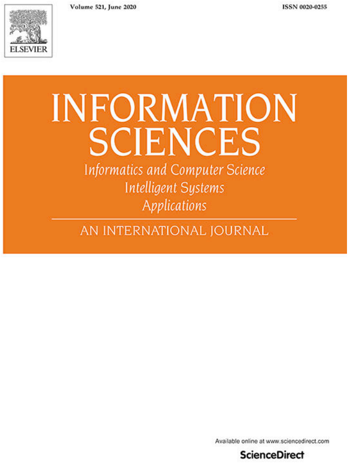

Volume 521, June 2020

ISSN 0020-0255

ELSEVIER

INFORMATION SCIENCES

Informatics and Computer Science
Intelligent Systems
Applications

AN INTERNATIONAL JOURNAL

Available online at www.sciencedirect.com

ScienceDirect

# Semantic Attribute-Based Encryption: A Framework for Combining ABE schemes with Semantic Technologies

Hamed Arshad, Christian Johansen, Olaf Owe, Pablo Picazo-Sanchez, Gerardo Schneider

PII: S0020-0255(22)01261-0
DOI: https://doi.org/10.1016/j.ins.2022.10.132
Reference: INS 18240

To appear in: *Information Sciences*

Received Date: 1 May 2022
Accepted Date: 28 October 2022

Please cite this article as: H. Arshad, C. Johansen, O. Owe, P. Picazo-Sanchez, G. Schneider, Semantic Attribute-Based Encryption: A Framework for Combining ABE schemes with Semantic Technologies, *Information Sciences* (2022), doi: https://doi.org/10.1016/j.ins.2022.10.132

# Semantic Attribute-Based Encryption: A Framework for Combining ABE schemes with Semantic Technologies

Hamed Arshad[a,*], Christian Johansen[b], Olaf Owe[a], Pablo Picazo-Sanchez[c], Gerardo Schneider[d]

[a] *Department of Informatics, University of Oslo, Norway*
[b] *Norwegian University of Science and Technology, Norway*
[c] *Chalmers University of Technology, Sweden*
[d] *Department of Computer Science and Engineering, University of Gothenburg, Sweden*

**Abstract**

Attribute-Based Encryption (ABE) is a cryptographic solution to protect resources in a fine-grained manner based on a set of public attributes. This is similar to attribute-based access control schemes in the sense that both rely on public attributes and access control policies to grant access to resources. However, ABE schemes do not consider the semantics of attributes provided by users or required by access structures. Such semantics not only improve the functionality by making proper access decisions but also enable cross-domain interoperability by making users from one domain able to access and use resources of other domains. This paper proposes a Semantic ABE (SABE) framework by augmenting a classical Ciphertext-Policy ABE (CP-ABE) scheme with semantic technologies using a generic procedure by which any CP-ABE scheme can be extended to an SABE. The proposed SABE framework is implemented in Java and the source code is publicly available. The experiment results confirm that the performance of the proposed framework is promising.

*Key words:* Attribute-Based Encryption, Semantic technologies, Security, Interoperability, Privacy, Access Control, Ontology

## 1. Introduction

Access Control (AC) is a fundamental security mechanism to restrict access to (sensitive) data. One of the most promising AC models is Attribute-Based Access Control (ABAC) [27, 39], which provides fine-grained protection based

[*] Corresponding author
*Email addresses:* hamedar@ifi.uio.no (Hamed Arshad), christian.johansen@ntnu.no (Christian Johansen), olaf@ifi.uio.no (Olaf Owe), Pablo.Picazo-Sanchez@cse.gu.se (Pablo Picazo-Sanchez), gerardo@cse.gu.se (Gerardo Schneider)

on a set of attributes and access control policies. However, ABAC, like every access control mechanism, relies on a trusted reference monitor that checks all access requests against access control policies and can be easily bypassed, e.g., by getting direct access to the data on a storage device. In contrast to ABAC, Attribute-Based Encryption (ABE) [11, 22] does not rely on a trusted engine (monitor), but uses cryptographic techniques to provide fine-grained data protection based on Access Structures (ASs) (i.e., access control policies) represented as a boolean formula over public attributes. Any user who holds a set of public attributes satisfying the AS can decrypt the ciphertext. For instance, if a picture is encrypted under the following AS: $((\mathit{Friend\_of\_Alice} \wedge \mathit{Age} > \mathit{30}) \vee (\mathit{Support} = \mathit{TeamA}))$, then only friends of Alice who are over 30 years old and those who support *TeamA* can decrypt the picture, where the *Friend_of_Alice* attribute is granted to users that Alice marked as friends. ABE makes it possible to encrypt data not only for a single user (identified by a unique attribute) but also for a group of users (identified by a set of public attributes).

