Compliance SSI System Property Set to Laws, Regulations, and Technical Standards

CHARNON PATTIYANON1 AND TOSHIKI AOKI2
1Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology (JAIST), Nomi 923-1211, Japan
2School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Nomi 923-1211, Japan
Corresponding author: Charnon Pattiyanon (s1920431@jaist.ac.jp)

ABSTRACT Digital identities, including names and age, provide modern information systems with valuable data. Identity management is a security feature that enables users to manage and utilize their digital identities when interacting with online services. A self-sovereign identity (SSI) system is a cutting-edge identity model that uses blockchain technology to foster peer-to-peer trust in service authentication and authorization. SSI systems typically adhere to a set of functional- and quality-related principles and properties. However, we discovered that the current information security and privacy principles and properties of SSI systems are not compliant with the laws, regulations, and technical standards. A compliance set of principles and properties must be established to support the implementation of SSI systems and to improve security and privacy. In this article, we propose CSSPS, a new compliance SSI system property set that expands upon the missing security and privacy controls specified in applicable laws, regulations, and technical standards. We used systematic comparative analysis and systematic review to identify inconsistent content from a vast collection of applicable sources, and then used them to extend the current properties of the SSI system. The proposed CSSPS increases the consistency of security and privacy controls, and is applicable in accordance with the functionality of real SSI systems, as determined by a qualitative evaluation. The proposed CSSPS contributes to SSI system implementation by facilitating correct implementation, while adhering to applicable information security and privacy sources. In addition, the proposed CSSPS can indirectly enhance the security and privacy of SSI systems.

INDEX TERMS Information security and privacy, legislative compliance, self-sovereign identity, standard compliance, system property.

I. INTRODUCTION
A digital identity, such as a user’s name, email address, and age, is the online persona of one or more subjects that allows them to access offline and online services. Digital identity is commonly regarded as personally identifiable information (PII) and requires special treatment in modern information systems. An identity management system is a security architecture for processing digital identities across the identity life cycle management, including identity registration, verification, issuance, authentication, and authorization.

Conventionally, identity management systems are centralized, which requires users to provide their digital identities to central authorities and rely on their data handling mechanisms. However, centralized identity management systems require users to incur administrative costs in order to manage their digital identities at every location they provide. In order to consolidate the administration of digital identities under a single identity provider, federated and user-centric identity management systems were developed. Disputes also arose regarding the security and privacy of identity providers’ digital identities. With the emergence of decentralized systems and distributed ledger technology, decentralized identity management systems have been introduced and have gained prominence in the field of information security and privacy.

A self-sovereign identity (SSI) system is an identity management system that authenticates and authorizes services using a claim-based trust mechanism and blockchain...
technology. The SSI system enables users to create zero-knowledge claims of their PII. An issuer must confirm the claim to convince a verifier that the user genuinely owns their PII. The immutable proof schema publicly available on the blockchain establishes trust between issuers and verifiers. The SSI system was implemented to reduce administrative overhead while ensuring that users can independently control their PII and credentials [1].

Due to the SSI system’s novelties, its fundamental concepts need to be comprehended in order to implement it correctly. Researchers and practitioners developed SSI principles [1], [2], [3] and system properties [4]. A principle is a set of sentences describing the constraints imposed on the SSI system and its environment (e.g., users, organizations). For instance, the “existence” principle has been defined [1] as follows:

“Existence - Users must have an independent existence. Any SSI is ultimately based on the ineffable ‘I’ at the heart of identity. An SSI simply makes public and accessible some limited aspects of the ‘I’ that already exist.”

However, principles have a broad definition and must be interpreted in a systemic context before being evaluated. Compared to a principle, a system property is expressed in a fashion that can be evaluated directly on the SSI system. For instance, the “consent” property has been defined [4] as follows:

“Consent - Every piece of identity data must be released to a third party only after the corresponding user has consented.…”

We can determine whether system properties are constrained in the defined manner through the SSI system’s functions when evaluating system properties. This article will focus on only system properties that are evaluable via the SSI system’s functions.

For user confidence to be developed, effective implementation is necessary. As a system that manipulates PII, the SSI system should adhere to the laws, regulations, and technical standards. Infractions of applicable laws and regulations may result in severe litigation against the SSI system’s implementers.

Law is a written document that is legally binding and is enacted by parliament. It can regulate, authorize, forbid, provide (funds), sanction, award, proclaim, and restrict. A regulation is delegated legislation that manages complex systems by enforcing prescribed rules or trends. A standard is an established norm or requirement for a repeated activity that pertains to the consistent and repeatable application of rules, conditions, guidelines, or characteristics for products or associated processes and manufacturing procedures and related management system practices. Since laws, regulations, and standards can be used as a source of information for auditing procedures, they will be referred to as source documents in this article.

In most cases, source documents can be transformed into a checklist of evaluable criteria for product owners or organizations. This checklist includes controls or requirements for evaluating the targeting subjects’ compliance. In this article, elements of the checklist will be referred to as controls to keep things simple. A control is a set of sentences describing endorsed tasks or conditions that the subjects undergoing evaluation must do or adhere to. For instance, the general data protection regulation (GDPR) [5] defines the “purpose limitation” control as follows:

“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;”

The auditing procedure establishes the uniformity of the evaluating subjects and controls. To persuade auditors, the evaluating subjects must prove how the endorsed tasks or conditions are met. If subjects were reported as inconsistent, it would take a considerable effort to correct and re-evaluate them. It would be advantageous if the subjects were consistent in the first place.

We observed that the SSI system’s properties are comparable to source document controls in that they are all defined for evaluating the desired SSI system implementation. As a result, while aligning the SSI system implementation with the source documents, we can check for consistency between property and control definitions, assuming the implementation appropriately possesses the properties. We consider that the consistency between system properties and controls is evidenced by the similarity of sentences in a system property definition to the endorsed tasks or conditions of the control. Currently, certain system properties provide a limited degree of consistency with controls in source documents. For example, while the consent property prohibits PII disclosure without the user’s consent, it is not fully consistent with GDPR article 5.1. (b) due to the lack of a restriction on further processing. We believe this is an opportunity to enhance the SSI system’s security and privacy through compliance.

This opportunity motivates researchers to devise new proposals for system properties to enhance the SSI system’s security and privacy. Allen [1] was the first to propose ten basic SSI principles restricting the SSI system’s core concepts. His proposal contained no constraints on security or privacy other than those outlined by the core concepts. Ferdous et al. [4] derived system properties from Allen’s basic SSI principles. They began incorporating security and privacy features into the properties without referencing source documents. López [2] extended 16 SSI principles to Allen’s list, aligning them with the criteria of the digital identity model. Finally, Naik and Jenkins [3] is the most recent proposal to reinforce Allen’s principles through GDPR compliance. Their proposal, however, limits the scope of data privacy to a single source document. We believe that the properties should be universally applicable across all scopes and as consistent as possible with a large number of source documents.
This article proposes a new SSI system property set, entitled “a compliance SSI system property set (CSSPS),” expanded from existing sets to be more consistent with relevant source documents. By ensuring the consistency of as many source documents as feasible, the CSSPS is designed to cover all tasks and capabilities that SSI systems must have to ensure the security and privacy of users. The huge number of reviewed source documents (i.e., more than 200) allows us to deduce a set of shared controls defined similarly across source documents. These shared controls serve as a reference point for comparing existing system properties and expanding them wherever they are inconsistent. The CSSPS contains revised, expanded, and new system properties enhanced to the maximum extent possible to be consistent with source documents. The main contributions of this article are summarized as follows:

1) We associate SSI principles to the existing SSI system properties and consolidate them into a single list of SSI system properties. This consolidation is beneficial to implementors as it makes them realize the most recent SSI system properties with minimizing repetition.

2) We derive shared controls that are commonly defined in source documents. The shared controls represent common aspects valid universally across multiple source documents.

3) We expand the existing SSI system property sets by including shared control parts into property definitions. This expansion contributes to implementors to develop SSI systems for the CSSPS while having a high chance of complying with most source documents.

This article is a revised and expanded version of a conference paper [6] we presented. By applying the systematic review approach, we were able to expand the scope of our previous work from only five to over 200 source documents. Meanwhile, this article expands the universality of the outcome (i.e., the CSSPS) by incorporating shared controls rather than all individual controls.

The remainder of this article is divided into the following sections: Section II outlines the SSI system’s terminology and features; Section III summarizes our approach. Section IV summarizes the consolidation of system properties; Section V describes our review of shared controls; Section VI describes our method for increasing consistency; Section VII summarizes our evaluation; Sections VIII and IX discuss the findings and compare previous work, and Section X concludes this article.

II. PRELIMINARIES ON SSI-SPECIFIC TERMINOLOGY, FEATURES, PRINCIPLES, AND PROPERTIES

The SSI system guarantees its users complete sovereignty in the absence of central government meddling. The SSI system is based on two industry-standard data models: decentralized identifier [7] and verifiable credential [8]. The SSI system becomes tailored to certain terminologies and characteristics by using these models. It is required to become familiar with the following terminology to comprehend the SSI system properties:

- **A subject** is any entity in which claims are made. Human beings, animals, and objects are examples of possible subjects. Often, the subject can take the holder role, but this is not always the case. For instance, a parent (the holder) may own a child’s claims (the subject).

- **A holder** is an entity role that holds one or more subject’s claims and presents them as credentials to verifiers for service authentication and authorization. A holder must be the role that has complete sovereignty over their PII located in a locally-owned device.

- **An issuer** is a type of entity role that validates claims about subjects and generates valid credentials in verifiable claims based on blockchain-based proof schemas.

- **A verifier** is an entity role that requires holders to present valid credentials (i.e., verifiable claims) before providing services to them.

- **A claim** is a declaration of the identity of a subject. It is a statement or structure that attests to the authenticity of a particular identity attribute. Using a zero-knowledge proof, the claim may imply the identity attribute so that it can convince verifiers without revealing actual information. Issuers must validate the claim in order for it to be a valid credential.

- **A decentralized identifier (DID)** is a unique identifier that combines the public/private key infrastructure (PKI) key exchange mechanism with blockchain technology. The DID is used as an address to resolve public keys contained in a DID document that has been published on the blockchain. Entities distribute DIDs in order to establish trust relationships.

This terminology acquaints readers with the SSI system and its properties. Certain alternate terminology for technology and implementation is used in various publications, but the meaning should remain the same.

The terminology establishes a conceptual foundation for the SSI system analogous to the data model for verifiable credentials. In Fig. 1, we display the high-level data communication among components to provide a general perspective of the SSI system. As can be seen, three distinct roles are being played as system components that store and communicate
data objects. It is a mechanism by which a holder can create a claim attesting to a subject’s identity attributes (for example, the subject being over the age of 18) and submit it to an issuer for validation. If the claim is genuine, the issuer will transform it into a verifiable claim using an on-chain proof schema and metadata. The holder will receive the verifiable claim, which will be provided to the verifiers for authentication and authorization of the service. The validity of a claim is indirectly validated using the blockchain’s proof schema and metadata. The SSI system’s key requirement is that issuers and verifiers do not communicate directly with one another. Understanding SSI-specific terminology and high-level data communication are sufficient for comprehending principles and properties of the SSI system. We will avoid discussing more advanced approaches or features.

In recent years, numerous principles and properties of the SSI systems have been proposed. Allen [1] proposed the first and most-cited set of ten principles for governing SSI systems based on the SSI system concept, as illustrated in Fig. 2. López [2] proposed a second set of 16 principles for SSI systems to comply with the digital identity model by aligning it with Allen’s principles, as shown in Fig. 3. Then, Naik and Jenkins [3] attempted to increase the privacy protection of the principles by making them GDPR-compliant. As depicted in Figures 4 and 5, they proposed a new set of governing privacy-protecting principles.

