Updating Process in the Decentralized Identifier Systems," in *Proceedings of the 2021 International Conference on Information Networking*, ICOIN 2021, pp. 517–520, 2021.

[19] K.-H. Kim, S. Lim, D.-Y. Hwang, and K.-H. Kim, "Analysis on the Privacy of DID Service Properties in the DID Document," in *Proceedings of the 2021 International Conference on Information Networking*, ICOIN 2021, pp. 745–748, 2021.

[20] M. P. Bhattacharya, P. Zavarsky, and S. Butakov, "Enhancing the Security and Privacy of Self-Sovereign Identities on Hyperledger Indy Blockchain," in *Proceedings of the 2020 International Symposium on Networks, Computers and Communications*, ISNCC 2020, pp. 1–7, 2020.

[21] J. Steed and G. Shore, "Software Security Analysis - A Short Appreciation," *IFAC Proceedings Volumes*, vol. 16, no. 18, pp. 53–58, 1983.

[22] The Mitre Corporation, "Common Weakness Enumeration (CWE)." https://cwe.mitre.org, 2006. Accessed: 2022-08-22.

[23] "SERIES X: Data Networks, Open System Communications and Security – Cybersecurity information exchange – Vulnerability/state exchange – Common Weakness Enumeration," Recommendation ITU-T X.1524, International Telecommunication Union, Mar. 2012.

[24] The National Institute of Standards and Technology, "National Vulnerability Database (NVD)." https://nvd.nist.gov, 2005. Accessed: 2022-08-23.

[25] MythX, "Smart-Contract Weakness Classification (SWC)." https://swcregistry.io, 2020. Accessed: 2022-08-23.

[26] W. Zheng, J. Y. Cheng, X. Wu, R. Sun, and X. Wang, "Domain knowledge-based security bug reports prediction," *Knowledge-Based Systems*, vol. 241, p. 108293, 2022.

[27] E. Hariyanti, A. Djunaidy, and D. Siahaan, "Information security vulnerability prediction based on business process model using machine learning approach," *Computer & Security*, vol. 110, p. 102422, 2021.

[28] R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, "Predicting Vulnerable Software Components via Text Mining," *IEEE Transactions on Software Engineering*, vol. 40, no. 10, pp. 993–1006, 2014.

[29] M. Nadeem, E. B. Allen, and B. J. Williams, "A Method for Recommending Computer-Security Training for Software Developers: Leveraging the Power of Static Analysis Techniques and Vulnerability Repositories," in *Proceedings of the 2015 12th International Conference on*

Information Technology - New Generations, ITNG 2015, pp. 534–539, 2015.

[30] P. Hustinx, “Privacy by Design: Delivering the Promises,” *Identity in the Information Society*, vol. 3, no. 2, pp. 253–255, 2010.

[31] A. Cavoukian, “Privacy by design: The 7 foundational principles,” 2011.

[32] N. N. Kumar and R. Shyamasundar, “Realizing Purpose-Based Privacy Policies Succinctly via Information-Flow Labels,” in *Proceedings of the 2014 IEEE Fourth International Conference on Big Data and Cloud Computing*, pp. 753–760, 2014.

[33] R. Joshaghani, S. Black, E. Sherman, and H. Mehrpouyan, “Formal Specification and Verification of User-Centric Privacy Policies for Ubiquitous Systems,” in *Proceedings of the 23rd International Database Applications and Engineering Symposium*, IDEAS ’19, (Athens, Greece), 2019.

[34] L. Lesavre, P. Varin, P. Mell, M. Davidson, and J. Shook, “A Taxonomic Approach to Understanding Emerging Blockchain Identity Management Systems,” White Paper NIST CSWP 9, National Institute of Standards and Technology (NIST), Jan. 2020.

[35] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, “uPort: A Platform for Self-Sovereign Identity.” https://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf, 2016. Accessed: 2022-09-17.

[36] IBM Corporations, “IBM Verify Credentials: Documentation.” https://docs.info.verify-creds.com. Accessed: 2022-09-17.

[37] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” white paper, 2008.

[38] “Distributed Ledger Technologies & Blockchain : Technological Risks and Recommendations for the Financial Sector,” white paper, Commission de Surveillance du Secteur Financier, 2022.