Until now, a considerable number of ABE schemes have been proposed and employed in several domains such as eHealth [35], online social networks [40], hardware security [21], fog computing [28], and storing sensitive data in public clouds [34]. Furthermore, real world companies like Zeutro[1] deploy security systems based on ABE. Moreover, standards like ETSI[2] have been defined (TS 103 458 and TS 103 532) presenting applications to industrial IoT and cloud.

However, the existing ABE schemes are not semantic-aware, i.e., they do not take into account the semantics of attributes. Semantic awareness could improve the functionality of ABE schemes and allow for cross-domain interoperability of systems based on ABE.

Consider the application of ABE in online social networks [6, 40] where platforms (e.g., Twitter, Facebook, and LinkedIn) use different terminologies (for attributes). For instance, Facebook users can share information (e.g., events, photos, videos) with different groups of audiences such as the *Public*, *Friends*, and *Specific Friends*, whereas in Twitter the target audiences can be specified as *Everyone*, *People you follow*, and *Only people you mention*. Furthermore, in LinkedIn the visibility options for posts are *Anyone*, *Connections only*, *Group members*, and *Event attendees*. It is obvious that *Public*, *Everyone*, and *Anyone* have the same meaning, despite being syntactically different. Similarly, *Friends*, *Followers*, and *Connections* are semantic synonyms. Semantic technologies [2, 4, 5, 8, 25] are particularly useful for handling semantic translations, as commonly required for interoperability between different domains.

Interoperability problems are notoriously common also in eHealth where medical staff from different healthcare institutions need to access data like Electronic Health Records (EHR). When coupled with a growing trend of moving medical records into public clouds [7, 50][3], ABE gains even more relevance. The

[1] `https://bit.ly/3gvWRGE`
[2] `https://bit.ly/3xiLoQk`
[3] `https://ibm.co/2Tz2LxL`

power of semantic technologies goes beyond semantic synonymity and translations by allowing for more types of inferences.

Suppose the EHR of *PatientA* is encrypted based on the following AS: ($GP \vee (\textit{Medical Doctor} \wedge \textit{Employer} = \textit{Emergency Hospital})$). A surgeon working at the Emergency Hospital with attributes $\{\textit{Surgeon}, \textit{Employer} = \textit{Emergency Hospital}\}$ will not be able to access the EHR of *PatientA* because she does not hold the *Medical Doctor* attribute. Even though a surgeon is a medical doctor, ABE works syntactically and cannot infer such knowledge. Any basic medical ontology would have *Surgeon* as a subconcept of *Medical Doctor* and would allow an inference engine to infer this information, which can then be added as the extra attribute needed in such emergency cases. It is worth mentioning that the power of semantic technologies is not limited to such simple translations. Semantic technologies allow having more complex inference rules in addition to the basic ones that are based on inheritance. For instance, $HospitalA$ may have an attribute *Senior Surgeon* for surgeons who have worked more than 5 years as a surgeon and hold $X$ and $Y$ certificates. However, $HospitalB$ may not use such an attribute and only use the *Surgeon* attribute. Hence, a surgeon at $HospitalB$ who has worked more than five years and holds both $X$ and $Y$ certificates would not be able to access (decrypt) a file that is protected (encrypted) based on the *Senior Surgeon* attribute at $HospitalA$ as she does not hold such an attribute.

