SecureOnt: A Security Ontology for Establishing Data Provenance in Semantic Web

Archana Patel[1,*], Narayan C. Debnath[1] and Prashant Kumar Shukla[2]

[1]*Department of Software Engineering, School of Computing and Information Technology, Eastern International University, Vietnam*
[2]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India*
E-mail: archana.patel@eiu.edu.vn; narayan.debnath@eiu.edu.vn; prashantshukla2005@gmail.com
[*]*Corresponding Author*

Received 18 January 2022; Accepted 05 March 2022;
Publication 18 April 2022

Abstract

Security becomes a primary concern during sharing of information over the web. To overcome this problem, many security ontologies have been developed so far. The available security ontologies help to track data provenance and contain different aspects of data security like confidentiality, integrity, data availability, and access control. This paper provides a security ontology for establishing data provenance in the semantic web. The proposed ontology contains a comprehensive knowledge base of data security by consolidating all the available security ontologies and derives data provenance with annotations at the extensional level and thus lower maintenance cost. By this paper, analysts and researchers find a road map, an overview of what exists in terms of security ontologies.

Keywords: Security ontology, data provenance, ontology evaluation tools, semantic web, ontology richness, anomalies, pitfall rate.

Journal of Web Engineering, Vol. 21_4, 1347–1370.
doi: 10.13052/jwe1540-9589.21415

1 Introduction

An ontology represents real-world entities in a machine-understandable manner. Many security ontologies have been developed for secure communication and contain information about different aspects of data security. The main characteristics of security are confidentiality (allowing only authorized users to access the information), integrity (only authorized users must alter the information), and availability (the information must be available as per user request), data provenance (description of the origin of data and process by which data was produces) [1]. The available security ontologies can hold the different characteristics of the security. These ontologies play a vital role in securing the information on the semantic web. The vision of the semantic web is to link the data instead of documents and offers the software program with metadata that allows the semantic interpretation of data and information. All the layers of semantic web architectures add some security features that ensure the overall security of information on the web. However, the current architecture is lacking provide the complete security aspects. To achieve the vision of the semantic web, various languages (like DAML, RDF, RDFS, OWL, etc.) have been developed to encode the ontology. OWL-S language is developed for the security aspects, and many security ontologies utilize the constructs or operators of this language.

Ontologies are either developed by handcraft or by using ontology development tools. The OWL-S protégé editor provides an environment for creating and maintaining OWL-S service descriptions. It is open-source and can be integrated with the protégé tool, which is an open-source knowledge base tool. Soug et al. [2] have classified the security ontologies in eight families, namely: Beginning Security Ontologies, Security Taxonomies, General Security Ontologies, Specific Security Ontologies, Web Oriented Security Ontologies, Risk based Security Ontologies, Ontologies for security requirements, and Security Modelling ontologies. All these security ontologies are evaluated either by the expert of the domains or by ontology evaluation tools. The evaluation of an ontology shows the richness in terms of ontological elements (classes, properties, instances, etc.), correctness (detect the existing anomalies), and consistency (shows that ontology does not have any contradictions). The important criteria of ontology evaluation are Accuracy, Clarity, Completeness, Adaptability, Consistency, Computational efficiency [3]. Blanco et al. [4] discussed the richness of the four security ontologies by using OntoMetrics tool.

However, none of the available papers focus on the anomalies detections and pitfall rate of the security ontologies; also, there is no other ontology containing a comprehensive knowledge base about data security that focuses on data provenance. The five questions that have not been addressed so far are also the subject of this paper. These questions are- How many security ontologies are available?, What are the different types of security ontologies?, Which ontology is good in terms of richness?, What is the pitfall rate of the available security ontologies?, How to make a security ontology that integrates all the existing security ontologies and track data provenance? The organization of this paper is as follows: Section 2 provides a glance at security aspects of data, data provenance with ontology and secure semantic web. Section 3 describes the available articles and ontologies related to security. Section 4 shows the evaluation of the security ontologies developed for the semantic web. Section 5 discusses the proposed approach that merges the existing security ontologies. Finally, the last section concludes this paper.

2 Security Aspect of Data

Data security becomes one of the most daunting tasks and top concerns for researchers and IT professionals, as a large amount of data is being shared on the web via the internet. Data security has myriad aspects that protect the information at rest, in motion, and use. Confidentiality, integrity, data availability, access control, and data provenance (also known as data leakage) are essential requirements to achieve data security [5].

- Confidentiality (C): The confidentiality of data means that data should be accessed by authorized parties only. The confidentiality of the semantic web data is usually achieved with data encryption (cryptographic operations).
- Integrity (I): Data integrity refers that data should be modified or altered by only authorized parties. The integrity of semantic web data is typically achieved with digital signatures.
- Availability (A): Data availability means that data must be available and accessible when requested by the user. This is generally achieved by redundant data storage. Most of the time, two languages, namely modular and extensible policy language are used for the same.

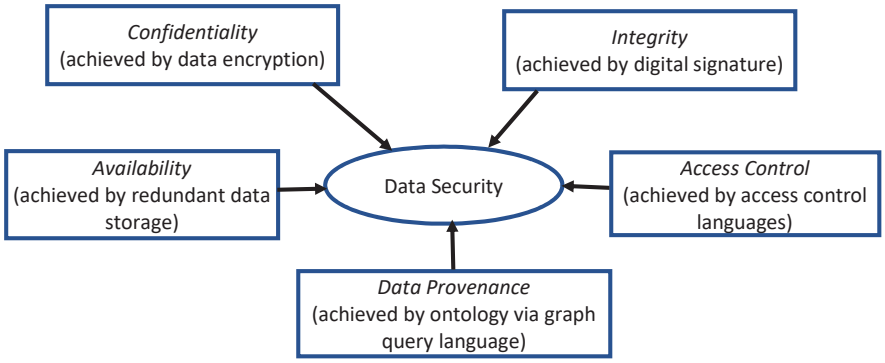

Figure 1 Data security aspects.