Ferdous et al. [4] attempted to convert Allen’s governing principles of SSI systems into a set of system properties that characterize the implementation of an SSI system. As shown in Fig. 6, they proposed five groups of 17 system properties.
In order to compare and contrast them with the applicable laws, regulations, and technical standards, this article provides these four proposals of SSI system principles and properties as inputs for our analysis. We do not consider the latter or other proposals to be within our purview.

III. APPROACH OVERVIEW

To propose the CSSPS, we employ a property enhancement approach that includes a systematic review of source documents and comparing present system properties to shared controls, as seen in Fig. 7. The approach is divided into property consolidation, shared control derivation, and property enhancement.

The property consolidation phase intends to develop the most recent and comprehensive view of the SSI system properties from various proposals. We noted that several SSI principles were described in a manner that was similar to the way SSI system properties are defined. Combining duplicates in these definitions is necessary to create a single input for our analysis. This phase begins by matching comparable or identical SSI principles and system properties. For instance, all four proposals defined the persistence property similarly. We combine duplicate constraints from the SSI...
principles and system properties into a single list and include the unique constraints. Then, we formalize the consolidated system properties akin to shared controls. Section IV details all the information regarding the consolidated list of properties.

The shared control derivation phase is meant to increase the CSSPS’s universality by discovering shared controls across several source documents that will be used to improve the existing system properties. We presume that the same evaluation criteria are included in the source documents governing information security and privacy. For example, the GDPR [5] and ISO/IEC 29100:2011 [9] both similarly describe privacy principles. As a result, this phase begins with a rigorous and systematic review of source documents to identify those pertaining to the SSI system. We construct selection criteria based on the SSI system’s requirements and the appropriateness of source documents. For instance, source documents that regulate organizational processes are incompatible with the SSI system properties. Then, we analyze the selected source documents to identify shared controls defined in numerous source documents. Finally, we formalize the shared controls in a manner analogous to how the system properties are formalized. Section V contains all pertinent information about shared controls.

The property enhancement phase aims to identify consistency between existing system properties and shared controls and leverage that consistency to improve property definition. We begin with the premise that some shared controls may be partially consistent or inconsistent with the existing system properties. To increase the likelihood of source document compliance, we can enhance the system properties by taking these partially consistent and inconsistent shared controls into account. This step begins by comparing each shared control to the consolidated system properties and classifying shared controls into three categories:

- A fully consistent control represents a control in which all endorsed tasks or conditions in the shared control are satisfied by system property constraints.
- A partially consistent control represents a control in which some endorsed tasks or conditions in the shared control are satisfied by property constraints.
- An inconsistent control represents a control in which none of the endorsed tasks or conditions in the shared control are satisfied by any system property.

Finally, we enhance the definition of SSI system properties by leveraging categories of shared controls. Indeed, we include the missing endorsed tasks or conditions resulting from partially consistent and inconsistent shared control into the applicable current system properties. On the other hand, we may introduce new system properties if the endorsed tasks or conditions are deemed unsuitable. Section VI discusses the CSSPS and our enhancement algorithms in detail.

IV. CONSOLIDATED LIST OF SSI SYSTEM PROPERTIES

This section will address existing SSI principles and system property proposals and how we develop input for property enhancement by removing duplicate constraints, consolidating, and formalizing the SSI principles and system properties.

A. CONSOLIDATING DUPLICATE SSI PRINCIPLES AND SYSTEM PROPERTIES

The SSI principles and system properties for the SSI system are related somehow, and one was derived from the others. These relationships result in duplications of SSI principles and proposed system properties. This section outlines our methodology for discovering and consolidating duplicate SSI principles and system properties.

Duplication happens when SSI principles and properties describe their evaluable constraints on the SSI system or its features similarly in multiple proposals. The transparency principle [1] and the transparency system property [4] are compared in Table 1, revealing the presence of duplicate constraints. We emphasize that both impose a constraint on the system, requiring it to be open source. We should begin by examining how they work [1].

| Property for the SSI System | SSI Principle |
|-----------------------------|--------------|
| Transparency. An SSI and its system must be transparent enough for every involved entity. A user should be well aware about all her/his 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 [4]. |
| 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 [1]. |

| TABLE 1. Comparison between the transparency SS principle and system property. |
particular constraints imposed by SSI principles, the SSI system and its features may or may not be evaluable. From Section I that the “existence” property has a constraint described as “any SSI is ultimately founded on the ineffable ‘I’ that is at the heart of identity,” which may make it difficult to evaluate the SSI system features directly. Constraints that are not evaluable should be excluded to express system properties that adhere to SSI principles.

We begin by evaluating the principles contained in three proposals [1], [2], [3] in order to determine which constraints can be applied to the SSI system’s features under the following conditions:

1. The SSI system or a system component thereof is the subject of the constraint.
2. The constraint directly restricts the SSI system features.
3. The constraint can be interpreted as being evaluable on the SSI system.

If at least one pair or group of constraints in SSI principles meets the criteria above, we consider them necessary. In Table 2, we present results for two additional SSI principles. As we can see, the “transparency” principle [3] imposes a constraint on the identity infrastructures that satisfy the condition (1). In contrast, the “interoperability” principle [1] includes the first constraint, which restricts the identity data object. This constraint cannot satisfy conditions (1) and (2). When we interpret its meaning, it is unclear how the SSI system is evaluated to ensure that it is as widely usable as possible. As a result, this constraint does not satisfy all specified conditions. When non-evaluable constraints are excluded, such SSI principles can be considered the same as SSI system properties.

Then, we compare SSI principles and system properties to discover duplicate constraints and combine duplicate and unique constraints into a single system property. The following criteria should be used to make the comparison:

1) They can be consolidated if the title of principles and properties are identical.
2) They can be consolidated if principles or system properties constrains an SSI system’s same element.
3) They can be consolidated if the constraints of principles and system properties are comparable or restrict the same component in the same way.

The criteria outlined above assist in determining which SSI principles and system properties can be consolidated. The points are that duplicate constraint that satisfies criterion (3) should be combined into a single constraint that contains all necessary information and that all principles and system properties that satisfy criteria (1) and (2) should be consolidated into a single system property. As shown in Table 3, the “existence” property is derived from three SSI principles and one system property. References to sources are marked by the IDs $S_x$, $y$, where $S$ is the source identifier for A denotes [1], F denotes [4], L denotes [2], and N denotes [3]; and $x$, $y$ denote the numbering scheme we use to refer to the $x$th constraint of the $y$th principle or system property.

As shown in Table 3, the first constraint on the “existence” property is derived from three sources. While the constraint definition might be expressed differently, we consolidate it to ensure that it contains all relevant information. While the final constraint is not a duplicate, it is still necessary and should also be included in the system property.

As a result, we obtain 20 consolidated system properties aligned with SSI principles that sufficiently represent the current state of knowledge about system properties. We can consider these system properties as a single set of constraints on implementing the SSI system.
B. FORMALIZING CONSOLIDATED SYSTEM PROPERTIES FOR THE SSI SYSTEM

The consolidated system properties obtained in the preceding section are in plaintext, which may require subjective interpretation in order to verify their consistency with shared controls. As a result, we propose a formalization of the consolidated system properties that minimize subjective interpretation and simplify subsequent analysis in this section.

We conducted an in-depth analysis of constraints in system properties and discovered that they all share a common meaning. We presume that the meaning of a constraint is that it restricts an SSI system activity or function that one or more components conduct on data items. For instance, the first constraint in Table 3 stipulates users that they must be able to represent their existence and characteristics. Assuming this, we define a system property as follows:

Definition 1: A system property is represented as a set of constraints, denoted by $P \subseteq S \times A \times O = \{c : c = (S_c, A_c, O_c)\}$, $S \subseteq S$, $A \subseteq A$, $O_c \subseteq O$ in which:

- $c$ denotes a constraint triple $(S_c, A_c, O_c)$,
- $S$ denotes a finite set of phrases describing the SSI system’s components that are subjects of the constraint,
- $A$ denotes a finite set of phrases describing the actions or functions of the SSI system that are under the constraint, and
- $O$ denotes a finite set of phrases describing the data objects associated with the constrained activities or functions.

According to the definition, a system property is composed of a collection of constraints, each of which can be represented by a triple $(S_c, A_c, O_c)$ consisting of sets of subjects, actions, and data objects restricted by the constraint. While the sets’ phrases need not be entirely drawn from the constraint definition, they must be explicit enough to convey the required meaning. On the other side, we discovered that certain constraints’ phrases might be omitted entirely, leaving readers to assume them. These circumstances increase the likelihood of errors in our systematic analysis. When formalizing system properties, they impose a requirement that such missing phrases be included in the triples and that further information is excluded concisely. To complete our formalization, we define the following criteria:

1) We must evaluate the constraint’s context and semantics and determine which phrases are omitted when the constraint’s subjects or data objects are empty (i.e., $S_c = \emptyset$ and/or $O_c = \emptyset$).
2) If the constraint definition contains excessive information, we must break it down and eliminate phrases that create an analysis overload.

The 20 consolidated system properties in the preceding section are formalized using the criteria above. Table 4 illustrates the formalized “decentralized” property in action. As can be seen, the decentralized property’s two constraints have been formalized individually. The first constraint is an example of an overly detailed constraint; it includes “by any proprietary organization” to clarify the context of the “centrally” phrase. We believe that only including the phrase “centrally” is adequate to explain its meanings. The second constraint is an illustration of data object omission. We discovered that the constraint omits data objects that the SSI system should register and manage. By examining the context and semantics of the constraint, we may assume that an identity attribute is the omitted data object, which will be included as the “(identity)” phrase. We use parenthesis to denote the omitted phrase. We summarize the titles of 20 system properties that we consolidated and formalized in Table 5. They will serve as a set of inputs for the property enhancement phase. Each of the formalized properties is denoted by the abbreviation $P_{CPx}$, which stands for the $x$th current system property (CP).

V. SHARED CONTROLS OF INFORMATION SECURITY AND PRIVACY

Another source of information for our property enhancement is the source documents’ controls. However, because hundreds of source documents relating to information security and privacy affect across business domains, the universality and application of the system properties will be limited if we enhance them with each individual source document. Apart from that, not all source documents are compatible with the SSI system or ideal for enhancing system properties. We assume that the source documents for information security and privacy contain a comparable set of evaluatable controls. As a result, we intend to derive such shared controls and use them as a universal set of knowledge sources in our property enhancement by using a systematic review approach.

### Table 4. Example of formalized “decentralized” system property.

| Property | Formalization |
|----------|---------------|
| Decentralized | An SSI system should not register and manage identities centrally by any proprietary organization. $c_1 = \{("SSI system"), \("not register"), \("not manage centrally")\}, \("(identity)\)\} |
|           | An SSI system should register and manage through a decentralized infrastructure mostly run publicly. $c_2 = \{("SSI system"), \("register"), \("manage through a decentralized infrastructure")\}, \("(identity)\)\} |

Note that the list of formalized system properties is provided in the “Property Consolidation” spreadsheet [10].