The aim of this paper is to combine ABE schemes with semantic technologies in order to address the two types of semantic enhancements exemplified above. In particular, we achieve *semantic-aware ABE schemes* by making ABE schemes able to use implicit knowledge from an ontology, while facilitating the interoperability between ABE schemes used in different domains. In more detail:

- In Section 3, we present SABE, our framework, which can be built around an arbitrary ABE scheme and an arbitrary inference engine working against an arbitrary ontology.
- In Section 4, we analyze the security of the proposed framework.
- We provide a prototype implementation, detailed in Section 5, where we use a specific Ciphertext-Policy Attribute Based Encryption (CP-ABE) scheme [11], and a specific inference engine called Pellet [43] that works with a quite popular semantic language OWL, over a mock ontology that we have made for this implementation; but the ontology, like the other two aspects, can be freely replaced.
- We evaluate, in Section 6, further properties of SABE in terms of modularity, scalability, extensibility, and generality.

In Section 2, we introduce some general terms and definitions used in this paper. Section 7 presents the related work while the paper concludes in Section 8.

## 2. Preliminaries

This section gives some background information on attribute based encryption and semantic technologies.

### *2.1. Attribute-Based Encryption*

In conventional public-key cryptography, data are encrypted for a particular receiver using the receiver's public key. Hence, if the same data should be encrypted for several receivers, all the public keys of the receivers are needed. This is even more problematic for data that need to be stored encrypted for sharing with future, yet unknown, users. In response to this, Goyal et al. [22] proposed the first ABE scheme by which the encryptor can encrypt a message under a set of public attributes (instead of just an identity as in identity-based encryption schemes [13]). Therefore, data can be encrypted for a group of recipients holding the same public attributes.

ABE is a public-key cryptography in which private keys of users and ciphertexts depend upon attributes. The private keys of users are associated to sets of public attributes, whereas each ciphertext has attached an access structure (in CP-ABE). Anyone who has a set of attributes that satisfies the AS of the ciphertext can decrypt it.

When a user joins the system, she claims to have a set of public attributes and a Trusted Authority (TA) is in charge to validate them. If deemed appropriate, the TA provides the user with a private key associated to her attributes. This authentication process is usually out of the scope of ABE schemes since it is assumed that the TA has the knowledge—or the corresponding mechanisms—to prove that users really have the attributes they claim to have. Consequently, in this paper we assume that users cannot cheat the TA and they are provided with the public attributes they actually have.

The advantages of using ABE are multiple: i) different groups of users can be defined according to public attributes; ii) all the encrypted data can be publicly stored in databases because only users that satisfy the AS will retrieve the plaintext, and; iii) security properties such as access control, user collusions and data disclosures are guaranteed by the underlying cryptographic infrastructure.

The main algorithms of a CP-ABE [11] are `Setup`, `KeyGen`, `Encryption`, and `Decryption`. While the first two algorithms are run by the TA, the last two are executed by the users.

**Setup($1^\lambda$)** This algorithm takes a security parameter as input and generates a master secret key ($MK$) and a set of public parameters ($PP$).

**KeyGen($MK$, $S$, $PP$)** This algorithm produces a private key ($SK$) for a provided set of attributes, $S = \{Att_1, ..., Att_N\}$, using the master secret key and public parameters.

**Encryption($M$, $\mathcal{T}$, $PP$)** It encrypts a message $M$ based on the access structure ($\mathcal{T}$) and public parameters, and returns a ciphertext $CT = (\mathcal{T}, C)$, where $C$ is the encrypted version of $M$.

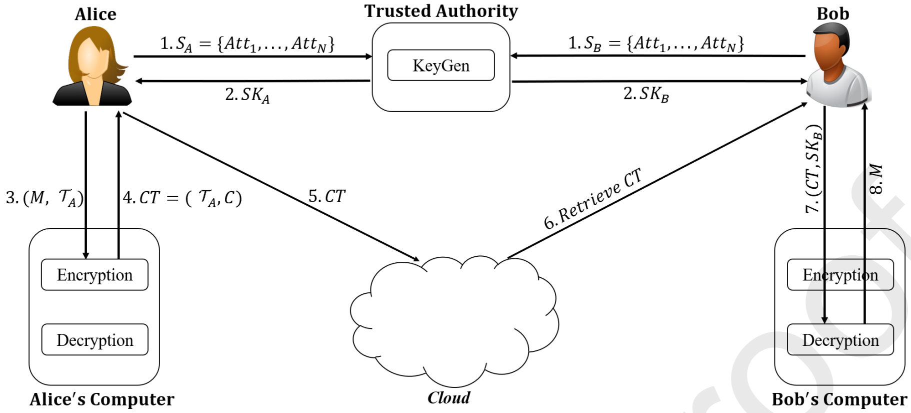

Figure 1: General architecture of a CP-ABE scheme.