[39] Object Management Group (OMG), “Meta-Object Facility (MOF) Specification, Version 2.0.” OMG Document Number formal/2006-01-01 (http://www.omg.org/spec/MOF/2.0), 2006.

[40] Object Management Group (OMG), "Unified Modeling Language (UML) Specification, Version 2.5.1." OMG Document Number formal/2017-12-05 (https://www.omg.org/spec/UML), 2017.

[41] The Mitre Corporation, "Common Vulnerability Exposure (CVE)." https://cve.mitre.org, 1999. Accessed: 2022-08-22.

[42] "Series X: Data Networks, Open System Communications and Security - Cyberspace security – Identity management: Entity Authentication Assurance Framework," Recommendation ITU-T X.1254, The Telecommunication Standardization Sector of International Telecommunication Union, 2020.

[43] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[44] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology behind Search*. USA: Addison-Wesley Publishing Company, 2nd ed., 2011.

[45] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[46] G. Paltoglou and M. Thelwall, "A Study of Information Retrieval Weighting Schemes for Sentiment Analysis," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL 2010, (Uppsala, Sweden), pp. 1386–1395, Association for Computational Linguistics, July 2010.

[47] Y. Zheng and D. X. Wang, "A survey of recommender systems with multi-objective optimization," *Neurocomputing*, vol. 474, pp. 141–153, 2022.

[48] T. Anwar and V. Uma, "CD-SPM: Cross-domain book recommendation using sequential pattern mining and rule mining," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 793–800, 2022.

[49] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, "Knowledge Graphs," *ACM Computer Survey*, vol. 54, July 2021.

[50] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A Survey on Knowledge Graphs: Representation, Acquisition, and Applications,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 494–514, 2022.

[51] “Peronal Information Protection Law of the Mainland,” law, The Standing Committee of the National People’s Congress, 2021.

[52] “Information Technology – Security Techniques – Information Security Management,” Standard ISO/IEC 27001:2013, International Organization of Standardization, 2013.

[53] D. Jackson, *Software Abstractions: Logic, Language, and Analysis*. The MIT Press, 2006.

[54] “Information Technology – Security Techniques – Privacy Framework,” Standard ISO/IEC 29100:2011, International Organization of Standardization, 2011.

[55] International Organization of Standardization, “List of International Standards.” https://www.iso.org. Accessed: 2022-08-22.

[56] D. Tofan, “Information Security Standards,” *Journal of Mobile, Embedded and Distributed Systems*, vol. 3, pp. 128–135, 2011.

[57] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, and V. R. Kebande, “A Review of Security Standards and Frameworks for IoT-Based Smart Environments,” *IEEE Access*, vol. 9, pp. 121975–121995, 2021.

[58] Wikipedia, “IT Security Standards.” https://en.wikipedia.org/wiki/IT_security_standards. Accessed: 2022-10-09.

[59] Wikipedia, “Cyber-Security Regulation.” https://en.wikipedia.org/wiki/Cyber-security_regulation. Accessed: 2022-10-09.

[60] Wikipedia, “Privacy Law.” https://en.wikipedia.org/wiki/Privacy_law. Accessed: 2022-10-09.

[61] Wikipedia, “Information Privacy Law.” https://en.wikipedia.org/wiki/Information_privacy_law. Accessed: 2022-10-09.

[62] Wikipedia, “Privacy Policy.” https://en.wikipedia.org/wiki/Privacy_policy. Accessed: 2022-10-09.

[63] C. Pattiyanon and T. Aoki, “Analysis and Enhancement of Self-sovereign Identity System Properties Compiling Standards and Regulations,” in *8th International Conference on Information Systems Security and Privacy*, pp. 133–144, 2022.

[64] Information Systems Audit and Control Association, *COBIT 2019 Framework: Governance and Management Objectives.* ISACA, 2018.

[65] “Payment Card Industry (PCI) Data Security Standard (DSS) – Requirements and Security Assurance Procedure 3.2.1,” Standard PCI-DSS, Payment Card Industry Security Standard Council, New York, NY, USA, 2018.

[66] “Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals With Regard to the Processing of Personal Data and on the Free Movement of Such Data,” Directive 95/46/EC, European Council, London, U.K., Oct. 1995.

[67] “Health Insurance Portability Accountability Act of 1996,” Law HIPAA, Centers of Medicare and Medicaid Services, MI, USA, 1996.