- Access Control (C): Access control provides a set of methods that regulates access to data. Two languages, namely Enterprise Privacy Authorization Language and the eXtensible Access Control Markup Language are XML-based control languages that provide rich, expressive policies compared to other access control languages.
- Data Provenance (P): Provenance of a resource is a record that defines the process and entities involved in producing and delivering or otherwise influencing that resource [6]. The credibility criteria that mean 'to find the origin of the data' leads to the provenance of data. Ontologies are used to track the provenance of data as they represent data in the graph format (subject-object-predicate) and connect all the data [7]. SPARQL query language uses to extract the information from the ontologies.

Figure 1 shows different aspects of data security, and all these aspects are collectively referred to as the 'CIACP'. These aspects of data security arise additional questions that need to be addressed –

- How can confidentiality of data be ensured on the web so that only authorized users can access the data?
- How can data integrity be achieved on the web so that any unauthorized modification of the data is detected?
- How can the authenticity of data be ensured on the web so that only data can be collected from verified sources?
- How can we provide access control mechanisms of data on the web so that only some part of the data or required data is displayed to the users?

- How can we track data provenance on the web so that unauthorized transmission of data from within a source to a destination can be detected?
- How can data be made available in open networks to parties whose access requests comply with predefined and transparently communicated policies?

(a) Ontology to Track Data Provenance: Ontology provides a way to present the knowledge in a machine-understandable manner by (a) listing the concepts (classes) of a domain and (b) adding the meaning (semantic) of these concepts by associating appropriate properties (data and object properties) and relations [8]. It represents knowledge in triple format. Suppose we have a concept, namely, Mona Lisa, and now try to build its knowledge. The following statements contain more concepts about the Mona Lisa:

- Mona Lisa is in Louvre Museum
- Louvre Museum is in Paris
- Louvre Museum is the same as Great Louvre
- French government take care of Great Louvre
- Great Louvre is near the Tuileries Gardens

These statements (facts) are represented in Figure 2(a). This graph depicts the statements in the form of A→B, where A and B are the concepts, and the arrow → explains the relationship that exists between A and B. Like, the statement Mona Lisa $\xrightarrow{is\_in}$ *Louvre Museum* explains that the is_in relationship exists between Mona Lisa and Louvre Museum. The connected graph is a powerful way of representing the concepts of a domain and simulating human knowledge. However, the meaning (semantic) of the relations that connect two concepts is still missing. Consequently, the machine cannot do reasoning, track data provenance and infer new knowledge as a human can do by applying logic. By such representation, a machine cannot answer simple questions like Is Louvre Museum in France? Is Mona Lisa in France?

By converting this graph into an ontology, the machine can do reasoning as human beings do and track the data provenance. The first step to transforming the connected graph into an ontology is adding is_a relationship to the concepts to mark them as a class and add the semantics later on. Figure 2(b) depicts five new classes namely Painting, Public Museum, City, Country, Public Garden, and Constitutional body. All concepts are mapped to the class and are called an instance of that class. For example, Mona Lisa

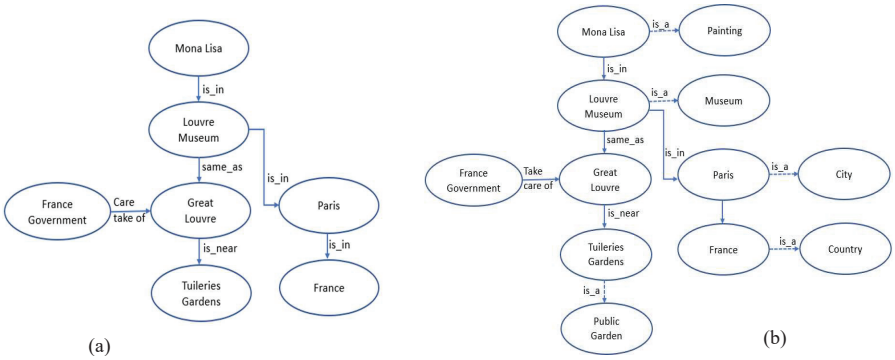

Figure 2 Semantic representation: ontology to track data provenance.

is an instance of a class Painting. Defining a class allows us to add more meaning to the concept by assigning relations that connect a class to another class. All instances of connected classes must hold the same relationship that exists between the classes and vice versa. For example, suppose a class Painting and class Country is related as $Painting \xrightarrow{is\_in} Country$, then from Figure 2(b), the machine can deduce that Mona Lisa is in France because Mona Lisa and France are instances of class Painting and Country respectively. By applying the logic to the existing relations presented in Figure 2(b), the machine can infer more relations, for example: Painting $\xrightarrow{is\_in}$ *City*, Painting $\xrightarrow{is\_in}$ Country, Museum $\xrightarrow{is\_in}$ *City*, Museum $\xrightarrow{is\_in}$ Country, Constitutional body $\xrightarrow{take\ care\ of}$ Public Garden. In addition, ontology allows us to add semantics to the relations. For example, a relation can be defined as transitive, disjoint, reflexive, complementary, and many more. All relationships that are assigned to the class must be automatically transferred to the instances of those classes. With the help of these relations, ontology infers more knowledge similar to human beings.

As ontologies represents semantic relationships among entities and grouped them based on different dimensions of the reality. Tuples of entities, called ontological coverages, express the scopes of data sets and granularities of data values in these dimensions. The semantic relationships between these entities induce a partial order among ontological coverages. This order is used to correlate scopes and granularities of data, enabling an estimation of data provenance [8–9].