**Decryption($CT$, $SK$, $PP$)** It decrypts a ciphertext $CT$ using a private key ($SK$), which is generated for a set of attributes satisfying the access structure included in $CT$, and public parameters.

We include a graphical presentation in Figure 1 where two users, Alice and Bob, join the system. First, they provide their public attributes ($S_A$ and $S_B$) to the TA (communication 1) and they receive the private keys ($SK_A$ and $SK_B$) associated to their attributes (2). After that, Alice runs the `Encryption` algorithm producing $CT$ (3 and 4) and sends the ciphertext to the cloud where Bob can get it (5). When Bob retrieves $CT$ from the cloud (6), he runs the `Decryption` algorithm (7) and finally, he obtains the plaintext (8). Figure 1 depicts all the steps in a CP-ABE scheme; however, it does not mean that all the steps are required for all kinds of operations. For instance, if Alice wants to encrypt a message, she only needs to run the `Encryption` algorithm and provide the message, the desired access structure, and the public parameters (i.e., for encryption, the data owner does not need to get a private key for her attributes).

### *2.2. Semantic technologies*

Semantic technologies are a collection of methods, languages, and tools that facilitate advanced data categorization, processing, and relationship discovery across a variety of data sets. Concepts (entities or data) and relationships between them in a certain domain can be described and represented by means of vocabularies. Vocabularies are useful not only for organizing knowledge, but also for resolving ambiguities when integrating different data sets.

RDF Schema (RDFS) [15] was proposed as a language for defining Resource Description Framework (RDF) vocabularies. RDFS is based on the notion of classes and inheritance relationships like those in object-oriented programming languages. RDFS makes it possible to define taxonomies (very simple vocabularies) and perform simple inferences about them. More complex vocabularies, which have thousands of concepts, are called ontologies that can be represented by classes, relations, and instances. The classes can be related to each other by means of relations. For example, in an ontology, class “Medical Doctor” maybe a *subclass* of a class “Medical Staff” or a relation *“is GP of”* may exist between the class “Medical Doctor” and a class “Patient”. There can also be some constraints between the relations among classes that determine which kind of values are allowed. For example, each “Patient” is always a “Person” (a one-to-one relationship) and a “Patient” cannot be a subclass of two “Persons”. The classes, relations, and constraints can be combined to form complex statements or assertions expressing the knowledge. In other words, they form the Terminological Knowledge (TBox).

An ontology can be populated by means of instances. In other words, all the classes can contain individuals with some relationships between them. For example, “Alice” can be an instance of the “Medical Doctor” class in the ontology that can have an “ID” and also a relation “is GP of” with another individual “Bob”, which is an instance of the class “Patient”. The knowledge about the individuals is called Assertional Knowledge (ABox). The terminological knowledge and assertional knowledge form a Knowledge Base, which represents an ontology. The Web Ontology Language (OWL) is a standard language for creating/defining ontologies and provides richer semantics than the RDFS.

Inheritance relationships in RDFS and OWL are simple relationships which can be used for performing very simple inferences. However, if some specific relationships need to be held under some conditions, then it is difficult to express them using the ontology markup languages, e.g., RDFS and OWL. For example, it is difficult to specify that a “Chief Physician” is a “Medical Doctor” who was hired more than 10 years ago. Such relationships can be handled using rules.

Semantic Web Rule Language (SWRL) [25] is a popular and standard rule markup language that combines Horn logic rules and OWL ontologies to define complex relationships between concepts. Using SWRL it is possible to specify complex inference rules in addition to the basic ones that are based on inheritance. Inference rules make it also possible to infer new knowledge (i.e., inferring implicit relationships from explicit ones). In order to do the inference, a reasoner (i.e., inference engine) is required. A reasoner uses a set of facts and axioms to get new logical consequences.