[68] “Information Security – Lightweight Cryptography,” Standards ISO/IEC 29192-2:2019, International Organization of Standardization, 2019.

[69] “Information Technology – Security Techniques – Prime Number Generator,” Standards ISO/IEC 18032:2005, International Organization of Standardization, 2005.

[70] “Personal Data Act,” Law PDA 1998:204, Swedish Data Protection Authority, 1998.

[71] “Information Technology – Security Techniques – Digital Signature Schemes Giving Message Recovery,” Standard ISO/IEC 9796-2:2010, International Organization of Standardization, 2010.

[72] “Information Technology – Security Techniques – Entity Authentication,” Standard ISO/IEC 9798-2:2008, International Organization of Standardization, 2008.

[73] “Security Techniques – Hash Functions,” Standard ISO/IEC 10118-1:2016, International Organization of Standardization, 2016.

[74] “Information Technology – Security Techniques – Key Management,” Standard ISO/IEC 11770-1:2010, International Organization of Standardization, 2010.

[75] “Information Security – Non-Repudiation,” Standard ISO/IEC 13888-1:2020, International Organization of Standardization, 2020.

[76] “Information Technology – Security Techniques – Time-Stamping Services,” Standard ISO/IEC 18014-1:2008, International Organization of Standardization, 2008.

[77] “Information Technology – Security Techniques – Random Bit Generation,” Standard ISO/IEC 18031:2005, International Organization of Standardization, 2005.

[78] “Information Security – Authenticated Encryption,” Standard ISO/IEC 19772:2020, International Organization of Standardization, 2020.

[79] “Information Technology – Security Techniques – Codes of Practices For Information Security Controls,” Standard ISO/IEC 27002:2013, International Organization of Standardization, 2013.

[80] “Security Techniques – Extension to ISO/IEC 27001 and ISO/IEC 27002 for Privacy Information Management–Requirements and Guidelines,” Standard ISO/IEC 27701:2019, International Organization of Standardization, 2019.

[81] “Security for Industrial Automation and Control Systems – System Security Requirements and Security Levels,” Standard IEC 62443-3-3:2013, International Electrotechnical Commission, 2013.

[82] “Cyber Security Standards,” Standards CIP-002-1 Through CIP-009-1, North American Electric Reliability Corporation, Atlanta, GA, USA, 2006.

[83] “NIST Special Publication 800-12 Revision 1: An Introduction to Information Security,” Standard NIST SP800-12 R1, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[84] “Cyber Essentials: Requirements for IT Infrastructure,” standard, National Cyber Security Centre, London, U.K., 2021.

[85] “OECD Privacy Framework,” standard, Organization for Economic Co-Operation and Development, Paris, France, 2013.

[86] "UN Personal Data Protection and Privacy Principles," Regulation UN PDPPP, United Nation High-Level Committee Management (HLCM), Bonn, Germany, 2018.

[87] "The Information Technology Act, 2000 (No. 20 200)," Law 20 200, Legislative Department, Gazette India, Ministry of Law, Justice Company Affairs, New Delhi, India, 2000.

[88] "The Personal Data Protection Code of Practice," law, Parliament of Malaysia, Malaysia, 1998.

[89] "Data Privacy Act 2012," Law Republic Act 10173, National Privacy Commission, Passay, Philippines, 2012.

[90] "Federal Law No.152-FZ of July 27, 2006 On Personal Data," Federal Law No.152-FZ, Federal Council, Russia, 2006.

[91] "Personal Data Protection Act," law, National Development Council, Taiwan, 2015.

[92] "Children's Online Privacy Protection Act of 1998," law, Federal Trade Commission, Washington, DC, USA, 1998.

[93] "Personal Information Protection Law of the Mainland," law, The Standing Committee of the National People's Congress, Beijing, China, 2021.

[94] "OWASP Application Security Verification Standard 4.0.2," Standard OWASP ASVS 4.0.2, Open Web Application Security Project (OWASP) Foundation, Annapolis, MD, USA, 2020.

[95] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Computer Science Review*, vol. 30, pp. 80–86, 2018.

[96] A. Grüner, A. Mühle, T. Gayvoronskaya, and C. Meinel, "A Quantifiable Trust Model for Blockchain-Based Identity Management," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1475–1482, 2018.