(b) Secure Data Management in the Semantic Web: The semantic web aims to provide a decentralized network of machine-readable data by

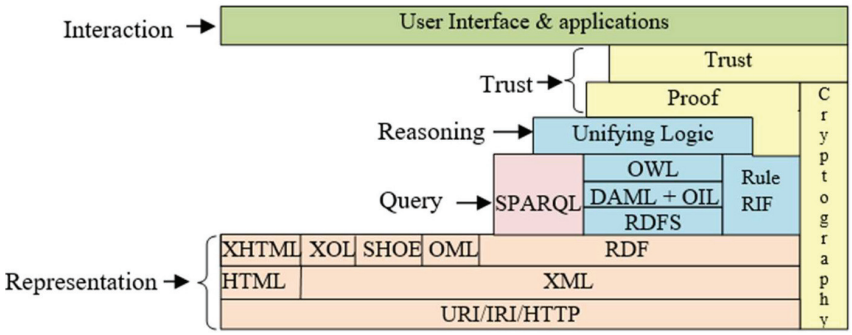

Figure 3 Semantic web layer cake.

associating meaning with the data and adding these above characteristics (confidentiality, integrity, availability of data, access control mechanism, and data provenance). Hence, the successful implementation of these data characteristics must be achieved by corresponding security mechanisms. However, the currently available reference architecture (semantic web layer cake) does not define any specific security mechanism that successfully implements these requirements. The semantic web layer cake defines the architecture (shows in Figure 3) of the semantic web and widely refers to when implementing semantic web applications [10]. The architecture divides the semantic web technologies into different disjoint layers and depicts the interdependencies of their layers. The different layers of this architecture from the upward hierarchy are URI/IRI/HTTP, XML, RDF, RDFS, DAML+OIL, OWL (ontology), Rule RIF/SWRL, SPARQL (query), Unifying logic, Proof, Crypto, Trust and User Interface & Applications.

- URI/IRI/HTTP: The information of this layer can be secured by using secure data communication protocols of TCP/IP (using acknowledgments and retransmissions), HTTPS (using SSL or TLS encryption).
- XML: It provides a standard syntax for the storage of structured information on the web. It can be secured by using different security standards like XML signature and encryption.
- RDF and RDFS: This provides a framework for encoding the information about the resources in RDF graphs (S-O-P). RDF-S is the extension of RDF, which added schema as well. To secure the RDF means secure the semantic and interpretation of data which is one of the challenging tasks. This layer verifies that data access is authentic and correct by signing the RDF graphs.

- OWL, Rule RIF/SWRL, and SPARQL: This layer captures the ontologies, which are the documents of OWL. OWL-S (previously known as DAML-S) is used to annotate the security metrics for the semantic web services. The SPARQL is a query language developed to query over the ontologies and maintains data provenance [11]. RIF and SWRL are the rule languages that use to interchange rules on the web.
- Crypto: This layer implements different cryptographic operations like digital signatures and data encryption. The information exchanged can be kept confidential and can be verified based on granted permission. However, all the security mechanisms are not based on cryptographic operations.
- Unifying logic, Proof, and Trust: These layers are not standardized technologies. Below the layer, all the layers extract the information from the web according to defined unified logic. The proof layer provides justification to agents on 'why the deduced result should be believed'. The reliable inputs from trusted sources, along with formal proof, provide complete confidence in the results.

3 Security Ontologies in Nutshell

As of now, many articles about security ontologies are available in the different repositories. We have queried four well-reputed repositories, namely Science Direct, ACM Digital Library, IEEEXplore, and google scholar. The total number of obtained articles from period 2010 to September 2021 from these repositories – Science Direct has 11,504 articles; ACM Digital Library has 132,986 articles; IEEEXplore has 1,764 articles, and Google scholar has 4,148 articles. Figure 4 shows the screening of the articles that have been done manually, and we have taken two months to complete this screening. Initially, we have got a total of 1,50,402 articles from the four repositories described above. The refined number of articles is 132,986 because 17,416 articles are found duplicates (out of 1,50,402) that have been excluded from the review. After reading the abstract of the obtained articles, we have seen only 1,764 articles where security ontology is the main topic.

Our focus is to cover all the security ontologies that have been developed in the context of the semantic web. To the best of our knowledge, 11 ontologies have been developed for the security aspect of the semantic web that support confidentiality, integrity, data availability, data provenance, and access control mechanisms. These ontologies are- Agent Security Ontology, Credential Ontology, OWL-S (Profile Ontology, Process Ontology, Grounding

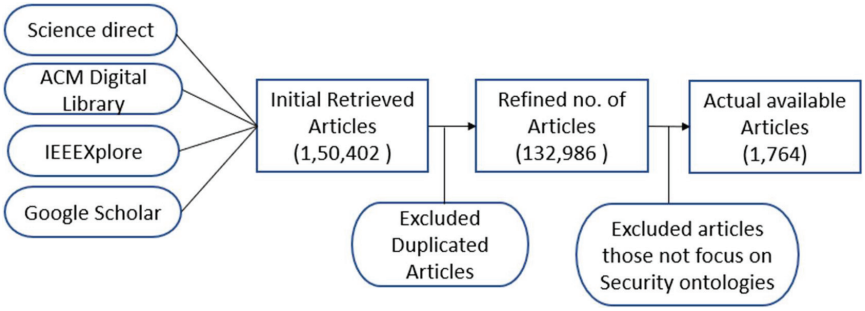

Figure 4 Screening of the articles.