### Table 5. List of 20 system property titles that we formalized.

| Formalized and Consolidated Property Name | $P_{CP1}$: Existence | $P_{CP2}$: Sovereignty | $P_{CP3}$: Access | $P_{CP4}$: Transparency | $P_{CP5}$: Persistence | $P_{CP6}$: Portability | $P_{CP7}$: Interoperability | $P_{CP8}$: Consent | $P_{CP9}$: Data Minimization | $P_{CP10}$: Protection | $P_{CP11}$: Recovery | $P_{CP12}$: Single Source | $P_{CP13}$: Availability | $P_{CP14}$: Standard | $P_{CP15}$: Cost Free | $P_{CP16}$: Decentralized | $P_{CP17}$: Verifiability | $P_{CP18}$: Scalability | $P_{CP19}$: Accessibility | $P_{CP20}$: Sustainability |
|-----------------------------------------|-----------------------|------------------------|-------------------|-------------------------|------------------------|------------------------|------------------------|------------------|--------------------------|------------------------|----------------------|-----------------------|----------------------|-----------------------|----------------------|-----------------------|----------------------|----------------------|---------------------|----------------------|
In this section, we organize the main contents into two subsections: the review on source documents and the formalization of shared controls.

A. SYSTEMATIC REVIEW ON SOURCE DOCUMENTS TO DERIVE SHARED CONTROLS

This section explains how we derive shared controls from source documents using a systematic review method. To adhere to the standard procedure for system reviews, we have structured our approach into four parts: (1) the formulation of the research question; (2) the selection of document resources; (3) the search procedure, search terms, and selection criteria; and (4) the discovery of the shared control. Each step of our method is described in detail in the following sub-sections.

1) FORMULATION OF RESEARCH QUESTIONS

In order to derive shared controls, a systematic review should be conducted to identify source documents that can improve the system properties of the SSI system. Following is our main research question for conducting a systematic review.

RQ. What are shared controls that are commonly found in multiple laws, regulations, and technical standards of information security and privacy?

To address the aforementioned research question, it is necessary to find answers to the following guiding questions:

Q1. Where can applicable laws, regulations, and technical standards be gathered for our review?
Q2. Which source documents are appropriate for SSI system property enhancement?
Q3. How can we determine which controls are shared in the selected source documents?

Since we can gather laws, regulations, and technical standards from a variety of sources, ranging from government institutions to online repositories, the first guiding question aims to identify the knowledge source that will be utilized for this research. Due to the fact that not all source documents are applicable to our property enhancement, a second guiding question is required to specify which source documents should be included in our analysis and enhancement. The final guiding question aims to define precisely which shared controls will be utilized in our property enhancement. We will conduct a systematic review to independently pursue answers to the main research question and guiding questions.

2) SELECTION OF DOCUMENT RESOURCES

To answer the first guiding question (Q1), we must first develop a suitable search strategy for source documents. The primary purpose of the search is to collect as many relevant source documents as possible that explain information security and privacy. Today, laws, regulations, and technical standards are readily available online. The Council of the European Union, for instance, publishes the GDPR [5] on their website. As a result, we limited the scope of our review to online resources that could provide us with an acceptable and dependable list of titles of renowned source documents.

There are two trustworthy online resources, in our opinion: collaborative or official websites and survey papers. We chose Wikipedia as our source since it is a collaborative effort in which anyone may contribute and verify facts. Although Wikipedia is continuously updated, it provides us with a measure of trust that the list of source documents published has been subjected to community review. Another website we choose is ISO.org [11]. ISO.org is the International Organization for Standardization’s (ISO) official website, which has published hundreds of technical standards. ISO is a reputable worldwide organization when it comes to acquiring technical standards.

Another online resource that could supply us with a list of source documents is survey papers, in which academics spend their efforts acquiring and critically analyzing specific information and contextualizing it. Survey papers are extremely reliable since they have been subjected to peer review. Online repositories such as IEEE Xplore, ScienceDirect, Google Scholar, and SpringerLink make bundles of survey papers available to academics.

3) SEARCH PROCEDURE, SEARCH TERMS, AND SELECTION CRITERIA

Using the selected online resources, we can search for laws, regulations, and technical standards that are applicable to our property enhancement. We defined our search procedure into three steps: (1) Perform a keyword-based search on the selected online resources; (2) Gather the resulting source document from the online resources; and (3) Assess the gathered source documents against our inclusion criteria.

First, we use the keyword-based search engines that these websites and repositories provide. We created keywords (i.e., search terms) to identify as many titles for information security and privacy laws, regulations, and technical standards. We describe the keywords used and the search results in Table 6.

As shown in Table 6, we searched for survey papers using the keywords “information security and privacy laws, regulations, and standards” in online repositories. We identified only two survey papers [12], [13] that assess the present state of knowledge regarding information security standards and meet our requirements. Other survey papers discovered lacked a list of source documents and referred to only one or a few. Due to the limited number of source document titles provided by the survey papers, we opted to include our effort [6], which involved a manual search of information security and privacy source documents. It should supplement scholarly studies, providing another perspective on source documents.

On the other hand, we used keywords to discover five pages on Wikipedia that matched our requirements. They looked at information technology security standards, cybersecurity regulations, information privacy laws, and privacy law policies. Each of these pages has a list of published source
documents. However, there were no matches for information privacy standards in Wikipedia. Finally, we identified many information security and privacy standards on ISO.org. All online resources are presented with the number of source document titles it mentions (#Source).

After the first step, we obtained all source document titles in a discriminate manner and attempted to obtain just those that were publicly accessible and sufficient for analysis. We gathered 211 distinct source documents, including 134 ISO technical standards, 65 laws and regulations, and 12 frameworks for assessing security or privacy.

The scope of this work is defined as being limited to the 211 source documents that we obtained. Additional source documents may exist and be valuable for property enhancement, but we believe they may not widely be recognized due to online resources’ absence. We believe that the source documents obtained in this step are sufficient and representative of various source documents.

Because we obtained source documents from online resources in a discriminating manner, it is possible that even if the source documents discuss information security and privacy, they will be unable to be used to enhance system properties. For instance, several technical standards specify requirements for organizational processes that safeguard against data manipulation. This source document should be excluded to ensure that our property enhancement is more appropriate.

To answer the second guiding question (Q2), this step defines selection criteria for reviewing all source documents and identifying those that may be valuable for enhancing the SSI system’s system properties. As follows, we describe our inclusion criteria and justify their necessity:

- **CDN-1:** A source document must be established to regulate or evaluate a software system. This criterion is defined to exclude source documents addressed to non-software-related targets. It is inappropriate to compare consistency between system properties and organizational or process controls.

- **CDN-2:** A source document must be established for regulating or evaluating technologically independent targets. This criterion is defined to exclude source documents that are related to a specific technology. For example, while the payment card industry data security standard (PCI-DSS) [19] is a security standard, it is defined for payment card processing technology that may not mandate the implementation of the SSI system.

- **CDN-3:** A source document must be already published and available for use. This criterion excludes source documents currently under development or is not operational. For instance, the EU Directive 95/46/EC [20] established a data protection rule in 1995. However, this source document is no longer in force, despite being referenced in websites or survey papers.

- **CDN-4:** A source document must not be domain-specific. This criterion is defined to exclude domain-specific source documents in their application. If this kind of source document is used to enhance property definitions, it may limit the SSI system’s and its properties’ applicability inside a specific domain by providing too fine-grained constraints. The health insurance portability and accountability act (HIPAA) [21], for example, establishes requirements for healthcare information systems that manage patient information.

- **CDN-5:** A source document must provide at least one requirement, control, principle, or measurement eligible to evaluate the identity management system. This criterion excludes source documents that did not provide any evaluable controls. The source documents require controls to provide information for comparing the system properties. For example, ISO/IEC29192-2:2019 [22] is a standard that defines just algorithms and designs for lightweight cryptography. We argue that such algorithms and designs are not considered controls.

- **CDN-6:** A source document must provide at least one requirement, control, principle, or measurement eligible to evaluate the identity management system. This criterion excludes source documents that provide incompatible controls with identity management systems. The SSI system may be intrinsically incompatible as an identity management system. ISO/IEC 18032:2005 [23], for example, provides requirements for prime number generators, but the SSI system might not always use prime number generators.

We comprehensively analyzed all 211 source documents obtained using the criteria outlined above. We infer that a source document meets a criterion if we can substantiate it with evidence or an appropriate justification. We intend to employ all source documents that satisfy all selection criteria.

### TABLE 6. Summary of online resources, keywords used, and the number of source documents gathered.

| Resource          | Keyword                      | Search Result | #Source |
|-------------------|------------------------------|---------------|---------|
| Online Repositories | Information Security Standards | A: [12]       | 10      |
|                   |                              | B: [13]       | 44      |
|                   | Information Privacy Standards |              |         |
|                   | (Information Security OR Information Privacy) AND (Laws OR Regulations) |              |         |
| Wikipedia         | Information Security Standards | C: [14]       | 20      |
|                   | Information Privacy Standards |              |         |
|                   | (Information Security OR Information Privacy) AND (Laws OR Regulations) |              |         |
| ISO.org           | (Security OR Privacy)        | H: [11]       | 122     |

Note that the list of source documents gathered is in the “Source Review” spreadsheet [10].
to derive shared controls. The review results are summarized in Fig. 8, demonstrating that we obtained 28 source documents that met the selection criteria.

Additionally, we show in Table 7 the 28 source documents that we believe are appropriate for property enhancement. SD, denotes the ith source document of the 211 source documents. Each entry in a source document is specified by the number of controls included within it (#Control) and the executable scope of the source document (Scope). For example, the Personal Data Act (PDA) [24] imposes nine controls on systems that process Swedish PII. The scope of this work has been condensed to include only the 28 source documents that met all of the selection criteria.

4) DISCOVERY OF THE SHARED CONTROLS

To answer the last guiding question (Q3), this final part is defined. We intend to identify similarities across controls that evaluate the same aspect and derive shared controls from these similarities. We observed situations when the same control appeared in numerous source documents. For instance, as illustrated in Table 8, the data accuracy control is defined in five source documents.

However, the control in various source documents is not always the same. Additionally, we must assess all endorsed tasks and conditions defined in the control. For example, while an article in SD denotes the 173th source document of the 211 source documents, it is not described as an accuracy control, it does set similar conditions on the accuracy of personal information. Using the following criteria, we find relevant controls, endorsed tasks, and conditions:

1) If a control is named or has a title, we look for comparable controls with similar names or titles and assume they are related.
2) When a control is named or entitled, we look for controls with different names but a common purpose. We assume that these controls are related.
3) If a control is unnamed or has a unique name, we find comparable controls with at least one endorsed task or condition, which may be justified as comparable or identical. We assume that these controls are related.

We collect all groups of controls that are considered related and create a taxonomy from which its elements can be identified as shared controls. Some endorsed tasks or conditions in the shared control may be comparable or identical, and we aim to consolidate them so that they can cover all required aspects. However, we found that different source documents employ distinct terminologies, making the consolidations problematic. We define representatives for equivalent terms defined in source documents to address this issue, as shown in Table 9. The table shows that we can, for example, substitute the terms “application,” and “PII processor” with the “system.”