[97] H. Chen, G. Cao, J. Chen, and J. Ding, "A Practical Framework for Evaluating the Quality of Knowledge Graph," in *Knowledge Graph*

and Semantic Computing: Knowledge Computing and Language Understanding (X. Zhu, B. Qin, X. Zhu, M. Liu, and L. Qian, eds.), (Singapore), pp. 111–122, Springer Singapore, 2019.

[98] "Personal Data Protection Act (PDPA), B.E. 2562 (2019)," law, Cabinet Parliament of Thailand, Royal Thai Government Gazette, Bangkok, Thailand, 2019.

[99] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K.-K. Raymond Choo, "Blockchain-based identity management systems: A review," *Journal of Network and Computer Applications*, vol. 166, no. 102731, 2020.

[100] A.-E. Panait, R. F. Olimid, and A. Stefanescu, "Analysis of UPort Open, an Identity Management Blockchain-Based Solution," in *Trust, Privacy and Security in Digital Business: 17th International Conference, TrustBus 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings*, (Berlin, Heidelberg), pp. 3–13, Springer-Verlag, 2020.

[101] A. Tobin and D. Reed, "The Inevitable Rise of Self-Sovereign Identity: A white paper from the Sovrin Foundation," white paper, Sovrin.org, 2017. https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf, Accessed: 2022-09-17.

[102] J.-b. Gao, B.-w. Zhang, X.-h. Chen, and Z. Luo, "Ontology-based model of network and computer attacks for security assessment," *Journal of Shanghai Jiaotong University (Science)*, vol. 18, no. 5, pp. 554–562, 2013.

[103] L. Lu, S. Huang, and Z. Ren, "A Weakness-Based Attack Pattern Modeling and Relational Analysis Method," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, pp. 1024–1028, 2014.

[104] S. A. Mokhov, J. Paquet, and M. Debbabi, "The Use of NLP Techniques in Static Code Analysis to Detect Weaknesses and Vulnerabilities," in *Advances in Artificial Intelligence* (M. Sokolova and P. van Beek, eds.), (Cham), pp. 326–332, Springer International Publishing, 2014.

[105] F. Sun, L. Xu, and Z. Su, "Detecting Logic Vulnerabilities in E-commerce Applications," in *Proceedings of the 21st Annual Network*

and Distributed System System Security Symposium, NDSS 2019, (San Diego, California, USA), The Internet Society, February 2014.

[106] Y. Son, Y. Lee, and S. Oh, "A Software Weakness Analysis Technique for Secure Software," *Advanced Science and Technology Letters*, vol. 3, pp. 5–8, 2015.

[107] Y. Yu, V. N. Franqueira, T. Than Tun, R. J. Wieringa, and B. Nuseibeh, "Automated analysis of security requirements through risk-based argumentation," *Journal of Systems and Software*, vol. 106, pp. 102–116, 2015.

[108] B. Kim, J.-h. Song, J.-P. Park, and M.-s. Jun, "Design of Exploitable Automatic Verification System for Secure Open Source Software," in *Advances in Computer Science and Ubiquitous Computing* (D.-S. Park, H.-C. Chao, Y.-S. Jeong, and J. J. J. H. Park, eds.), (Singapore), pp. 275–281, Springer Singapore, 2015.

[109] G. Stergiopoulos, P. Katsaros, and D. Gritzalis, "Execution Path Classification for Vulnerability Analysis and Detection," in *E-Business and Telecommunications* (M. S. Obaidat and P. Lorenz, eds.), (Cham), pp. 293–317, Springer International Publishing, 2016.

[110] C. Wang, Y. Jiang, X. Zhao, X. Song, M. Gu, and J. Sun, "Weak-Assert: A Weakness-Oriented Assertion Recommendation Toolkit for Program Analysis," in *Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, pp. 69–72, 2018.

[111] L. Han, M. Zhou, and C. Fu, "An Static Propositional Function Model to Detect Software Vulnerability," in *Security and Privacy in New Computing Environments* (J. Li, Z. Liu, and H. Peng, eds.), (Cham), pp. 564–575, Springer International Publishing, 2019.

[112] A. Brazhuk, "Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries," *International Journal of Open Information Technologies*, vol. 7, no. 3, pp. 38–41, 2019.