Ontology), Information Object Ontology, Main Security Ontology, Security Ontology, Security View Ontology, Service Ontology, and Service Security Ontology. These ontologies are summarized below:

- **Credential Ontology** [12]**:** Credential ontology aims to provide access to the web pages or services only to the authorized request. It defines those credentials that are commonly used in internet security. The main class of this ontology is credential (top class), which has one subclass, namely ComposedCredential, which is three subclasses: IDCard, Smart-Card, and SimpleCredential. The SimpleCredential has four subclasses: Login, Cookie, BioMetric, Certificate, OneTimePassword, and Key. All subclasses are pairwise disjoint like class key has two disjoint sub-classes, namely public key, and symmetric key. The certificate class has subclass 'X509Certificate', which has a specific class of X509 certificates in the XML Signature.
- **OWL-S** [13]**:** It is a web ontology language that provides essential constructs for web services. To describe the web services, it offers three ontologies, namely

  (a) *Profile Ontology*: It provides a higher level of description of services.
  (b) *Process Ontology*: It describes the detailed description of service's operation.
  (c) *Grounding Ontology*: It specifies the details about how an agent can access the services.

  The profile, process, and grounding define the services in terms of 'what the services do', 'how to use it', and 'how to interact with it', respectively.

- **Agent Security Ontology** [14]**:** This ontology allows to query of the web resources based on the requestor's requirements and capabilities. It has an agent class that presents the service requestor. This class has two properties, namely securityRequirement and securityCapability, that take values from classes, namely SecurityObjective and SecurityConcept.
- **Information Object Ontology (InfoObj)** [15]**:** This ontology is designed to capture web services' encrypted input and output data. It has a class Information Object (InfObj) that has two subclasses, signed Information Object (SigInfObj) and Encrypted Information Object (EnclnfObj). The class InfObj is marked as a range for input/output parameters of services mentioned in OWL-S. It uses the cryptoAlgUsed property to specify the algorithm that used in encryption.
- **Main Security Ontology** [14]**:** This ontology is designed for the security purpose of having information about the assets, threats, countermeasures, and vulnerabilities. It also contains information about defense strategies like prevention and detection as well as security goals such as confidentiality, integrity, etc.
- **Security Ontology** [13]**:** It describes a high level of abstraction of security mechanisms by accommodating various security standards and notations. The class SecurityMechanism has six main subclasses: Syntax, Signature, Encryption, Protocol, KeyFormat, SecurityNotation, and Protocol, whereas the Protocol class has two subclasses: KeyProtocol and DataTransferProtocol. The KeyProtocol has three subclasses, namely KeyRegistrationProtocol, KeyInformationProtocol, and KeyDistributionProtocol. Many properties and instances are also defined as per the requirement of the security mechanism.
- **Security View Ontology** [14]**:** This ontology imports the main security ontology to provide the view of security. The defined classes sort threats and countermeasures according to assists, defense strategies, and security goals.
- **Service Ontology** [16]**:** It defines general concepts of service. Service ontology uses three vocabulary GoodRelations, Schema.org and FOAF. There are three main classes, namely ServiceProvider (provides the service), ServiceConsumer (consumes the service), and ServiceLimitation (imposes the constraints on the services), which are used to define the service.
- **Service Security ontology** [17]**:** This ontology has extended the OWL-S profile ontology. The ServiceParameter class has two subclasses,

namely SecurityConcept and SecurityObjective; these classes are taken from the security ontology. It has a property, namely serviceParameter, which has two properties, securityRequirement, and secuirtyCapability. These properties describe the security requirements and capabilities in its service description. All the subproperties are associated with range and domain that define the security requirements and capabilities in terms of security objective or mechanism.

Apart from these ontologies, three ontologies, namely Security Algorithm ontology, Security Assurance Ontology, and NRL Security Ontology, are also available. However, they are not accessible because not available on the web.

4 Evaluation of Security Ontology

This section shows the evaluation results of the security ontologies based on four tools, namely OntoMetrics, OOPS!, W3C RDF Validation, and OWL2 validator.

- **OntoMetrics Tool:** It is a web-based tool that calculates the statistical information about an ontology. It has five types of metrics, namely Base metrics (show the quantity of ontology elements), Schema metrics (address the design/schema of an ontology), Knowledge base metrics (measure the amount of knowledge), Class metrics (measure the classes and relationships of an ontology), and Graph metrics (calculates the structure of an ontology) [8]. We have calculated base metrics, schema metrics, knowledgebase metrics, and graphs metrics of the security ontologies. We did not calculate the class metrics because all the important information about the classes is already available in other metrics. Table 1 shows the value of these metrics for available security ontologies, which are listed above. The mainSecurity ontology contains highest number of classes (460); Process ontology contains highest number of object properties (42); Grounding ontology contains highest number of data properties (14); whereas Security ontology contains highest number of instances (54), among the available security ontologies.
- **Ontology Pitfall Scanner! (OOPS!):** It is a web-based tool that shows the pitfalls or anomalies of an ontology. OOPS! shows the 41 types of pitfalls starting from P01 to P41. Basically, OOPS! groups the pitfalls under three categories, namely minor pitfalls (these pitfalls are not serious and no need to remove them), important pitfalls (not

Table 1 Richness of Security Ontologies via Ontometric Tool (a) Security Ontologies namely Agent security, Credential, Grounding, InfObj, MainSecurity, Process, Profile (b) Security, SecurityView, Service, ServiceView