We consolidate the endorsed tasks and conditions of related controls and construct a shared control from them. The right-hand column of Table 8 illustrates how the shared “data accuracy and quality” control is implemented.

| Source Document Identifier and Title/Name | #Controls | Scope |
|------------------------------------------|----------|-------|
| SD005 = ISO/IEC 9796-2:2010 [25]          | 3        | International |
| SD007 = ISO/IEC 9798-2:2008 [26]          | 6        | International |
| SD009 = ISO/IEC 10118-1:2018 [27]         | 1        | International |
| SD017 = ISO/IEC 11770-1:2010 [28]         | 5        | International |
| SD022 = ISO/IEC 13885-1:2020 [29]         | 2        | International |
| SD027 = ISO/IEC 18014-1:2008 [30]         | 2        | International |
| SD037 = ISO/IEC 18031:2005 [31]           | 5        | International |
| SD049 = ISO/IEC 19772:2020 [32]           | 2        | International |
| SD065 = ISO/IEC 27001:2013 [33]           | 16       | International |
| SD068 = ISO/IEC 27002:2013 [34]           | 16       | International |
| SD085 = ISO/IEC 27701:2019 [35]           | 20       | International |
| SD104 = ISO/IEC 29100-2011 [9]            | 11       | International |
| SD133 = IEC 62443-3:2013 [36]            | 7        | International |
| SD126 = NERC Cyber Security Standards [37]| 6        | India |
| SD136 = NIST Special Publication 800-12 Revision 1 [38] | 20 | International |
| SD141 = Cyber Essentials [39]             | 5        | International |
| SD142 = OECD Privacy Framework [40]       | 8        | International |
| SD146 = UN Personal Data Protection and Privacy Principles [41] | 10 | International |
| SD152 = General Data Protection Regulation [5] | 7 | International |
| SD160 = The Information Technology Act [42] | 17 | India |
| SD165 = The Personal Data Protection Code of Practice [43] | 19 | Malaysia |
| SD173 = Data Privacy Act [44]             | 12       | Philippines |
| SD175 = Federal Law on Personal Data [45]  | 10       | Russia |
| SD187 = Personal Data Act [24]            | 9        | Sweden |
| SD189 = Personal Data Protection Act [46]  | 3        | Taiwan |
| SD195 = Children’s Online Privacy Protection Act [47] | 3 | United States |
| SD207 = Personal Information Protection Law [48] | 13 | China |
| SD211 = OWASP Application Security Verification Standard [49] | 14 | International |

Note that the list of source documents and results for the selection criteria are provided in the “Source Review” spreadsheet [10].

It demonstrates that accuracy and quality control are used similarly in SD104, SD126, and SD207. Criterion (1) presupposes that they are related. Additionally, all controls assumed to be related share an endorsed task for determining whether personal data is accurate, complete, and up to date. This endorsed task is consolidated into the first endorsed task with shared control. Additionally, we replace our representative terms to make the text more comprehensible.

After our method, we identified 17 shared security and 14 shared privacy controls similarly stated in various source documents. The titles of all shared controls are listed in Table 10, separated into two categories: security and privacy. S. j, and P. k are used to denote the jth security shared control, and the kth privacy shared control, respectively.

B. FORMALIZATION OF SHARED CONTROLS

Similar to the consolidated system properties, shared controls are in plaintext and may require subjective interpretation. This section proposes a formalization of shared controls analogous to the consolidated system properties.
We begin by comprehensively analyzing the control definition. We break down control definitions into sentences and consider them as endorsed tasks. We noticed that each endorsed task is defined to restrict the functions or processes in which the evaluation targets must perform on data. Endorsed tasks are deemed to be similarly defined as system property’s constraints. Assuming this idea, we define a shared control as:

Definition 2: A shared control is represented as a set of endorsed tasks, denoted by $C \subseteq T \times F \times I = \{e : e = (T_e, F_e, I_e), T_e \subseteq T, F_e \subseteq F, I_e \subseteq I\}$ in which:

- $e$ denotes an endorsed task triple $(T_e, F_e, I_e)$,
- $T$ denotes a finite set of phrases describing the targets of evaluation,
- $F$ denotes a finite set of phrases describing the functions of the evaluation targets to be controlled, and
- $I$ denotes a finite set of phrases describing the information or data objects associated with the controlled functions.

According to the definition, a shared control is composed of a collection of endorsed tasks, each of which can be represented by a triple $(T_e, F_e, I_e)$, consisting of sets of evaluation targets, controlled functions, and information controlled by the endorsed task. However, omission and excessive amounts of information circumstances have the potential to be analogous to system properties. We also establish the following criteria to complete our formalization:

1) We must analyze the endorsed task’s context and semantics and determine which phrases are omitted when the endorsed task’s targets or information are empty (i.e., $T_e = \emptyset$ and/or $I_e = \emptyset$).

2) If the endorsed task’s sentences contain excessive information, we must break it down and eliminate phrases that create an analysis overhead.

We formalize all 17 shared security and 14 shared privacy controls using the definition and criteria above. Table 11 illustrates an example of a formalized data integrity shared control. It demonstrates that the data integrity shared control includes five endorsed tasks, each of which is formalized by Definition 2. Except for the fourth task ($e_4$), none of the tasks endorsed include evaluation targets. After examining the control’s context, we found that the target should be the system. We incorporate the phrase “system” into the set of evaluation targets and indicate this with parenthesis. On the other hand, an excessive amount of information is omitted from the second task ($e_2$), specifically the phrase “for message signing and verification,” which elaborates on the “key” phrase.

VI. COMPLIANCE SSI SYSTEM PROPERTY SET (CSSPS)

In Sections IV and V, we prepared two sets of inputs for property enhancement. We are now able to determine whether the consolidated system properties are consistent with the shared control. This section provides a clear meaning of the consistency among system properties and shared controls. Then we present our systematic method for determining whether shared controls are already consistent with any system properties and how to use the partially
A. CONSISTENCY AND INCONSISTENCY BETWEEN SYSTEM PROPERTIES AND SHARED CONTROLS

TABLE 10. List of shared control titles categorized into security and privacy shared controls.

| Security Shared Control | Privacy Shared Control |
|-------------------------|------------------------|
| S.1. Data Integrity      | S.10. Key Protection   |
| S.2. Data Confidentiality| S.11. Malware Protection|
| S.3. Data Availability   | S.12. Communication Security |
| S.4. Authentication      | S.13. Physical and      |
| S.5. Authorization        | Environmental Security  |
| S.6. Accountability       | S.14. Password Security  |
| S.7. Non-Reputation       | S.15. Configuration Security |
| S.8. Validation and Sanitization | S.16. Session Security |
| S.9. Error Handling       | 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 |

Note that the list of shared controls and their endorsed tasks is provided in the “Source Review” spreadsheet [10].

TABLE 11. Example of formalized “data integrity” shared control.

| Control                        | Formalization                                      |
|--------------------------------|-----------------------------------------------------|
| S.1. Data Integrity            | Personal data message must be signed and authenticated using digital signatures.  
| e₁ = (("system"), {"sign", "authenticate"}, {"personal data"}) |
| Keys for message signing and verification must be shared in a secure way.  
| e₂ = (("system"), {"share", "in a secure way"}, {"key"}) |
| Digital time-stamping should be used in a non-forgable way.  
| e₃ = (("system"), {"use", "time-stamping", "in a non-forgable way"}, {"key"}) |
| Systems must validate the integrity of the personal data.  
| e₄ = (("system"), {"validate the integrity"}, {"personal data"}) |
| If the message authentication uses a hash function, it must be collision resistant.  
| e₅ = (("system"), {"use", "a hash function", "collision-resistant"}, {"personal data"}) |

Note that the list of all shared controls is provided in the “Source Review” spreadsheet [10].

consistent and inconsistent shared controls to enhance system properties.

A. FORMALIZATION OF CONSISTENCY

This section will explain how we determine consistency between SSI system properties and shared controls to identify the lack of constraints in existing SSI system property proposals.

As mentioned in Section I, we recognized that SSI system properties are comparable to controls in source documents. Both define how the targets should perform some operations on data objects. Therefore, the consistency of a system property and a shared control should be reflected in three aspects: (1) the evaluation targets in the shared control are comparable to the subject of constraints in the system property, and (2) the controlled functions in the shared control are comparable to the constrained actions in the system property, and (3) the affected data objects in the shared control and the system property are comparable. Adopting the idea outlined above, we defined a comparable operator as follows:

Definition 3 (Comparable Operator): A comparable operator \( \simeq \) denotes the similarity of two sets of phrases. \( X \simeq Y \) if \( \exists x \in X, \exists y \in Y \) such that \( x \) can be justified to be comparable to \( y \) based on evidence.

Then, by referring to the element of the formal definitions (Definitions 1 and 2), we may use the comparable operator to define the consistency between system properties and shared controls as follows:

Definition 4: A system property and a shared control are consistent if and only if \( \exists c = (S_c, A_c, O_c) \in P \) and \( \exists e = (T_e, F_e, I_e) \in C \) s.t. \( S_c \simeq T_e, A_c \simeq F_e, O_c \simeq I_e \).

According to the above definition, a system property and a shared control are consistent if at least one constraint triple includes elements that are comparable to those of the endorsed task of the shared control.

B. METHODS FOR IDENTIFYING CONSISTENCIES AND ENHANCING PROPERTIES

Following the previous section’s formalization of consistency, we can determine which shared controls have any consistent system properties. The primary goal of determining consistency is to ascertain the number of shared controls consistent with the existing set of system properties. This section will describe our method for identifying consistencies and utilizing the resulting data to improve properties. We leverage algorithms over formalizations to make our method systematic. We illustrate our method overview in Fig. 9.

As illustrated in the figure, we present two methods: identifying consistency and enhancing system properties. The first method compares each pair of an endorsed task and a constraint to ascertain their consistency. Additionally, the first method classifies shared controls according to the number of consistent endorsed tasks. The second method shall employ various categories of shared control to enhance existing system properties by updating missing endorsed tasks or introducing new system properties. Each method will be discussed in detail in the next sections.

1) IDENTIFYING CONSISTENCY

In Algorithm 1, we create a semi-automatic algorithm to reflect our method for identifying consistency between shared controls and system properties. The algorithm declares two critical procedures: identifyConsistency() and...
classifyControl(). The identifyConsistency() procedure determines which pairs of an endorsed task and a constraint are consistent. The classifyControl() procedure attempts to classify a shared control into one of three categories based on the number of consistent endorsed tasks: fully consistent (FC), partially consistent (PC), or inconsistent (IC).

On the inputs: a shared control (C), a set of consolidated system properties (CPL), and a set of preset phrasal associations (PA), the identifyConsistency() procedure outputs a quadruple indicating a corresponding control (C) along with a classified category (type), a set of consistency resultant triples (R\text{consist}) and a set of missing endorsed tasks (R\text{lack}). In line 2, local variables are initialized, including an empty set of consistency resultant triples (R\text{consist}), an empty set of missing endorsed tasks (R\text{lack}), a degree of consistency (degree) as a zero integer, and a flag of consistency (flag) as false. Then, we iterate over each endorsed task (T_i, F_i, I_i) in the given shared control C. We compare the endorsed task with each constraint (S_k, A_k, O_k) in every system property P_j \in CPL. At line 7, two functions: evaluate() and determineDirection() are used to evaluate how two sets of phrases are consistent.

Because phrases in the SSI system property’s constraints and shared control’s endorsed tasks are specified differently, we may necessitate the subjectivity of judgment to evaluate their associations. However, in both system property and shared control definitions, sets of phrases are utilized consistently. As a result, we can establish a preset of phrasal associations that indicate how two phrases are consistent, using our domain knowledge. We illustrate our preset of phrasal associations in Table 12. For example, the term “infrastructure” in constraints is consistent with the term “system” in endorsed tasks. We formulate pairs of phrasal associations in a set (PA) such as {(infrastructure, system), . . . } and use them in Algorithm 1.

On the inputs: two sets of phrases (e.g., S_i and T_j) and the preset of phrasal associations (PA), the evaluate() function returns a Boolean value, verifying that the phrases in the sets are consistent, based on the comparable operator (Definition 3). The evaluate() function may require manual input when phrases are not defined in the preset. However, the phrases in the sets defining controlled functions (F_i) in endorsed tasks and constrained actions (A_k) in constraints vary widely and are difficult to associate in advance. This issue is addressed by the determineAssociation() function, which requires manual input to determine the subjective consistency of two sets of phrases.