[113] M. Byun, Y. Lee, and J.-Y. Choi, "Analysis of software weakness detection of CBMC based on CWE," in *Proceedings of the 2020 22nd International Conference on Advanced Communication Technology*, ICACT 2020, pp. 171–175, 2020.

[114] Z. Liu, P. Qian, X. Wang, Y. Zhuang, L. Qiu, and X. Wang, “Combining Graph Neural Networks with Expert Knowledge for Smart Contract Vulnerability Detection,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2021.

[115] C. Skandylas, L. Zhou, N. Khakpour, and S. Roe in *Proceedings of the 2021 IEEE/ACM 2nd International Workshop on Engineering and Cybersecurity of Critical Systems, series =.*

[116] N. A. Noor Aidee, M. G. M. Johar, M. H. Alkawaz, A. Iqbal Hajamydeen, and M. S. Hamoud Al-Tamimi, “Vulnerability Assessment on Ethereum Based Smart Contract Applications,” in *Proceedings of the 2021 IEEE International Conference on Automatic Control & Intelligent Systems*, I2CACIS 2021, pp. 13–18, 2021.

[117] S. Jeon and H. K. Kim, “AutoVAS: An automated vulnerability analysis system with a deep learning approach,” *Computers & Security*, vol. 106, p. 102308, 2021.

[118] R. Agarwal, T. Thapliyal, and S. K. Shukla, “Vulnerability and Transaction Behavior Based Detection of Malicious Smart Contracts,” in *Cyberspace Safety and Security* (W. Meng and M. Conti, eds.), (Cham), pp. 79–96, Springer International Publishing, 2022.

[119] A. Satybaldy, M. Nowostawski, and J. Ellingsen, “Self-Sovereign Identity Systems: Evaluation Framework,” *IFIP Advances in Information and Communication Technology*, vol. 576, pp. 447–461, 2020.

[120] R. Soltani, U. T. Nguyen, and A. An, “A Survey of Self-Sovereign Identity Ecosystem,” *Security and Communication Networks*, vol. 2021, no. 8873429, 2021.

[121] L. Stockburger, G. Kokosioulis, A. Mukkamala, R. R. Mukkamala, and M. Avital, “Blockchain-enabled decentralized identity management: The case of self-sovereign identity in public transportation,” *Blockchain: Research and Applications*, vol. 2, no. 2, p. 100014, 2021.

[122] M. Shuaib, N. H. Hassan, S. Usman, S. Alam, S. Bhatia, D. Koundal, A. Mashat, and A. Belay, “Identity Model for Blockchain-Based Land Registry System: A Comparison,” *Wireless Communications and Mobile Computing: Security Threats and Challenges in Future Mobile Communication Systems*, vol. 2022, no. 5670714, 2022.

[123] M. Shuaib, N. H. Hassan, S. Usman, S. Alam, S. Bhatia, A. Mashat, A. Kumar, and M. Kumar, "Self-Sovereign Identity Solution for Blockchain-Based Land Registry System: A Comparison," *Mobile Information Systems*, vol. 2022, no. 8930472, 2022.

[124] S. Farheen, N. A. Day, A. Vakili, and A. Abbassi, "Transitive-closure-based model checking (TCMC) in Alloy," *Software and Systems Modeling*, vol. 19, pp. 721–740, 2020.

[125] J. B. Bernabe, M. David, R. T. Moreno, J. P. Cordero, S. Bahloul, and A. Skarmeta, "ARIES: Evaluation of a reliable and privacy-preserving European identity management framework," *Future Generation Computer Systems*, vol. 102, pp. 409–425, 2020.

[126] A. Grüner, A. Mühle, and C. Meinel, "Using Probabilistic Attribute Aggregation for Increasing Trust in Attribute Assurance," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence*, SSCI 2019, pp. 633–640, 2019.

[127] M. Luecking, C. Fries, R. Lamberti, and W. Stork, "Decentralized Identity and Trust Management Framework for Internet of Things," in *Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency*, ICBC 2020, pp. 1–9, 2020.

[128] D. Kirupanithi and A. Antonidoss, "Self-Sovereign Identity creation on Blockchain using Identity based Encryption," in *Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems*, ICICCS 2021, pp. 299–304, 2021.

Appendix A

The Improved SSI System Properties in CSSPS

In this chapter, the complete definitions of the improved SSI system properties in CSSPS resulted from the improvement procedure is provided. The implementation of the SSI system properties can use the following definition as constraints for the desired SSI management system.