## 3. SABE: A Semantic ABE Framework

Augmenting ABE schemes with semantic technologies results in semantic-aware schemes by including implicit knowledge in controlling access and improves cross-domain interoperability. We developed a semantic component consisting of 1) a domain ontology; 2) SWRL rules, and; 3) an inference engine. The domain ontology represents the semantic relationships between attributes in a given domain (or domains). The SWRL rules are used to define more complex relationships that are not possible to be defined using ontology data modeling languages. The inference engine performs the reasoning and infers the implicit knowledge.

In the following, we propose two approaches, namely *Semantically-Enriched Key* and *Semantically-Enriched Access Structure*, to augment ABE schemes with semantic technologies. First, we define what an ontology is (see Definition 1) as well as the relationships we use between concepts in an ontology (see Definition 2). Definition 3 and Definition 4 define “semantically relevant attributes” (or “semantically relevant concepts” as attributes are represented as concepts in the ontology) in the proposed *Semantically-Enriched Key* and *Semantically-Enriched Access Structure* approaches, respectively.

**Definition 1** (Ontology)**.** *An ontology is a tuple $O = \langle \boldsymbol{C}, \boldsymbol{R}, \boldsymbol{I} \rangle$, where $\boldsymbol{C}$, $\boldsymbol{R}$, and $\boldsymbol{I}$, denote, respectively, sets of concepts (classes), relationships between concepts, and instances (individuals) belonging to concepts. Concepts represent the attributes in a domain.*

**Definition 2** (Relationships)**.** *Our proposals use the following relationships between concepts in an ontology:*

- `subClassOf` ($\subseteq$)*: if the semantic scope of a concept $C_1 \in \boldsymbol{C}$ is narrower than that of another concept $C_2 \in \boldsymbol{C}$, i.e., every instance of $C_1$ is also an instance of $C_2$, then $C_1$ is a subclass of $C_2$. In other words, $C_1 \subseteq C_2$ iff $\forall i \in \boldsymbol{I} : (i \in C_1 \rightarrow i \in C_2)$, where $C_1, C_2 \in \boldsymbol{C}$.*

- `equivalentClass` ($\equiv$)*: if the semantic scope of a concept $C_1 \in \boldsymbol{C}$ is equal to that of another concept $C_2 \in \boldsymbol{C}$, i.e., both concepts have exactly the same set of individuals, then $C_1$ is an equivalent class of $C_2$. In other words, $C_1 \equiv C_2$ iff $\forall i \in \boldsymbol{I} : ((i \in C_1 \rightarrow i \in C_2) \wedge (i \in C_2 \rightarrow i \in C_1))$, where $C_1, C_2 \in \boldsymbol{C}$.*

**Definition 3** ($SR^{Att}_{SEK}$)**.** *Given a concept $C_{in} \in \boldsymbol{C}$ from an ontology $O$, the semantically relevant concepts in SABE-SEK denoted by $SR^{Att}_{SEK}(C_{in})$ are defined as the set of all concepts $C_j \in \boldsymbol{C}$ such that $C_{in} < C_j$, where $<$ is defined as the smallest relation satisfying the following rules:*

- *$C_2 < C_1$ if $C_2 \subseteq C_1$, i.e., $C_2$* `subClassOf` *$C_1$*
- *$C_2 < C_1$ if $C_2 \equiv C_1$, i.e., $C_2$* `equivalentClass` *$C_1$*
- *$C_2 < C_1$ if $C_2 < C_3$ and $C_3 < C_1$*

**Definition 4** ($SR^{Att}_{SEAS}$)**.** *Given a concept $C_{in} \in \boldsymbol{C}$ from an ontology $O$, the semantically relevant concepts in SABE-SEAS denoted by $SR^{Att}_{SEAS}(C_{in})$ are defined as the set of all concepts $C_j \in \boldsymbol{C}$ such that $C_j < C_{in}$, where $<$ is defined as the smallest relation satisfying the rules provided in Definition 3.*

### *3.1. SEK: Semantically-Enriched Key*

One type of semantic ABE extension, which we call SABE-SEK, is composed of six algorithms: `ABE.Setup`, `ABE.KeyGen`, `ABE.Encryption`, `ABE.Decryption`, `SABE.UpdateAtt`, and `SABE.KeyGen`, of which the first four are identical to those of a conventional CP-ABE scheme as described in Section 2.1. The last two (extra) algorithms, which the TA runs, are defined below:

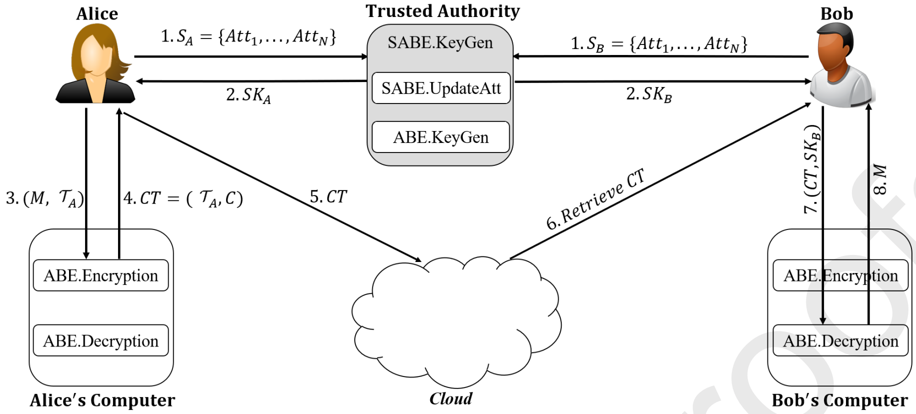

Figure 2: General architecture of the proposed SABE-SEK.

**`SABE.UpdateAtt(`$S$`,` $Ontology$`)`** updates a provided set of attributes, $S$, based on the semantic relationships defined in a provided ontology, $Ontology$, and returns a new set of attributes, $S'$.

**`SABE.KeyGen(`$MK$`,` $S$`,` $PP$`)`** connects the `ABE.KeyGen` algorithm to the semantic component, i.e., the `SABE.UpdateAtt` algorithm. `SABE.KeyGen` first calls `SABE.UpdateAtt` to update the provided set of attributes, and then calls `ABE.KeyGen`($MK$, $S'$, $PP$), which generates a private key, $SK$, for the updated set of attributes using the master secret key and public parameters.

Note that the TA is trusted to hold the *Ontology* secret, which is part of the initialization process. Otherwise, an adversary may manipulate the ontology to its own advantage.

In the proposed SABE-SEK scheme, we use semantic technologies in the key generation process. The key generation (`ABE.KeyGen`) algorithm generates a private key associated to a set of public attributes that a user provides. The idea is to extend the set of user's attributes by adding all semantically relevant attributes as defined in Definition 3. Accordingly, the `SABE.KeyGen` algorithm (by calling the `ABE.KeyGen` algorithm) generates a private key based on the extended set of attributes, which in turn means that the user capabilities (in terms of access) will be enhanced as both explicit and implicit knowledge are used in the generation of the user's private key.

Figure 2 depicts the architecture of the proposed SABE-SEK scheme. Note that SABE-SEK differs from CP-ABE in the key generation process. The other algorithms, i.e., `Setup`, `Encryption`, and `Decryption`, do not need any changes and are identical to the CP-ABE ones, which is why they have an `ABE` prefix.

As demonstrated in Algorithm 1, when a user submits a set of attributes to a TA, the `KeyGen` algorithm (i.e., the `SABE.KeyGen` algorithm) does not generate a private key as in the classical CP-ABE schemes. Instead, it (i.e., the

**Algorithm 1:** Pseudocode of SABE-SEK key generation

```
Input: Master secret key ($MK$), A set of attributes ($S$), Public
       parameters ($PP$)
Output: A private key ($SK$)
// Updating the provided set of attributes ($S$) using a domain ontology
   ($Ontology$). The updated set of attributes is $S'$.
1 Call SABE.UpdateAtt,
    $S' \leftarrow$ SABE.UpdateAtt($S$, $Ontology$)
// Generating a private key using the KeyGen algorithm of a CP-ABE scheme.
2 Call ABE.KeyGen,
    $SK \leftarrow$ ABE.KeyGen($MK$, $S'$, $PP$)
3 Return $SK$
```