If the conditions in line 7 are valid, a consistency among phrases is found. Then, the determineDirection() function is developed to determine how much an endorsed task is different from a constraint because two sets of phrases may be consistent but one set contains more information than another. We define the function as a partial map determineDirection: ((S, A, O), (T, F, I)) \rightarrow \{<, =, >\}, where > indicates that the constraint has more information than the endorsed task, < indicates that the constraint has less information than the endorsed task, and = indicates that both of them contain equal or similar information. The procedure uses the variable sign for denoting the direction. The direction of each pair that is consistent will aid us in choosing the suitable way to enhance the system property.

In lines 9 and 10, the procedure maintains the consistent pairs of an endorsed task and a constraint in a consistency resultant set (R\text{consist}) and set the flag of consistency (flag) to true. An endorsed task can be consistent with multiple constraints. Then, the condition in line 11 determines whether the endorsed task has any consistent constraint. If so, the degree of consistency (degree) will be increased, indicating how many endorsed tasks in the given shared control (C) are consistent. Otherwise, the endorsed task will be maintained as a missing endorsed task in the set R\text{lack}.

The classifyControl() procedure returns a category (type \in \{“FC”, “PC”, “IC”\}) of shared control based on the inputs: the given shared control (C) and its degree of consistency (degree). The degree of consistency will be compared to the cardinality of the endorsed task (i.e., |C|). If they are equal, all endorsed tasks are consistent with any constraint, and the given shared control can be classified as fully consistent. If the degree of consistency is greater than zero but less than the cardinality, the given shared control can be classified as partially consistent. Apart from that, the shared control is classified as inconsistent.
These procedures are sufficient for determining which shared controls are consistent with current system properties. A running example of Algorithm 1 is provided below, using a constraint from the persistence system property and an endorsed task from the shared control for data erasure and rectification:

\[ c_1 = \{ \text{SSI system}, \text{dispose if wish to}, \text{identity} \} \]  
\[ e_1 = \{ \text{system}, \text{“erase if no necessary”}, \text{data} \} \]

When the algorithm reaches line 7, the evaluate() function checks to see if the preset (PA) defined the association between the phrases “SSI system” and “system.” As can be seen in Table 12, these phrases are associated. A similar situation occurs when the phrases “identities” and “data” are used. The evaluate() functions in line 7 all return true in these cases. Additionally, the determineAssociation() function determines the consistency of the controlled functions and actions. This determination should be made manually, and we discovered that the phrases are consistent since they make similar statements regarding the removal function. When we encounter the determineDirection() function, we consider the endorsed task (\( e_1 \)) to have more information because the identity is no longer required if the owner wishes to remove it or if the identity accomplishes its objectives. The constraint \( c_1 \) merely outlines the action of disposing of it when users decide to do so. As a consequence, the function returns the sign variable as >. However, among the four tasks in the shared control, this endorsed task is the only one that is consistent. This shared control will be classified as partially consistent by the classifyControl() procedure in Table 13. The table is presented with the category of shared control, the consistency resultant triples, and the missing endorsed tasks. The non-repudiation control, for example, is classified as partially consistent. Despite that the shared control has only one endorsed task that is consistent with the protection property, it has the > direction sign, indicating that the endorsed task has more information than the constraint. Our method reports 4 fully consistent, 13 partially consistent, and 14 inconsistent shared controls.

2) ENHANCING SYSTEM PROPERTIES

In Algorithm 2, we create a semi-automatic algorithm to reflect our method for enhancing system properties of the SSI system based on three categories of shared controls. The key idea of this algorithm is to include missing endorsed tasks in partially consistent or inconsistent shared controls into the current system properties. The enhanceProperty() procedure delivers the formalized CSSPS (CSSPS) on the inputs: a set of all results from Algorithm 1 on all shared controls (\( R \)), and a set of consolidated system properties (CPL). In line 2, an empty set CSSPS is initialized to be a resultant set of which the revised system properties are appended. Each quadruple \( (c, t, R_{\text{consist}}, R_{\text{lack}}) \in R \) will be used to revise system properties according to the category (type) provided.

If the shared control \( c \) is partially consistent (PC), the determineFitOrCreate() function on line 6 will scan the set of consolidated system properties CPL and the latest CSSPS for the fit property for each missing endorsed task \((T_i, F_i, I_i) \in R_{\text{lack}}\). If there is a fit property, the missing endorsed task will be added as a new constraint to the property and the property will be moved to CSSPS. Otherwise, the function will append
TABLE 13. Result of the consistency analysis on all shared controls, denoting a list of controls along with their category, consistency resultant triples, and missing endorsed tasks.

| Control C | Category | Consistency Triple R<sub>consist</sub> | Resultant Triple R<sub>resultant</sub> | Missing Task R<sub>task</sub> |
|-----------|----------|----------------------------------------|----------------------------------------|-------------------------------|
| S.1       | PC       | (S.1.3, CP.10.4, >)                    | S.1.1, S.1.2, S.1.4                  | S.2.2, S.2.3                  |
| S.2       | PC       | (S.2.1, CP.10.4, <)                    | S.2.2, S.2.5                        | S.3.1, S.3.2                  |
| S.3       | PC       | (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.5.1, S.5.2                  |
| S.5       | IC       | -                                      | S.6.1, S.6.2                        | S.6.6, S.6.7                  |
| S.6       | IC       | -                                      | S.7.1, S.7.2                        | S.8.1, S.8.2, S.8.3           |
| S.7       | PC       | (S.7.1, CP.10.4, >)                    | -                                     | -                             |
| S.8       | IC       | -                                      | -                                     | -                             |
| S.9       | PC       | (S.9.2, CP.11.1, >)                    | S.9.1                                | S.10.1, S.10.2                |
| S.10      | PC       | (S.10.3, CP.10.6, >)                   | S.11.1, S.11.2                      | S.11.2, S.12.3                |
| S.11      | PC       | (S.12.1, CP.10.6, >)                   | S.12.2, S.12.3                      | S.12.4, CP.10.6, >            |
| S.12      | IC       | -                                      | -                                     | -                             |
| S.13      | IC       | -                                      | S.13.1, S.13.2                      | S.14.1                        |
| S.14      | IC       | -                                      | S.14.1                                | S.15.1, S.15.2                |
| 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.17.1                        |
| S.17      | IC       | -                                      | -                                     | -                             |

P.1       | PC       | (P.1.1, CP.3.7, =)                     | P.1.3, P.1.4, P.1.5                  | P.1.6, P.1.7                  |
| P.2       | PC       | (P.2.1, CP.8.1, =)                     | P.2.2, P.2.3                        | P.3.4                        |
| P.3       | IC       | -                                      | P.3.1                                | P.4.2                        |
| P.4       | IC       | -                                      | P.4.1, P.4.2                        | P.5.5                        |
| P.5       | PC       | (P.5.1, CP.8.2, >)                     | P.5.1, P.6.4                        | P.6.6                        |
| P.6       | PC       | (P.6.2, CP.5.1, >)                     | P.6.3, CP.5.3, >                     | P.7.1, P.7.2                 |
| P.7       | IC       | -                                      | P.8.1, P.9.1, P.9.2                 | -                             |
| P.8       | PC       | (P.9.1, CP.5.2, >)                     | -                                     | P.9.5                        |
| P.9       | PC       | (P.9.1, CP.5.2, >)                     | -                                     | P.10.2                       |
| P.10      | PC       | (P.10.1, CP.13.1, >)                   | P.10.2                               | P.11.1                       |
| P.11      | IC       | -                                      | P.11.1, P.11.2, P.11.3              | P.11.4                       |
| P.12      | PC       | (P.12.1, CP.4.1, =)                    | P.13.1, P.13.2, P.13.3              | P.13.4                       |
| P.13      | IC       | -                                      | -                                     | -                             |
| P.14      | PC       | (P.14.1, CP.6.2, =)                    | -                                     | -                             |

Note that the list of results is provided in the “Consistency Analysis and Property Enhancement” spreadsheet [10].

a new property to CSSPS holding the endorsed tasks that are missing.

In line 7, the loop condition iterates over all consistency resultant triples in R<sub>consist</sub>. This iteration is to revise existing constraints that include less information than the corresponding endorsed task. The condition in line 8 checks whether the direction (sign) is set to >. If so, the merge() function will combine the existing constraint (S<sub>i</sub>, A<sub>i</sub>, O<sub>i</sub>) and the corresponding endorsed task (T<sub>j</sub>, F<sub>j</sub>, I<sub>j</sub>) so that S<sub>merge</sub> = S<sub>i</sub> ∪ T<sub>j</sub>, A<sub>merge</sub> = A<sub>i</sub> ∪ F<sub>j</sub>, and O<sub>merge</sub> = O<sub>i</sub> ∪ I<sub>j</sub>. However, the sets S<sub>merge</sub> and O<sub>merge</sub> should contain only SSI-specific terminology and the phrases in T<sub>j</sub> and I<sub>j</sub> that are associated by the preset of phrasal associations (Table 12) must be excluded from the sets. The revise() function will then put the merged constraint triple (S<sub>merge</sub>, A<sub>merge</sub>, O<sub>merge</sub>) in the corresponding system property in CPL and move the property to CSSPS.

If the shared control (C) is inconsistent (IC), all endorsed task (T<sub>i</sub>, F<sub>i</sub>, I<sub>i</sub>) ∈ R<sub>task</sub> will be determined to find the fit property or append a new property to CSSPS using the same determineFitOrCreate() function. In the last line of the procedure, the includeExistingProperty() function on line 14 will compare the set of consolidated system properties (CPL) to CSSPS in order to find and relocate the system properties that were not improved to CSSPS.

Algorithm 2 can be demonstrated using the same pair of endorsed tasks and constraints as in the previous section. One element in the consistency resulting set (R<sub>consist</sub>) is (c<sub>1</sub>, e<sub>1</sub>, >) as a result of Algorithm 1. The merge() function on line 9 will combine the constraint c<sub>1</sub> with the endorsed task e<sub>1</sub>, resulting in the following altered constraint:

\[ (\{"SSI system\}, \{"dispose if users wish to\", "erase if no longer necessary\"\}) \]

As can be seen, the phrases “SSI system” and “system” are comparable, and we, therefore, include only the phrase “SSI system” in S<sub>merge</sub>. The terms “identities” and “data” refer to the same idea in O<sub>merge</sub>. By contrast, the constrained action set contains elements from both A<sub>j</sub> and F<sub>j</sub>. The revise() function at line 10 will revise the triple of merged constraints and copy the corresponding property to CSSPS. As a result, when Algorithm 2 was executed, 42 system properties for the SSI system in the CSSPS were acquired. The resulting system properties are in a formalized form.

C. FORMULATING THE CSSPS

As a result of our approach, the CSSPS is described in the formalized system properties, which may make adoption more difficult. This section describes our method for converting the formalized system properties back to plaintext.

We discovered that the enhanced system properties constraint should be written in plaintext to make acceptance and translation into system requirements easier. The formalized constraint is compared to the constraint specifications for the consolidated system properties and the endorsed task to achieve this. The excessive amount of information that may be excluded under our Sections IV-B and V-B criteria should be carefully considered. Our method aims to incorporate all specific information required by the formalized CSSPS and provide a plaintext description of the CSSPS. Table 14 demonstrates how we formulate the persistence system property in CSSPS. It demonstrates how we revise the constraint by inserting all two critical points from the set of constrained actions (A) and substituting information such as the sentence “the corresponding user wishes to modify, revoke, or delete them” into the plain text.

On the other hand, we realized that the CSSPS has over 40 system properties, which may make it difficult for system implementors to integrate it quickly. We categorize the system properties into five categories to aid in adoption if system implementors need to comply with only a subset of them: SSI foundation, information security, system security, information privacy, and ease of use. In Fig. 10, we present the taxonomy of system properties in CSSPS. Each system property in CSSPS is designated by the identifier IP<sub>x</sub>, where <sub>x</sub> denotes the xth system property in CSSPS according to our ordering approach.