Table A.1: Definition of the improved SSI system properties in CSSPS (1).

SSI System Property and Definition
IP1. Existence. An SSI system must enable individuals to encode their personal characteristics in order for their information to be shown in the digital domain. An SSI system must make identities open and accessible to the public. An SSI system must associate individual users' identities with their unique identifiers. An SSI system should enable the correlation of confidential and biometric data with identities and credentials.
IP2. Sovereignty. As ultimate authority, an SSI system must enable users to control and manage their identities, authenticators (i.e., identifiers), digital credentials, and certificates. An SSI system must grant all users the required permissions for referring to, updating, removing, or even hiding their identities. An SSI system must prevent other individuals, organizations, or governments from controlling the identities of users. Other entities should be able to make claims against a user through an SSI system, but they should not be able to act as the central authority.

Table A.2: Definition of the improved SSI system properties in CSSPS (2).

SSI System Property and Definition
IP3. Single Source. An SSI system should enable users to act as the primary source of identity, eventually owning and controlling their own identities. An SSI system must enable users to make self-asserted claims and/or accumulate third-party asserted claims by leveraging their identities and distributing them as needed. An SSI system must have a mechanism to prevent that third party from collaborating with other parties to exchange identities without the users' knowledge. An SSI system must enable users to delegate work to an autonomous agent that is under their control. An SSI system must store and maintain identities and other important data on a storage device that is typically owned or controlled by the relevant user. An SSI system must not maintain all of its users' identities in a centralized repository or on a distributed ledger/blockchain managed by another party.
IP4. Standard. An SSI system must use data and communication formats that are based on open standards for identities and credentials.
IP5. Cost Free. An SSI system should provide identities and services to everyone for free or at a very low cost, without hidden costs, licensing fees, or other financial charges. As a minimum, an SSI system may incur costs associated with other resources and implementation.
IP6. Decentralized. An SSI system should not be used to centrally register and administer identities by a proprietary entity. An SSI system should be used to register and administer identities via a decentralized infrastructure that is mostly operated by the public.
IP7. Verifiability. An SSI system should enable the verification of identities and credentials in a manner comparable to that of a physical credential. An SSI system should enable issuers to sign and secure identities and credentials cryptographically. An SSI system should be capable of verifying identities and credentials without involving the corresponding issuer.
IP8. Scalability. An SSI system should enable users to have many, decomposable, extendable, and gradual identities. An SSI system should be scalable at any demand, i.e., it should be necessary for a large number of users, organizations, and entities.
IP9. Accessibility. An SSI system should be simple to use and accessible to the widest potential audience.
IP10. Sustainability. An SSI system should be environmentally, economically, technically, and socially sustainable for the long term.
IP11. Access Control. An SSI system must provide users unrestricted access to their identities. An SSI system must allow users to retrieve, update, share, hide, or delete their identities. An SSI system should accept support and assistance from identity providers or organizations but not provide them ownership of the underlying identity. An SSI system must not conceal users' identity or rely on gatekeepers. An SSI system should restrict access to users' identities and interactions with them by identity providers or organizations based on their security level. An SSI system must not provide equal access of a user to other users' data. An SSI system must restrict access based on a fine-grained access control mechanism administered by users and allowing access only to those with valid credentials. An SSI system must remove identities and credentials on devices that are not controlled by users. An SSI system should prompt users to review their identity access to ensure that it is correct and that it corresponds to the user's needs as well as the appropriate roles and responsibilities for access are set. An SSI system should provide procedure to ensure that modification, suspension, and termination of access to users' identities is accomplished within 24 hours of a change in access status.

Table A.3: Definition of the improved SSI system properties in CSSPS (3).