Ontologies→		Agent		Grounding		main		
Metrics↓		Security	Credential	Ontology	InfObj	Security	Process	Profile
Base Metrics	Axioms	9	146	144	31	1953	322	70
	Logical axioms count	4	83	74	16	1131	180	40
	Class count	3	30	13	13	460	41	11
	Object properties	2	4	13	2	29	42	10
	Data properties	0	11	14	0	1	9	9
	Instances	0	0	0	6	30	4	0
Schema metrics	Attribute richness	0.0	0.366667	1.076	0.0	0.002	0.219	0.818
	Inheritance richness	0.0	1.2	1.384	0.384	1.630	1.487	0.818
	Relationship richness	1.0	0.454545	0.419	0.285	0.25	0.483	0.526
	Attribute class ratio	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Equivalence ratio	0.0	0.0	0.0	0.0	0.278	0.146	0.0
	Axiom/class ratio	2.666	4.866667	11.076	2.384	4.245	7.853	6.363
	Inverse relations ratio	0.0	0.0	0.0	0.0	0.448	0.047	0.0
	Class/relation ratio	1.5	0.454545	0.419	1.857	0.46	0.347	0.578
Knowledgebase	Average population	0.0	0.0	0.0	0.461	0.065	0.097	0.0
Metrics	Class richness	0.0	0.0	0.0	0.384	0.010	0.048	0.0
Graph metrics	Absolute root cardinality	3	6	6	8	121	14	8
	Absolute leaf cardinality	3	21	8	11	281	20	8
	Absolute sibling cardinality	3	30	11	13	373	25	9
	Absolute depth	3	88	16	18	2386	43	10
	Average depth	1	2.933	1.454	1.384	3.873	1.72	1.111
	Maximal depth	1	5	2	2	8	3	2
	Absolute breadth	3	30	11	13	616	25	9
	Average breadth	3	3.0	2.75	4.333	4.219	4.166	4.5
	Maximal breadth	3	6	6	8	121	14	8
	Ratio of leaf fan-outness	1	0.7	0.615	0.846	0.610	0.487	0.727
	Ratio of sibling fan-outness	1	1.0	0.846	1.0	0.810	0.609	0.818
	Tangledness	0	0.2666	0.307	0.0	0.267	0.146	0.090
	Total number of paths	3	30	11	13	616	25	9
	Average number of paths	3	6.0	5.5	6.5	77.0	8.333	4.5

(a)

Table 1 Continued

Ontologies→ Metrics↓		Security	SecurityView	Service	ServiceSecurity
Base Metrics	Axioms	148	55	55	10
	Logical axioms count	93	34	23	5
	Class count	57	30	5	2
	Object properties	6	2	11	3
	Data properties	0	0	0	0
	instances	54	0	0	0
Schema metrics	Attribute richness	0.0	0.0	0.0	0.0
	Inheritance richness	0.228	0.633	0.4	0.5
	Relationship richness	0.606	0.472	0.846	0.75
	Attribute class ratio	0.0	0.0	0.0	0.0
	Equivalence ratio	0.245	0.5	0.0	0.0
	Axiom/class ratio	2.596	1.833	11.0	5.0
	Inverse relations ratio	0.0	0.0	0.363	0.0
	Class/relation ratio	1.727	0.833	0.384	0.5
Knowledgebase	Average population	0.947	0.0	0.0	0.0
Metrics	Class richness	0.210	0.0	0.0	0.0
Graph metrics	Absolute root cardinality	44	11	3	1
	Absolute leaf cardinality	51	20	3	1
	Absolute sibling cardinality	57	30	3	2
	Absolute depth	76	69	3	3
	Average depth	1.333	2.3	1.0	1.5
	Maximal depth	4	5	1	2
	Absolute breadth	57	30	3	2
	Average breadth	8.142	2.727	3.0	1.0
	Maximal breadth	44	11	3	1
	Ratio of leaf fan-outness	0.894	0.666	0.6	0.5
	Ratio of sibling fan-outness	1.0	1.0	0.6	1.0
	Tangledness	0.0	0.0	0.0	0.0
	Total number of paths	57	30	3	2
	Average number of paths	14.25	6.0	3.0	1.0

(b)

Table 2 Obtained pitfalls (a) important pitfalls (b) Minor Pitfalls (c) Critical Pitfalls

Pitfalls → Ontologies ↓	P10	P11	P12	P24	P25	P26	P30	P34	P41
AgentSecurity	1	×	×	×	×	×	×	×	1
Credentail	×	7	×	×	×	×	×	×	1
Grounding	×	22	×	7	×	×	×	×	1
InfObj	1	×	×	×	×	×	×	×	1
MainSecurityOntology	×	×	×	1	1	1	2	2	×
Process	×	16	×	5	×	×	×	×	1
Profile	×	26	5	5	×	×	×	×	1
Security	1	×	×	×	×	×	×	×	1
SecurityViewOnt	1	×	×	×	×	×	×	×	1
Service	1	×	×	×	×	×	×	×	1
ServiceSecurity	1	3	×	×	×	×	×	×	1

(a)

Pitfalls→ Ontologies ↓	P02	P04	P08	P13	P20	P22
AgentSecurity	×	1	5	2	×	×
Credential	×	1	41	4	×	×
Grounding	2	4	127	53	×	1
InfObj	×	5	14	2	×	×
MainSecurity Ontology	27	1	468	8	1	1
Process	2	4	94	42	×	1
Profile	2	4	115	52	×	1
Security	×	13	62	6	×	1
SecurityViewOnt	×	2	23	×	×	×
Service	×	2	23	×	×	×
ServiceSecurity	×	1	5	3	×	×

(b)

Pitfalls → Ontologies ↓	P31
AgentSecurity	×
Credential	×
Grounding	1
InfObj	×
MainSecurity Ontology	6
Process	1
Profile	1
Security	×
SecurityViewOnt	×
Service	1
ServiceSecurity	×

(c)

very serious pitfalls but need to remove), critical pitfalls (these pitfalls hamper the quality of an ontology and need to remove them before using ontology) [8]. Table 2 shows the obtained pitfalls of the available security ontologies. The mentioned number indicates the total number of cases corresponding to each pitfall. Only five ontologies, namely main security, grounding, profile, service ontology, and process ontology have critical pitfalls that need to be removed before using them