VOLUME 10, 2022 99385
Algorithm 2 Enhancing System Properties

1: procedure enhanceProperty(\(\mathcal{R}, \mathcal{CPL}\))
2: \(\mathcal{CSSPS} \leftarrow \emptyset\)
3: for \((C, \text{type}, R_{\text{consistent}}, R_{\text{lack}}) \in \mathcal{R}\) do
4:   if type = “PC” then
5:     for \((T_i, F_i, I_i) \in R_{\text{lack}}\) do
6:       CSSPS \leftarrow determineFitOrCreate\((C, (T_i, F_i, I_i), \mathcal{CPL}, \mathcal{CSSPS})\)
7:   end if
8:   for \(((S_i, A_i, O_i), (T_j, F_j, I_j), \text{sign}) \in R_{\text{consistent}}\) do
9:     if sign = > then
10:        \((\mathcal{S}_{\text{merge}}, \mathcal{A}_{\text{merge}}, \mathcal{O}_{\text{merge}}) \leftarrow \text{merge}(S_i, A_i, O_i), (T_j, F_j, I_j))\)
11:       CSSPS \leftarrow \text{revise}((\mathcal{S}_{\text{merge}}, \mathcal{A}_{\text{merge}}, \mathcal{O}_{\text{merge}}), (S_i, A_i, O_i), \mathcal{CPL})\)
12:     end if
13:   end for
14: return includeExistingProperty(\(\mathcal{CPL}, \mathcal{CSSPS}\))

TABLE 14. Example of the plaintext persistence system property formulated from the formalization.

| Formalized Property | Plaintext System Property in CSSPS |
|---------------------|----------------------------------|
| Persistence:        | Persistence: ... 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. ... |

VII. EVALUATION

The CSSPS is offered as a new set of system properties for the SSI system. These properties should be of high quality in two areas: (1) likelihood of consistency with as many source documents as possible and (2) applicability in real-world situations. This section will analyze and assess these two areas.

A. ANALYSIS OF THE DEGREE OF CONSISTENCY

The degree of consistency indicates how closely the CSSPS system properties adhere to the controls in the source documents. With 42 system properties in the CSSPS, 12 are preserved, 11 are enhanced, and 19 are added, we hypothesized that the CSSPS would be significantly compliant with most source documents about information security and privacy. To verify and evaluate this hypothesis, we perform an experiment in which we solicit expert opinion on the effect of CSSPS on the degree of consistency relative to the current system properties.

1) EXPERIMENTAL SETTINGS

By adjusting independent variables and measuring dependent variables, we build an experiment to test our hypothesis. All of the settings in our experiment are as follows:

a: DEPENDENT VARIABLE

The percentage of consistency and the difference in consistency between the existing set of consolidated system properties and the CSSPS are employed as dependent variables in this experiment. To calculate the percentage of consistency, we use the following equation:

\[
PC_{\text{prop, source}} = \frac{XC_{\text{prop, source}}}{TC_{\text{source}}} \times 100
\]

where \(PC_{\text{prop, source}}\) denotes the percentage of consistency between a set of properties (prop) with a source document (source), \(XC_{\text{prop, source}}\) is the number of controls in the source document (source) that the analyst considers to comply with any properties in the set (prop), and \(TC_{\text{source}}\) denotes the total number of controls contained in the source document (source). The percentage of consistency is comparable and can be reflected in terms of the difference as below:

\[
\Delta_{A,B} = PC_{A,\text{source}} - PC_{B,\text{source}}
\]

where \(\Delta_{A,B}\) is the difference in compliance between two sets of properties \((A, B)\), \(PC_{A,\text{source}}\) and \(PC_{B,\text{source}}\) denote the percentage of compliance with the source document (source) of the sets of properties \(A\) and \(B\), respectively. Between -100 and +100, the difference might be either positive or negative. A positive difference indicates that the first set of system properties \((A)\) has greater consistency than the second set of system properties \((B)\).

b: INDEPENDENT VARIABLE

The sets of system properties and the set of source documents consistent with the SSI system’s system properties are two groups of independent variables in our experiment.

We defined two distinct sets of system properties: one for the present proposal and another for the CSSPS. The existing set is derived from the list of consolidated system properties obtained in Section IV, as it combines four proposals. This should be an excellent depiction of the property proposal’s current state. On the other side, we will evaluate the CSSPS in contrast to the existing set of consolidated system properties.

For the set of source documents, we divide our source documents into two categories: those that were used to
enhance our work (GDPR [5] and OWASP Application Security Verification Standard - OWASP ASVS [49]) and those that were not used because they did not fit our selection criteria but remained relevant (Health Insurance Portability And Accountability - HIPAA [21] and Thailand’s Personal Data Protection Act - PDPA [50]). These source documents have been chosen to demonstrate the extent to which the CSSPS can be applied universally. GDPR and OWASP ASVS were chosen to demonstrate how the CSSPS can improve the degree of consistency compared to existing ones. HIPAA was chosen due to its domain-specific nature and requirement for integrating the SSI system into a healthcare software application. Thailand’s PDPA was chosen because it matched five of six criteria, except for not yet being in force. Thailand’s PDPA, once enacted, will meet all of the criteria for regulating the SSI system.

c: QUALIFICATION FOR PARTICIPATION
To provide input data for our experiment, we require expert opinion to demonstrate how closely the system properties satisfy the controls in the provided source documents. To avoid author bias, we enlisted the help of a system analyst with at least three years of expertise in building software and security awareness training. Constraints on system properties, we believe, can be understood similarly to system requirements, simplifying their comprehension of the analyst. On the other hand, security awareness training should familiarize the analyst with the controls contained in source documents.

2) EXPERIMENTAL PROCEDURE AND FINDINGS
In this section, we describe our experimental procedure and provide our findings in the following manner:

a: EXPERIMENTAL PROCEDURE
Prior to the experiment session, we familiarize the analyst with the fundamental concepts and functionalities of the SSI system. We encourage the analyst to ask as many questions as necessary and to request additional explanations. This ensures that the analyst has a clear understanding of the SSI system before the experiment session.

We illustrate an overview of our experimental procedure in Fig. 11. As illustrated in the figure, our experimental procedure consists of two rounds. The analyst is asked to compare the existing set of consolidated system properties to the four source documents in the first round. We request justification from the analyst on why the system properties are consistent with the controls in each source document. In the second round, we request that the analyst compare the identical set of source documents to the CSSPS’s system properties. After both rounds, we collect the analyst’s decisions, justifications, and comments. Then, using Equations 3 and 4, we calculate the percentage of consistency and the difference between the existing set of consolidated system properties and the CSSPS.

b: FINDINGS
We summarize the computed percentages of consistency and differences in Table 15. It can be observed that the source column indicates the source documents used in the comparison, and the TC column represents the number of controls provided by the source document. The XC_EXIST and PC_EXIST columns indicate the number of controls in the source document (source) and the percentage of consistency that are consistent with system properties in the existing set, respectively. Similarly, the XC_CSSPS and PC_CSSPS denote the number of controls and the percentage of consistency that properties in the CSSPS are consistent with the source document (source). The last column reports the difference in consistency between the existing set of consolidated system properties and the CSSPS.

There are crucial pieces of information that demonstrate the CSSPS’s success in terms of consistency percentages. Below, we discuss each observation we made:

Finding 1: The differences (ΔEXIST_CSSPS) indicate a higher level of consistency across all source documents used in our enhancement.

In almost every control, the CSSPS can be consistent with two source documents in the case of the source documents utilized in our enhancement. Each source document contains a single control that is deemed inconsistent. The analyst contends that the existence system property appears to violate the fundamental concepts of the SSI. For example, the GDPR’s storage limitation control is consistent with the CSSPS’s properties, except that it defines storage for public interest purposes, which may contradict the users’ initial consent.
Finding 2: The CSSPS affects the consistent percentage of source documents that were not included in our enhancement.

In the case of source documents that were not used in our enhancement, the CSSPS can also help increase the percentage of consistency. It demonstrates how shared controls can help increase the CSSPS’s universality while enhancing its ability to be consistent with various source documents. However, the analyst adds that the CSSPS may be inconsistent, lowering the likelihood of conformance with the source documents. For example, Thailand’s PDPA section 31 regulates data controllers’ use of personal data, yet the properties of SSI systems restrict other users from directly manipulating personal data.

We demonstrate an example of the analyst’s justification of the consistency between the sets of system properties to the “error handling and logging” control in [49]. The analyst notes that the CSSPS contains two consistent system properties (i.e., the “accountability” and the “error handling” system properties), whereas the existing set of consolidated system properties contains none.

B. ANALYSIS OF APPLICABILITY IN REAL-WORLD SSI SYSTEMS

This section assesses the CSSPS’s applicability to real-world SSI systems. We define success in terms of application in 2 dimensions: first, that existing real-world SSI systems have already achieved system properties in the CSSPS, and second, that freshly developed SSI systems can achieve system properties in the CSSPS. Regrettably, we cannot locate any newly developed SSI systems when this work is being conducted. As a result, we shall analyze only the first dimension in this section.

The CSSPS, we hypothesized, would be adaptable to current real-world SSI systems. To validate our hypothesis, we undertook another experiment to examine and justify the number of CSSPS system properties that can be incorporated into current real-world SSI systems. Our experiment is designed as a manual analysis session during which we analyze documentation for real-world SSI systems and establish the plausibility of system property possession with justification.

1) EXPERIMENTAL SETTINGS

By adjusting independent variables and measuring dependent variables, we build an experiment to test our hypothesis. Below, we detail each of the settings used in our experiment:

a: DEPENDENT VARIABLE

We are interested in possession of system property in real-world SSI systems as a dependent variable. However, we examine such possessions solely based on the SSI system’s documentation. Occasionally, the documentation is insufficient to ascertain whether the SSI system possesses the required system properties. As a result, we establish three distinct types of system property possessions:

- Yes indicates that the documentation proves that the SSI system possesses the system property.
- Unclear indicates that the documentation did not provide any evidence that the SSI system possesses the system property, but we can justify it based on the SSI system’s functions and internal mechanisms that the SSI system may possess.
- No indicates that the documentation provides some evidence that the SSI system’s functions or internal mechanisms conflict with the system property.

Each system property and real-world SSI system will be assigned one of the three types of possessions.

b: INDEPENDENT VARIABLE

In this experiment, we define two independent variables: the set of CSSPS system properties and real-world SSI systems. We would include all 42 system properties in the CSSPS in the analysis, even if our work did not enhance them.

For the set of real-world SSI systems, we chose three industrial-level products of the SSI system to cover the broad landscape of the real-world adoption: uPort, Sovrin, and IBM Verify Credentials. These SSI systems are different in size, features, and marketing targets.

uPort is an SSI system product that does not adhere to the SSI system’s commonly used open standards, namely DID and the Verifiable Credential data model. Numerous articles [51], [52] have alluded to uPort as a sample of SSI systems. uPort has been phased out and replaced by another product called Veramo. However, it remains an excellent representative of SSI systems with distinct features and technology. We use a white paper by uPort [53] as a data source for our evaluation.