SSI System Property and Definition
IP12. Transparency. As appropriate and whenever possible, an SSI system, including its subsystems, protocols, and algorithms, must be transparent enough for every involved entity. An SSI system must be open to users in order for them to be aware of their identities and interactions. An SSI system must make it simple for users to retrieve traces of identity interactions. An SSI system, including its subsystems, protocols, and algorithms, must be free, open-source, and as decoupled from any particular architecture or proprietorship as possible in order for their internal mechanisms to be examined by anyone. An SSI system should publish cryptographic proofs of ownership of decentralized identifiers, cryptographic proofs of ownership of verifiable claims, and cryptographic proofs of validity of verifiable claims in a public decentralized network.
IP13. Persistence. An SSI system must persist identities forever, or at least as long as they are required by the user and necessary for the purpose fulfilment. An SSI system should maintain identities until they become obsolete due to the introduction of newer identity systems or until the asserted authority ceases to exist, and then remove them. An SSI system must discard or permanently destroy identities if the corresponding user wishes to modify, revoke, or delete them over time or after the purposes for which they were created have been fulfilled. An SSI system should not tie identities and claims forever. An SSI system should collect sensitive data and identities in a discriminatory manner. They should carefully consider the type and quantity of such data that will be required to accomplish a particular task prior to collecting such data. An SSI system must enable users to erase identities, credentials (i.e., the verifiable claim), and certificates when the persistence period is end. An SSI system must enable users to erase identities, credentials (i.e., the verifiable claim), and certificates when the corresponding users have withdrawn their consent. An SSI system must enable users to erase identities, credentials (i.e., the verifiable claim), and certificates when the SSI system have violated the laws, regulations or agreements when processing identities.
IP14. Portability. An SSI system must enable the transport of identities, digital wallets, and identity-related services. When a previous medium or platform vanishes from the landscape for a variety of reasons, an SSI system must enable users to transfer their identities to a new medium or platform. An SSI system must restrict the ability of third parties or the new medium or platform to control identities and maintain user control over theirs. While transferring to a new medium or platform, an SSI system's identity must remain portable for an extended period of time.
IP15. Interoperability. Without risking user control, an SSI system should be able to communicate with another identity or service system at any scale. For a period of time, an SSI system should be able to be backwards compatible with legacy identity systems.
IP16. Consent. An SSI system must enable users to consent to the use of their identities and to control when and to which entity they wish to release their identities for whatever purpose. An SSI system must release identities to a third party lawfully and fairly, and only with the consent of the corresponding user. An SSI system must enable users to consent to the presentation of their claims by others. An SSI system must enable the deliberate and well-understood presentation of consent. An SSI system must allow users to revoke their consent to the processing of their identities. Unless the corresponding user later consents, an SSI system must restrict the processing of identities beyond the initial consent given by the corresponding user.
IP17. Data Minimization. An SSI system should minimize identity disclosures by requiring only the data necessary to complete the task at hand. In order to share identities, an SSI system must employ selective disclosure, range proof, and other zero-knowledge techniques.
IP18. Protection. An SSI system should safeguard the freedom and rights of all users under all circumstances. In the event of a conflict between users and the identity network, an SSI system should prioritize the protection of users' rights.

Table A.4: Definition of the improved SSI system properties in CSSPS (4).

SSI System Property and Definition
IP19. Data Authentication. A decentralized SSI system must provide authentication for accessing identities that are censorship- and force-resistant. For each interaction involving decentralized identifiers, an SSI system must authorize and properly authenticate the corresponding user. An SSI system should have formal registration and de-registration procedures for accessing a digital wallet, as well as for granting and revoking access to all services.
IP20. Physical Authentication and Protection. An SSI system must authenticate external connections to private peers, equipment, and network components. Organizations must restrict access to an SSI system's private peers, equipment, and associated operating environments to authorized individuals. Organizations must ensure that private-peer devices and software are not vulnerable to known security flaws.
IP21. Data Confidentiality. An SSI system must safeguard identities using cutting-edge cryptographic mechanisms and securely store them. An SSI system must employ cryptographic modules that are unpredictable, unbiased, and self-contained, as well as secure in both forward and backward secrecy.
IP22. Data Recovery. An SSI system must enable users to successfully recover and retrieve their identities, claims, credentials, and certificates in the event of a lost key, wallet, device, or complete identity loss. An SSI system should not rely on artifacts that can be lost, stolen, destroyed, or falsified to store identities, credentials, or certificates. An SSI system should include a mechanism for off-site backups of identities and other sensitive data. An SSI system must validate the data integrity of identities restored from backups.
IP23. Data Availability. An SSI system must ensure that identities are readily available and accessible from a variety of platforms on a consistent basis or as requested by users. An SSI system should be accessible to all, regardless of their ethnic origin, gender, socioeconomic status, or language.
IP24. Communication Security. At all times, an SSI system must transmit identities and other sensitive data using robust algorithms and ciphers over a secure communication channel that does not disclose them. An SSI system must employ any mechanism necessary to ensure that identities and other sensitive data are transmitted to their intended recipients. An SSI system should ensure that the internet is used to access only safe and necessary network components.
IP25. Data Integrity. An SSI system must use a digital signature to sign and authenticate messages pertaining to identities. An SSI system should employ unforgeable data time-stamping. An SSI system should verify the integrity of all messages received. With the message authentication mechanism, an SSI system should use collision-resistant hash functions.
IP26. Data Authorization. An SSI system should restrict and control use and privilege allocation. All users' roles and responsibilities must be clearly defined in an SSI system.
IP27. Accountability. An SSI system must maintain sufficient log information to track system activities related to identity processing. An SSI system should provide evidence of compliance measures of information security and privacy to each involving entity.
IP28. Non-Repudiation. An SSI system must include a mechanism to provide users with irrefutable evidence to refute a false rejection or refusal of an obligation.
IP29. Data Validation and Sanitization. An SSI system should validate data input, whether it is strongly typed, validated, range or length checked, or at worst, sanitized or filtered, to ensure it is correct and appropriate. An SSI system should validate data output to ensure that the processing of data is correct and appropriate.
IP30. Error Handling. An SSI system must check and validate to detect and appropriately handle data corruption caused by processing errors or deliberate acts.