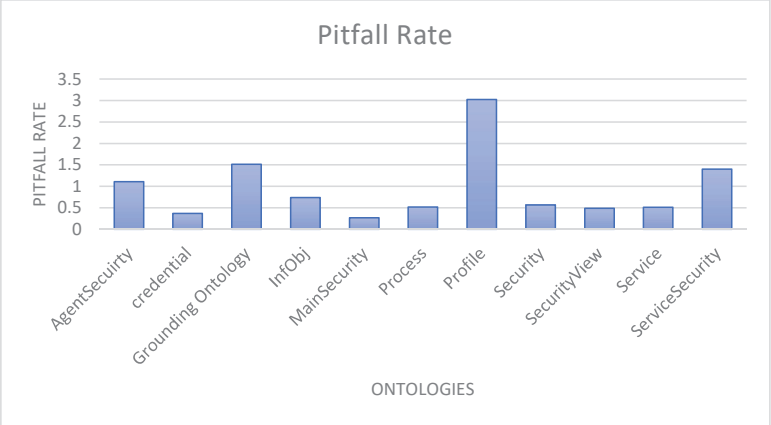

Figure 5 Pitfall rate of security ontologies.

in any application. The description of obtained pitfalls is mentioned on http://oops.linkeddata.es/catalogue.jsp.

The pitfall describes the number of features that could create problems during reasoning. We have calculated the pitfall rate by using the following formula –

$$\frac{\sum_{i=1}^{n} P_i}{N}$$

P_i represents the total number of pitfall cases according to the pitfall type P_i, and N is the total number of tuples (ontology size). The high value of the pitfall rate implies a more significant number of anomalies and vice-versa. Figure 5 shows the pitfalls rate of the available security ontologies. Four ontologies have pitfalls rate higher than 1.

- W3C RDF Validation: This tool is used to validate the RDF document [8]. It tracks the RDF issues and displays a warning message when an error occurs. This tool shows the graphical representation of the ontology and also calculates the number of tuples available in the ontology. We have run this tool over all the available security ontologies, and obtained results show that developed security ontologies are syntactically correct.
- OWL2 Validator: This tool validates the ontology against different profiles of the OWL2 like OWL2RL, OWL2DL, OWL2QL, and OWL2EL. This validator ensures that the concepts and properties of the ontologies must be defined according to the W3C standard. We have successfully passed through all the security ontologies via this tool.

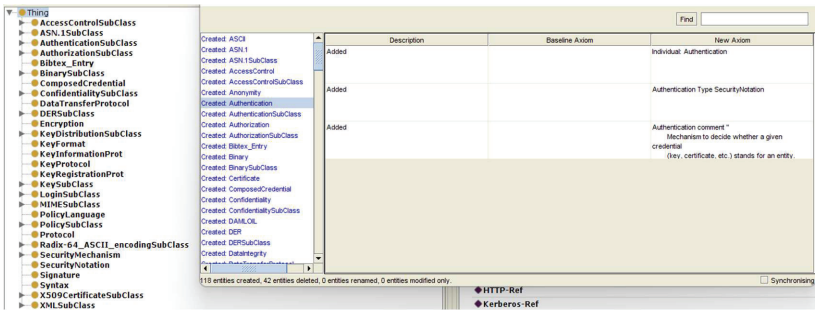

Figure 6 Comparing security ontology and grounding ontology.

5 A Simplified Schema Matching Approach

Many security ontologies have been developed for the semantic web; however, no alone ontology contains sufficient information. There are two solutions of this problem, either develop a new ontology that contain all aspects of security or merge all the existing security ontologies. We follow the second approach and merge all the available security ontologies, which provides an ontology called SecureOnt ontology. The SecureOnt ontology contains a comprehensive knowledge base of data security and derives data provenance with annotations at the extensional level. We propose a semi-automatic Schema Merging Approach (SMA-SecureOnt) that integrates all existing security ontologies. The proposed approach is implemented in Python programming language. This approach has three modules namely Ontology Comparison, Relationship Identification, and Ontology Merging.

(1) Ontology Comparison: Ontology comparison module compares two ontologies and shows the information about the number of entities that has been created, deleted, renamed, and modified with respect to other ontology. We use protégé tool for this purpose. It has a plugin that take ontologies and compare them based on four parameters namely ignore refactors, conservative about identifying refactors operations, make a reasonable attempt to find refactor operations, and apply all algorithm. The Figure 6 shows the comparison result of two ontologies namely security and grounding ontology. The comparison process created 118 entities and deleted 42 entities. In the same way, all the security ontologies are compared with each other.

(2) Relationship Identification: The second module of the SMA-SecureOnt approach is to identify the relationship between the entities. We propose an algorithm that select two ontologies (O_1 and O_2) from the ontology repository and then extract all the labels of the concepts in both ontologies with the help

of Id and IRI. The labels of the concepts of source and target ontologies are stored in a x, y dimensional array (a[x] and b[y]) separately where, x = number of classes in O_1 and y = number of classes in O_2. The SMA-SecureOnt algorithm picks one label of O_1 and then matches it with all the labels of O_2. The matching between the labels is performed according to the Levenshtein and synonym matchers (we import python library and extract the words from nltk library). If two labels are matched based on synonyms, then a 0.9 similarity value will be assigned to them. The matching result (similarity value between the labels) is stored in the x $\times$ y matrix (Avg[x][y]). All the pairs whose similarity value is greater than α (where α is a threshold for the similarity value) are considered to be correspondences. During experiments, we have set parameter $\propto = 0.8$. The extracted relationship are three types –

- Equivalence relationship ($=$): It shows that both entities are same, hence must be merged in the resultant ontology. For example, the class Human $\in O_1$ is same as class human $\in O_2$ (Human = human).
- Subclass relationship ($\subseteq$): It shows that one entity is subset or superset of other entity, hence must be link with the subclass or superclass relationship. For example, the class Human $\in O_1$ is a subset of class Animal $\in O_2$ (Human $\subseteq$ Animal).
- Disjoint relationship ($\perp$): It shows that both entities are disjoint i.e. no relationship exist between them. For example, the class Human $\in O_1$ has no relationship between class NonlivingThing $\in O_2$ (Human $\perp$ NonlivingThing).