Then, Sovrin is an SSI governance network that provides trustworthy blockchain validators who act as the network’s stewards. Sovrin simply provides the decentralized network and connects implementors’ development applications to it. However, Sovrin suggests an architecture of edge-cloud agents that is compatible with the Sovrin
Table 15. Percentages of consistency and the differences of consistency collected from the experiment session.

| Source | Existing Properties | CSSPS | ΔExistence | TC |
|--------|---------------------|-------|------------|----|
| General Data Protection Regulation [5] | 7 | 14 | 28.57 % | 4 |
| OWASP Application Security Verification Standard [49] | 14 | 8 | 35.72 % |
| Health Insurance Portability and Accountability [21] | 17 | 10 | 11.77 % | 58.82 % |
| Thailand’s Personal Data Protection Act [50] | 14 | 11 | 14.29 % | 78.57 % |

Table 16. Difference in consistency between the existing set of system properties and the CSSPS.

| Control | Justification |
|---------|---------------|
| OWASP ASVS 4.0.2: Error Handling and Logging | The application should collect log containing both sensitive and public information. The application must ensure that the logged information is handled securely and protected along its lifetime. |
| Existing: Not Consistent | - The analyst justifies that there is no system property in the original set that constrains the logging of information. |
| CSSPS: Consistent | - The analyst justifies that the new “Accountability” property in the CSSPS is defined in the way that constrains the logging of information in the SSI system, as well as the new “Error Handling” property is defined in a similar name. |

Note that we provide justifications collected in the “Experiment Resources” folder [10].

Network’s connectivity. Sovrin is a representative of the SSI system that operates under the stringent administration of a decentralized network. We analyzed using data from a white paper [54] provided on Sovrin’s website.

Finally, IBM Verify Credentials (VC) is a comprehensive solution for the IBM Corporation’s SSI system. The IBM VC uses Hyperledger Indy as blockchain technology, similar to Sovrin. On the other hand, the IBM VC provides customers with implementations of the decentralized network and edge-cloud architectures in addition to open standards. This product is an excellent example of a fully functional SSI system built on open standards. We do our analysis using the documentation available on the product’s website [55] as a data source.

2) EXPERIMENTAL PROCEDURE AND FINDINGS
In this section, we describe our experimental procedure and provide our findings in the following manner:

a: EXPERIMENTAL PROCEDURE
First, we analyzed the targeted products’ features and functions utilizing the data source, white papers or online documentation. This step enables us to gain a firm understanding of the targeted products and conduct analysis. Then, for each system property in the CSSPS, we compare it to the targeted products individually to determine whether they possessed the system property via a feature or function. We track which system properties are present in the targeted products and categorize them as Yes, No, or Unclear.

b: FINDINGS
Table 17 summarizes our experimental findings regarding the types of system property possessions. We found two crucial observations as follows:

Finding 3: Of the 42 system properties, 11 are already possessed by all real-world SSI systems, 24 have some unclear possessions, and eight are not possessed or are difficult to possess in some real-world SSI systems.

We can observe that 11 system properties in the CSSPS are marked as Yes in all products (for example, the existence property). This case demonstrates that real-world products already possess those eleven system properties. Following that, we consider 23 system properties indicated by a combination of Yes and Unclear. It indicates that real-world products may not place a premium on documenting such system properties. However, given the existing technology, features, and functions available, we may justify that these system properties have the potential to be included in products. Finally, we can see that those six system properties in the CSSPS are marked as No due to incompatibility with the technology employed or difficulty in adopting such system properties.

Finding 4: Possessions of system properties in the CSSPS may vary to the technology used by the product or may not be presented in the documentation.

Table 18 demonstrates an example of the possession types of the “data recovery” system property in three products. This property is worth focusing on because it was marked as Yes, Unclear, and No in different products. The uPort is the only product that clearly mentions the mechanism to recover the wallet through the recovery contacts. It is adequate to mark uPort as Yes. However, the documentation for Sovrin implies an edge-cloud design, in which personal data is kept on a potentially lost artifact. It is up to implementors to provide procedures for recovering edge agents. We cannot locate such mechanisms in the present documentation. As a consequence, we are classifying this as Unclear. Finally, the IBM VC is listed as No since its edge agents exclusively serve as credential storage. The IBM VC only permits the re-issuance of credentials when the edge agent is lost or destroyed. As a result, data recovery looks to be challenging to execute in the present state of the IBM VC.

VIII. DISCUSSION
A. INCREMENT OF THE DEGREE OF CONSISTENCY
As a major success of the CSSPS, the increase in consistency should be ensured in this work. We reveal advantages in two interesting aspects of the experimental results in Section VII-A.
TABLE 17. Applicability analysis results in three real-world SSI systems.

| Property Name                  | Possession of Properties in | uPort | Sovrin | IBM VC |
|-------------------------------|-------------------------------|-------|--------|--------|
| IP1. Existence                | Yes                           | Yes   | Yes    | Yes    |
| IP2. Sovereignty              | Yes                           | Yes   | Yes    | Yes    |
| IP3. Single Source            | No                            | Yes   | Yes    | Yes    |
| IP4. Standard                 | Yes                           | Yes   | Yes    | Yes    |
| IP5. Cost Free                | Yes                           | Yes   | Yes    | Yes    |
| IP6. Decentralized            | Yes                           | Yes   | Yes    | Yes    |
| IP7. Verifiability            | Unclear                       | Yes   | Yes    | Yes    |
| IP8. Scalability              | Yes                           | Yes   | Yes    | Yes    |
| IP9. Accessibility            | Yes                           | Unclear | Unclear | Unclear |
| IP10. Sustainability          | No                            | Yes   | Yes    | Yes    |
| IP11. Access Control          | Unclear                       | Unclear | Unclear | Unclear |
| IP12. Transparency            | Yes                           | Yes   | Yes    | Yes    |
| IP13. Persistence             | Unclear                       | Unclear | Unclear | Unclear |
| IP14. Portability             | Yes                           | Yes   | Yes    | Yes    |
| IP15. Interoperability        | Unclear                       | Yes   | Yes    | Yes    |
| IP16. Consent                 | Unclear                       | Unclear | Unclear | No      |
| IP17. Data Minimization       | No                            | Yes   | No     | No     |
| IP18. Protection              | Unclear                       | Unclear | Unclear | Unclear |
| IP19. Data Authentication and Protection | Unclear | Yes   | Yes    | Unclear |
| IP20. Physical Authentication and Protection | Unclear | Unclear | Unclear | Unclear |

First, we observed that our property enhancement could explicitly identify consistency between the existing system properties and the shared controls and use the results as information to enhance the system properties. The differences (Δ) in Section VII-A reflect that the CSSPS has enhanced its consistency in all cases. The results also indicate that, in the cases of source documents used to enhance the system properties, the CSSPS can achieve greater than 85% of consistency. Only one control is missing for each source document due to a conflict with the SSI system's fundamental concepts. Additionally, the results ensure that an SSI system achieving the CSSPS is highly likely to comply with applicable laws, regulations, and technical standards.

Second, we observed that deriving shared controls contributes to the universality of the CSSPS. Incorporating missing endorsed tasks in shared controls into system properties enhances the consistency with source documents with information security and privacy controls and evaluation objectives. The experimental findings reported in Table 15 show that the CSSPS enhances approximately 10% of consistency in the case of source documents that were not used in our enhancement. This benefits implementors who are concerned about and seek to enhance the level of information security and privacy in their implementation of the SSI system with minimal effort.

We identified one shortcoming that affects the degree of consistency: the broad definition of controls. A control may specify the tasks for which it is endorsed, which may apply to both software applications and organizations. Incorporating such endorsed tasks into the CSSPS may help ensure the SSI system is implemented in accordance with them, but this does not ensure that auditors will agree that the controls are being followed in all areas.

B. APPLICABILITY IN REAL-WORLD SSI SYSTEMS

The ability to apply the CSSPS in real-world SSI systems is necessary to ensure the CSSPS's adoption. We discovered three noteworthy advantages and a shortcoming of the CSSPS, and we will discuss them in this section.

First, we observed that current SSI systems already influence most system properties in the CSSPS. According to the findings in Section VII-B, 35 of 42 system properties in the CSSPS are considered as Yes or Unclear, indicating that about 84% of them have the potential to be adopted by existing SSI systems.

Second, we observed that system properties designated as Unclear are potentially useful even if they are not documented. The findings indicate that current technologies and architectures of SSI systems may support the CSSPS. It enables implementors to examine the CSSPS and incorporate them into future versions of SSI systems to elicit requirements.
Finally, we discovered that the CSSPS might be used to improve data security and privacy. The CSSPS meets the consistency with source documents pertaining to information security and privacy while allowing adoption in real-world SSI systems, as shown in Sections VII-A and VII-B. This is advantageous to implementors because it enables them to focus their efforts on developing an SSI system and possessing system properties in the CSSPS while also increasing their chance of complying with applicable laws, regulations, and technical standards. When compliance is critical, this should assist them save time and effort.

The CSSPS’s broad constraint definitions are the only shortcoming we identified in terms of application. According to the technology or architecture, implementors must subjectively interpret system properties and express them as technical requirements. The nature of system properties dictates that they act as constraints rather than requirements. It is important to define corresponding system requirements to achieve system properties. We are unable to define system properties precisely in the similar manner to the system requirement elicitation, which results in adoption challenges. This is a trade-off that must be balanced. The broad constraint definition may affect the CSSPS’s accuracy of application.

C. THREATS TO VALIDITY
1) INTERNAL THREATS
We discovered three internal threats to the validity of this work. First, the analysis of the degree of consistency (Section VII-A) participates with only one system analyst, which appears to be a small sample. The experiment is meant to require the analyst to make a comparison between two sets of texts. The conclusions drawn from the analyst’s decisions are straightforward because the two texts are comparable and backed up by concrete facts. The analyst’s opinions had little effect on the outcome. More participants are not required to verify the conclusions, which may not result in significant differences.

Second, the applicability analysis is mostly dependent on the documentation for SSI system products. The implementation may diverge from the documentation. However, we believe that documentation is given to inform end users about the features and functionalities of the product. The documentation should be seen as a promise and should not differ much from the implementation. Our analysis categorizes system property possessions as Yes, No, or Unclear to eliminate erroneous cases. The Unclear type suggests the possibility that we discovered potential in the product’s present architecture and technology. Our justification is offered to demonstrate how the system properties denoted as Unclear may be adopted.

Finally, the shared controls required to enhance the property are retrieved from a specific version of the source documents. Shared controls may become outdated when laws, regulations, and technical standards are regularly changed. We believe that the shared controls were derived and integrated from several source documents, and that not all source documents will be changed at the same time. Furthermore, the concepts of information security and privacy are likely to stay intact. The shared controls should not be affected by modifications made to specific source documents.

2) EXTERNAL THREATS
We detected an external threat to the validity of this work: the analysis of applicability was limited to three real-world SSI systems. However, this threat is mitigated by various product sizes, features, and target markets. We made an effort to incorporate as many real-world SSI systems as feasible. We selected to cover the various sizes with a smaller-sized product (uPort) and a larger-sized product (Sovrin and IBM VC). Additionally, although others do, uPort does not fully embrace the SSI system’s fundamental concepts. This may reflect some differences in the features. This external threat should be addressed during the careful selection process.

IX. RELATED WORK
In this section, we compare our proposal for the CSSPS with other proposals for SSI system principles and properties. In addition, we compare our work to other studies to determine the value of utilizing CSSPS properties.

Table 19 details the comparison between our work and other proposals. As the principles and properties are deemed crucial to the development of SSI systems, there have been numerous proposals to analyze and enhance them. The table indicates that, to our knowledge, five works have been published previously [1], [2], [3], [4], [6].

Allen [1] offered ten basic principles, generally acknowledged as the SSI system’s guiding principles. His proposal, however, is limited to constraining the fundamental concepts. Regarding the degree of consistency, the CSSPS is more compliant with information security and privacy requirements than his proposal, as we incorporate missing endorsed tasks of source documents’ controls. In terms of application, we believe that Allen’s proposals and the CSSPS are equivalent, as several of the system properties in the CSSPS are similar to those in Allen’s. The CSSPS is superior to Allen’s proposal since it addresses information security and privacy concerns and adheres to source documents more closely.