Table A.5: Definition of the improved SSI system properties in CSSPS (5).

SSI System Property and Definition
IP31. Key Protection. An SSI system must secure key generation (public and private keys) and their use in cryptographic modules. An SSI system must enable users to securely share keys with trusted third parties.
IP32. Malware Protection. To prevent harmful code from causing damage or gaining access to identities or other sensitive data, an SSI system must restrict the execution of known malware and untrusted software in private peer systems. An SSI system must secure and properly handle malicious activities in order to avoid interfering with the processing of identities.
IP33. Password Security. An SSI system should enable users to manage and control their passwords through a formal management process, such as password aging or suspension.
IP34. Configuration Security. An SSI system should control and protect both physical and logical access to diagnostic and configuration ports. An SSI system must manage its business processes in a sequential and orderly way. When receiving untrusted files as evidence of identity, an SSI system should handle them with well treatment. By default, an SSI system's subsystems and infrastructure should be configured securely.
IP35. Session Security. After a defined period of inactivity, an SSI system must control and deactivate inactive sessions. An SSI system should limit connection times to a specified time period. An SSI system must ensure that each individual's session is unique and cannot be guessed or shared.
IP36. Data Classification. An SSI system should classify the security level of each data.
IP37. Fairness and Lawfulness. An SSI system must process identities and other sensitive data fairly and lawfully.
IP38. Purpose Specification and Limitation. An SSI system must specify the purposes for data collection and processing and notify users no later than at the time data is collected. An SSI system must limit any subsequent processes that are incompatible with the original purposes.
IP39. Use and Disclosure Limitation. An SSI system must process only those identities and other sensitive data that are necessary and relevant for the processing purposes. When an SSI system releases, communicates, or shares an individual's identity with a third party, whether intentionally or unintentionally, the disclosure must be recorded.
IP40. Data Accuracy and Quality. An SSI system should validate whether identities, credentials, and certificate are relevant to the purposes for which they are to be used and that they are accurate, complete, and up to date to the extent necessary for those purposes. Individuals who discover that their personal information is inaccurate or incomplete must be able to request corrections and supplementation through an SSI system.
IP41. Notification. Prior to or as soon as reasonably practicable after collecting and processing users' identities and other sensitive data, an SSI system must notify and inform users about their privacy notice. An SSI system must inform users of the necessity of disclosing their identities. An SSI system must take immediate corrective action and notify users in the event of personal information leakage, tampering with, or loss, or when such events may have occurred. The collection of children's identities and other personal information must be informed by an SSI system, which must obtain parental consent as the children's agent.
IP42. Individual Participation. An SSI system must inform users whether it holds data about them within a reasonable timeframe; at a reasonable fee, if any; in a reasonable manner; and in a format that is readily understandable to them. An SSI system must provide users with reasons for denials of their requests and the right to challenge such denials. An SSI system must enable users to challenge data that is personally identifiable, and if the challenge is successful, the data must be deleted, rectified, completed, or amended regarding the challenge.