Pseudocode: Relationship Identification

- Select O_1 and O_2 from the ontology's repository
- Fetch all labels of both O_1 and O_2 ontologies

```
for (i=1, i ≤ x, i++)   // x∈ number of classes in O1
    fetch Id of the ith concept
        if (Id has label)
            fetch and store label in a[i]
        else
            fetch IRI of the ith concept
            and find label by splitting IRI
            store label in a[i]
```

```
for (j=1, j ≤ y, i++)   // y∈ number of classes in O2
    fetch Id of the jth concept
        if (Id has label)
            fetch and store label in b[j]
        else
            fetch IRI of the jth concept
            and find label by splitting IRI
            store label in b[j]
```

- Matching

```
for (i=1, i<x, i++)
    for (j=1, j<y, j++)
        S1 ← Run synonym matcher over a[i] and b[j]
        S2 ← Run Levenshtein matcher over a[i] and b[j]
        Avg[i][j] ← Average of S1 and S2
Display all the pairs where Avg[i][j] ≥ α
```

Results for P04: Creating unconnected ontology elements.	6 cases | Minor
Results for P07: Merging different concepts in the same class.	1 case | Minor
Results for P08: Missing annotations.	236 cases | Minor
Results for P10: Missing disjointness.	ontology* | Important
Results for P11: Missing domain or range in properties.	1 case | Important
Results for P13: Inverse relationships not explicitly declared.	35 cases | Minor
Results for P22: Using different naming conventions in the ontology.	ontology* | Minor
Results for P30: Equivalent classes not explicitly declared.	2 cases | Important
Results for P41: No license declared.	ontology* | Important

Figure 7 OOPs Results of SecureOnt ontology.

(3) Ontology Merging: Ontology merging is performed based on the identified relationships between the entities. We take two ontologies and extracts all the classes, properties and instances. Matching module first matches the classes then properties followed by the instances. The used set of matchers (Levenshtein and synonym) are same as relationship identification module. After merging all the available security ontologies, we got one coherent ontology (thereby called as SecureOnt ontology). This ontology contains all the information that is available in existing security ontologies. In the SecureOnt ontology, the number of classes are 565, data properties are 122, object properties are 44, and instances are 89. We have maintained the different versions of the ontologies via annotation properties that help to track the data provenance. This information determines where provenance (where is the piece of data stored in for of tuples), why provenance (which part of the data tuple are provided to the result), how provenance (how provided data is helping forward to achieve for a result). Along with data provenance, the developed ontology contains all the aspects of data security: confidentiality, integrity, data availability, and access control mechanisms. For the evaluation of the SecureOnt Ontology, we use pellet reasoner for consistency checking, and OOPs tool for anomalies (pitfall) detection. Figure 7 shows that there is no critical pitfall in the SecureOnt ontology and all the minor and important pitfalls are removed by extending the ontology.

6 Conclusion

This paper presented a SecureOnt Ontology that contains a comprehensive knowledge base of security data (confidentiality, integrity, data availability, and access control mechanisms) by accommodating the existing security ontologies. The developed ontology tracks data provenance in the semantic

web via annotation properties and graph query languages. The evaluation of accessible and downloadable security ontologies is shown via two tools, namely OntMetrics and OOPS!. The evaluation of these ontologies determines the richness and accuracy. We have also calculated the pitfall rate for the investigation of anomalies. The future work of this paper will focus on ontology enrichment.

References

[1] Buneman, P., Khanna, S., and Wang-Chiew, T. (2001, January). Why and where: A characterization of data provenance. In *International conference on database theory* (pp. 316–330). Springer, Berlin, Heidelberg.

[2] Souag, A., Salinesi, C., and Comyn-Wattiau, I. (2012, June). Ontologies for security requirements: A literature survey and classification. In *International conference on advanced information systems engineering* (pp. 61–69). Springer, Berlin, Heidelberg.

[3] Patel, A., and Debnath, N. C. (2022). Development of the InBan_CIDO Ontology by Reusing the Concepts Along with Detecting Overlapping Information. In *Inventive Computation and Information Technologies* (pp. 349–359). Springer, Singapore.

[4] Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Medina, E., Toval, A., and Piattini, M. (2008, March). A systematic review and comparison of security ontologies. In *2008 Third International Conference on Availability, Reliability and Security* (pp. 813–820). Ieee.

[5] Dalpra, H. L., Costa, G. C. B., Sirqueira, T. F. M., Braga, R. M., Campos, F., Werner, C. M. L., and David, J. M. N. (2015). Using Ontology and Data Provenance to Improve Software Processes. In *ONTOBRAS*.

[6] Alneyadi, S., Sithirasenan, E., and Muthukkumarasamy, V. (2016). A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, *62*, 137–152.

[7] Thuraisingham, B., Cadenhead, T., Kantarcioglu, M., and Khadilkar, V. (2014). *Secure data provenance and inference control with semantic web*. CRC Press.

[8] Patel, A., Debnath, N. C., Mishra, A. K., and Jain, S. (2021). Covid19-IBO: a Covid-19 impact on Indian banking ontology along with an efficient schema matching approach. *New Generation Computing*, *39*(3), 647–676.

[9] Fileto, R., Medeiros, C. B., Liu, L., Pu, C., and Assad, E. D. (2003). Using domain ontologies to help track data provenance. In *Embrapa*

Informática Agropecuária-Artigo em anais de congresso (ALICE). In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 18., 2003, Manaus. Anais... Manaus: Universidade Federal do Amazonas, 2003. pp. 84–98.