As Allen was the first to propose a set of guiding principles for the SSI system, all subsequent works referenced his proposal, and three of them [2], [3], [6] used Allen’s principles to improve security and privacy. Nevertheless, it is clear that only one of the existing proposals [6] addresses both security and privacy. None of them, however, included compliance with a large number of laws, regulations, and technical standards.

Naik and Jenkins [3] were the first to incorporate concerns about data privacy into the SSI system’s principles. They enhanced Allen’s basic principles to make them consistent with GDPR. The inclusion of concerns about information privacy from the GDPR’s privacy principles in their proposal contains a few references to demonstrate which sections are adopted. The CSSPS is more compliant in term of
consistency, as we included missing endorsed tasks from 28 source documents’ shared controls. This should provide a higher level of coverage than solely conforming system properties to a source document. Their proposal is evaluated against three comparable real-world products. The findings are comparable, implying a similar degree of application. In summary, the CSSPS is more comprehensive than theirs in terms of consistency and shares similar applicability.

In conclusion, the CSSPS could meet the requirements of security and privacy in the SSI system’s principles and properties and cover a broader range of sources, thereby increasing its reliability. The comparison with related works emphasizes that we can place the CSSPS on equal level with other proposals, but the CSSPS has a greater capacity to comply with laws, regulations, and technical standards. The CSSPS could be utilized as a set of evaluation criteria in future research.

X. CONCLUSION
We enhanced the SSI system’s existing properties to increase its likelihood of compliance with applicable laws, regulations, and technical standards. We use a methodical approach to reviewing source documents and determining whether their controls are consistent with existing system properties. We searched through 211 source documents and discovered 28 that are suitable for property enhancement. We extracted 17 shared security and 14 shared privacy controls from the selected source documents. Then, we combined four different proposals for SSI principles and system properties to arrive at a relative set of 20 system properties. We analyzed shared controls and consolidated system properties in order to identify tasks that were not endorsed. We introduced the CSSPS, a collection of 42 system properties that should be consistent with shared controls from source documents.

Through a series of experiments, we evaluated the CSSPS. The findings indicate that the CSSPS increases the percentage of consistency with both source documents involved in our enhancement and with other source documents. On the other hand, the findings reveal that real-world SSI systems can implement and possess over 85% of the CSSPS.

We recommend future directions for this work to address the shortcomings associated with subjective interpretation of technical requirements by specifying and verifying such system properties using some formal methods. Additionally, we look forward to distributing and standardizing the CSSPS. Conforming system properties to shared controls derived from laws, regulations, and technical standards is also applicable to other critical systems and information systems that manipulate PII. It should provide a significant opportunity to improve the security and privacy of other systems.

ACKNOWLEDGMENT
The authors would like to thank the members of the FMSD laboratory for their comments and feedback during their research activities, as well as the participant in the experiment who made a valuable contribution to this research.

DISCLOSURE STATEMENT
The authors report that there is no competing interest to declare.

REFERENCES

[1] C. Allen. Path to Self-Sovereign Identity. Life With Alacrity. Accessed: Jul. 28, 2022. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html

[2] M. A. López. “Self-sovereign identity—The future of identity: Self-soverieignty, digital wallets and blockchain,” 2020. [Online]. Available: https://publications.iadb.org/en/self-sovereignty-future-identity-self-sovereignty-digital-wallets-and-blockchain and https://ieeepublications.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf

[3] N. Naik and P. Jenkins, “Governance principles of self-sovereign identity applied to blockchain enabled privacy preserving identity management systems,” in Proc. 6th IEEE Int. Symp. Syst. Eng., 2020, pp. 1–6.

[4] 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.

[5] (2016). The General Data Protection Regulation. European Parliament and the Council of the European Union. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[6] C. Pattiyanon and T. Aoki. “Analysis and enhancement of self-sovereign identity system properties compiling standards and regulations,” in Proc. 8th Int. Conf. Inf. Syst. Secur. Privacy, 2022, pp. 131–142.

[7] M. Sornyn, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen, (2021). Decentralized Identifiers (DIDs) V.1.0—Core Architecture, Data Model, and Representations. [Online]. Available: https://w3.org/TR/did-core/

[8] M. Sornyn, D. Longley, and D. Chadwick. (2019). Verifiable Credential Data Model V.1.1—Expressing Verifiable Information on the Web. [Online]. Available: https://w3.org/TR/vc-data-model/

[9] Information Technology—Security Techniques—Privacy Framework, Standard ISO/IEC 29100:2011, International Organization of Standardization, 2011.

[10] “Data sheets—Compliance SSI system property set to laws, regulations, and technical standards,” Zenodo, Aug. 2022. [Online]. Available: https://ieeepublications.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf, doi: 10.5281/zenodo.6951404.
List of International Standards, International Organization of Standardization, Accessed: Jul. 28, 2022. [Online]. Available: https://www.iso.org

D. C. Tofan, “Information security standards,” J. Mobile Embedded Distrib. Syst., vol. 3, no. 3, pp. 128–135, 2011.

J. F. Karim, M. Schön, W. Yang, C. Shi, and V. R. Kebande, “A review of security standards and frameworks for IoT-based smart environments,” IEEE Access, vol. 9, pp. 121975–121995, 2021.

IT Security Standards. Wikipedia. Accessed: Jan. 2022. [Online]. Available: https://en.wikipedia.org/wiki/IT_security_standards

Cyber-Security Regulation. Wikipedia. Accessed: Oct. 2021. [Online]. Available: https://en.wikipedia.org/wiki/Cyber-security_regulation

Privacy Law. Wikipedia. Accessed: Jan. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Privacy_law

Information Privacy Law. Accessed: Jul. 28, 2022. [Online]. Available: https://www.iso.org

Wikipedia

Cyber-Security Regulation. Accessed: Oct. 2021. [Online]. Available: https://en.wikipedia.org/wiki/Cyber_security_regulation

Privacy Policy. Wikipedia. Accessed: Jan. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Privacy_policy

Payment Card Industry (PCI) Data Security Standard (DSS)—Requirements and Security Assurance Procedures 3.2.1, Payment Card Ind. Secur. Standards Council, New York, NY, USA, 2018.

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, European Council, London, U.K., Oct. 1995.

Health Insurance Portability Accountability Act of 1996, Centers Medicare Medicaid Services, Independence, MI, USA, 1996.

Information Security—Lightweight Cryptography, Standard ISO/IEC 29102:2019, International Organization of Standardization, 2019.

Information Technology—Security Techniques—Prime Number Generation, Standard ISO/IEC 18032-2005, International Organization of Standardization, 2005.

Personal Data Act (1998:204), Swedish Data Protection Authority, 1998.

Information Technology—Security Techniques—Digital Signature Schemes Giving Message Recovery, Standard ISO/IEC 9796-2:2010, International Organization of Standardization, 2010.

Information Technology—Security Techniques—Entity Authentication, Standard ISO/IEC 9798-2:2008, International Organization of Standardization, 2008.

Security Techniques—Hash-functions, Standard ISO/IEC 10118-1:2016, International Organization of Standardization, 2018.

Information Technology—Security Techniques—Key Management, Standard ISO/IEC 11770-1:2010, International Organization of Standardization, 2010.

Information Security—Non-Reputation, Standard ISO/IEC 13888-2, International Organization of Standardization, 2020.

Information Technology—Security Techniques—Time-Stamping Services, Standard ISO/IEC 18014-1:2008, International Organization of Standardization, 2008.

Information Technology—Security Techniques—Random Bit Generation, Standard ISO/IEC vol. 18031, p. 2005, International Organization of Standardization, 2005.

Information Security—Authenticated Encryption, Standard ISO/IEC 19772:2020, International Organization of Standardization, 2020.

Information Technology—Security Techniques—Information Security Management, Standard ISO/IEC 27001:2013, International Organization of Standardization, 2013.

Information Technology—Security Techniques—Codes of Practices For Information Security Controls, Standard ISO/IEC 27002:2013, International Organization of Standardization, 2013.

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.

Security for Industrial Automation and Control Systems—System Security Requirements and Security Levels, document IEC 62443-3-3, International Electrotechnical Commission, 2013.

Cyber Security Standards—CIP-002-1 through CIP-009-1, North Amer. Electr. Rel. Corp., Atlanta, GA, USA, 2006.

NIST Special Publication 800-12 Revision 1: An Introduction to Information Security, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2017.

Cyber Essentials: Requirements for IT Infrastructure, Nat. Cyber Secur. Centre, London, U.K., 2021.

The OECD Privacy Framework, Org. Econ. Co-Operation Develop., Paris, France, 2013.

UN Personal Data Protection and Privacy Principles, United Nation High-Level Committee Manage. (HLCM), Bonn, Germany, 2018.

TOSHIKI AOKI received the B.S. degree from the Science University of Tokyo, in 1994, and the M.S. and Ph.D. degrees from the Japan Advanced Institute of Science and Technology, in 1996 and 1999, respectively. He is currently a Professor with the Division of Transdisciplinary Sciences and the School of Information Science, JAIST. His research interests include formal methods, formal verification, theorem proving, model checking, automotive systems, and embedded systems.

VOLUME 10, 2022

CHARNON PATTIYANON was born in Bangkok, Thailand, in 1991. He received the B.Eng. degree in computer engineering (international program) from the King Mongkut’s University of Technology Thonburi, Bangkok, in 2014, and the M.S. degree in software engineering from Chulalongkorn University, Bangkok, in 2017. He is currently pursuing the Ph.D. degree in information science with the Japan Advanced Institute of Science and Technology (JAIST), Noimi, Ishikawa, Japan.

He was a Senior Technical Leader with the Department of Identity and Access Management, Security Division, G-Able Company Ltd., from 2017 to 2019. His research interests include software engineering, identity and access management, information security and privacy, natural language processing, and formal methods.

The Information Technology Act, 2000 (No. 20 200), Legislative Dept., Gazette India, Ministry Law, Justice Company Affairs, New Delhi, India, 2000.

The Personal Data Protection Code of Practice, Personal Data Protection Commissioner, Singapore, 1998.

Repub. Act 10173 - Data Privacy Act 2012, Nat. Privacy Commission, Pasay, Philippines, 2012.

Federal Law No.152-FZ of July 27, 2006 On Personal Data, Fed. Council, Russia, 2006.

Personal Data Protection Act, Nat. Develop, Council, Taiwan, 2015.

Children’s Online Privacy Protection Act of 1996, Federal Trade Commis- sion, Washington, DC, USA, 1998.

Personal Information Protection Law of the Mainland, Standing Commit- tee Nat. People’s Congr., Beijing, China, 2021.

OWASP Application Security Verification Standard 4.0.2, Open Web Appl. Secur. Project (OWASP) Found., Annapolis, MD, USA, 2020.

Personal Data Protection Act (PDPA), B.E. 2562, Cabinet Parliament Thailand, Roy. Thai Government Gazette, Thailand, 2019.

Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K. K. R. Choo, “Blockchain-based identity management systems: A review,” J. Netw. Comput. Appl., vol. 166, Sep. 2020, Art. no. 102731.

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. Cham, Switzerland: Springer, 2020, pp. 3–13.

C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. (2016). uPort: A Platform for Self-Sovereign Identity. [Online]. Available: http://blockchainlab.com/pdf/uPort_whitepaper_DRFA20161020.pdf

A. Tobin and D. Reed, The Inevitable Rise of Self-Sovereign Identity: A White Paper From the Sovrin Foundation [White Paper]. Accessed: Jul. 28, 2022. [Online]. Available: https://sovrin.org/wp-content/uploads/2017/06/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf

IBM Corporations. IBM Verify Credentials. Accessed: Jul. 28, 2022. [Online]. Available: https://docs.info.verify-creds.com

**