[10] Kasten, A. (2016). *Secure semantic web data management: confidentiality, integrity, and compliant availability in open and distributed networks* (Doctoral dissertation, Universität Koblenz-Landau).

[11] Xu, G., and Wang, Z. (2010, June). Data provenance architecture based on semantic web services. In *2010 Fifth IEEE International Symposium on Service Oriented System Engineering* (pp. 91–94). IEEE.

[12] Herzog, A., Shahmehri, N., and Duma, C. (2007). An ontology of information security. *International Journal of Information Security and Privacy (IJISP)*, *1*(4), 1–23.

[13] Vorobiev, A., and Han, J. H. J. (2006, November). Security attack ontology for web services. In *2006 Semantics, Knowledge and Grid, Second International Conference on* (pp. 42–42). IEEE.

[14] Vorobiev, A., and Bekmamedova, N. (2010). An ontology-driven approach applied to information security. *Journal of Research and Practice in Information Technology*, *42*(1), 61–76.

[15] Ekelhart, A., Fenz, S., Klemen, M. D., and Weippl, E. R. (2006, December). Security ontology: Simulating threats to corporate assets. In *International Conference on Information Systems Security* (pp. 249–259). Springer, Berlin, Heidelberg.

[16] OWL-S Ontology: http://www.daml.org/services/owl-s/security.html

[17] Bhaumik, A. (2012). *An Approach in Defining Information Assurance Patterns Based on Security Ontology and Meta-modeling*. University of Nebraska at Omaha.

Biographies

Archana Patel is working as a faculty of the Department of Software Engineering, School of Computing and Information Technology, Eastern International University, Binh Duong Province, Vietnam. She has completed her Postdoc from the Freie Universität Berlin, Berlin, Germany. She has filed a patent titled "Method and System for Creating Ontology of Knowledge Units in A Computing Environment" in Nov 2019. She has received Doctor of Philosophy (Ph.D.) in Computer Applications and PG degree both from the National Institute of Technology (NIT) Kurukshetra, India in 2020 and 2016 respectively. She has qualified GATE and UGC-NET/JRF exams in year 2017. Dr. Patel has also contributed in research project funded by Defence Research and Development Organization (DRDO), for the period of two year and her contribution in teaching is also remarkable. Dr. Patel is an author or co-author of more than 30 publications in numerous referred journals and conference proceedings. She has been awarded best paper award (four times) in the international conferences. She has served as a reviewer in various reputed journal and conferences. Dr. Patel has received various awards for presentation of research work at various international conferences, teaching and research institutions. She has edited five books and served as a guest editor in three journals namely Recent Advances in Computer Science and Communications, Recent Patents on Engineering, and Current Material Science. Dr. Patel served as a keynote at ICOECA-2022. Her research interests are Ontological Engineering, Semantic Web, Big Data, Expert System and Knowledge Warehouse.

Narayan C. Debnath is the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years (1989–2017). He was elected as the Chairperson of the Computer Science Department at Winona State University for 3 consecutive terms and assumed the role of the Chairperson of the Computer Science Department at Winona State University for 7 years (2010–2017).

Dr. Debnath earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. In the past, he served as the elected President for 2 separate terms, Vice President, and Conference Coordinator of the International Society for Computers and their Applications, and has been a member of the ISCA Board of Directors since 2001. Before being elected as the Chairperson of the Department of Computer Science in 2010 at Winona State University, he served as the Acting Chairman of the Department. Dr. Debnath received numerous Honors and Awards while serving as a Professor of Computer Science during the period 1989 to 2017. During 1986–1989, Dr. Debnath served as an Assistant Professor of the Department of Mathematics and Computer Systems at the University of Wisconsin-River Falls, USA, where he was nominated for the prestigious US National Science Foundation (NSF) Presidential Young Investigator Award in 1989.

Dr. Debnath is an author or co-author of over 450 publications in numerous refereed journals and conference proceedings in Computer Science,

Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering. Dr. Debnath is an editor of several books published by Springer, Elsevier, CRC Press, and Bentham Science Press on emerging fields of computing. For last many years, Dr. Debnath has been organizing highly successful Special Sessions attracting a large number of quality submissions at both IEEE INDIN and IEEE ICIT. He also served as a guest editor of the Journal of Computational Methods in Science and Engineering (JCMSE) published by the IOS Press, the Netherlands. Dr. Debnath has been an active member of the ACM, IEEE Computer Society, Arab Computer Society, and a senior member of the ISCA.

Prashant Kumar Shukla received his Master of Engineering in Software Systems from RGPV, Bhopal, Madhya Pradesh in 2010 and Ph.D. (Computer Science & Engineering) in 2018 from Dr. K. N. Modi University, Rajasthan, India. Presently, he has been working as an Associate Professor (Research) in the Department of computer science and engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh from October 2021. He has also worked as an Assistant Professor (SG) and Research Coordinator in Jagran Lakecity University, Bhopal (MP) India from July 2019 to September 2021. He has also worked as Associate Professor in SISTech, Bhopal (MP) from Jan 2019 to July 2019 and as Associate Professor in SIRT, Bhopal (MP) India from 2011 to 2019.

He has been in teaching, research, innovation and industry for the past 20 years and working in the research areas like Artificial Intelligence, Machine Learning, Deep Learning, Data Science, Computer Vision, Internet of Things (IOT), Neural Networks, Software Engineering, Computer Networking and

Network Security concerns. He has been granted 04 patents. He has been published 26 Indian patents and one Indian Copyright is granted. He has received funding for 2 research projects from Govt. He has published 16 research papers in SCI (Science Citation Index), 7 papers in Scopus indexed and 11 papers in peer reviewed journals and conferences. He has also published 02 Chapters in Scopus indexed edited book.