`SABE.KeyGen` algorithm) first uses our semantic component, which includes an inference engine and a domain ontology (along with SWRL rules), to obtain all other attributes that are semantically relevant (according to Definition 3) to those submitted by the user. `SABE.KeyGen` algorithm does this by calling the `SABE.UpdateAtt` algorithm. For example, every *Surgeon* is a *Medical Doctor*; however, every *Medical Doctor* is not necessarily a *Surgeon*. Hence, if a user submits the *Surgeon* attribute, then the semantic component infers that this user implicitly holds the *Medical Doctor* attribute as well based on the domain ontology. After retrieving all the attributes that are semantically relevant to the submitted attributes (i.e., extending the set of attributes in most cases), the `SABE.KeyGen` algorithm calls the `ABE.KeyGen` algorithm, which is the `KeyGen` algorithm of a conventional CP-ABE scheme, to generate a private key for the extended set of attributes (i.e., the attributes that the user submitted and those added by the semantic component) as in the classical CP-ABE schemes. Therefore, a user who has the *Surgeon* attribute would be able to decrypt any ciphertext encrypted under the *Surgeon* attribute or the *Medical Doctor* attribute.

To infer and find the semantically relevant attributes based on Definition 3, an ontology describing the relationships between the attributes in the given domain is required. For example, we created a proof-of-concept ontology for the healthcare domain (for SABE-SEK), where the concepts represent the attributes. This ontology is provided to the TA during system initialization. When the TA receives a request (a set of attributes) for generating a private key, it calls the semantic component providing the received set of attributes and the ontology. The semantic component finds the semantically relevant attributes according to Definition 3 and Algorithm 2. It first assigns a dummy OWL Named Individual to the concepts in the ontology that are related to the received attributes. For example, if a user submits a *Surgeon* attribute, then a dummy Individual, e.g., "TestUser", will be assigned to the concept "Surgeon" in the ontology. In other words, some assertions will be generated (based on the submitted attributes) and inserted into the ABox of the knowledge base. Then, a reasoner (an inference engine) will be executed to infer all

**Algorithm 2:** Finding semantically relevant attributes in SABE-SEK

```
Input: $S_x$, the received set of attributes
Output: $S_x'$, updated set of attributes
Load the ontology
Create a dummy OWLNamedIndividual, $TestUser$
for every attribute $Att_i$ in $S_x$ do
    if $Att_i$ is a property, e.g., $hasTraveled = 3$ then
        Create a property assertion axiom for $TestUser$ as ($Att_i$,
          $TestUser$, $value$). For example, $hasTraveled$ and 3 will be used
          as $Att_i$ and $value$, respectively, for $hasTraveled = 3$.
        Add the created axiom to the ontology
    else
        Create a class assertion axiom for $TestUser$ as ($Att_i$, $TestUser$)
        Add the created axiom to the ontology
Synchronize the reasoner (inference engine) to obtain inferred axioms
Add inferred axioms to the ontology
Realize the ontology to run SWRL rules
$S_x' \leftarrow$ Find all the classes (concepts) in the ontology that $TestUser$
  belongs.
Return $S_x'$ as the updated set of attributes
```

the semantically relevant attributes. For example, according to the ontology that we developed for this paper, for the *Surgeon* attribute, the inference engine infers *Medical Doctor*, *Physician*, *Lege*[4], and *Person* attributes as semantically relevant attributes based on Definition 3. However, some more attributes may be derived in the case of having SWRL rules. Therefore, the semantic component extends the set of submitted attributes and returns the extended set to the key generation algorithm, which generates a private key for the extended set of attributes using the `KeyGen` algorithm of a conventional CP-ABE scheme (i.e., using `ABE.KeyGen`). Finally, the inserted assertions (for the attributes submitted by the user) will be deleted from the ABox.

The following example shows the difference between a CP-ABE scheme and the proposed SABE-SEK scheme.

**Example 1.** *Suppose that Alice's EHRs are encrypted based on the following AS:* $(Bob \vee (Medical\ Doctor \wedge Employer = Emergency\ Hospital))$*, where Bob is Alice's GP. If Charlie, who is a surgeon working at the Emergency Hospital with attributes* $\{Surgeon,\ Employer = Emergency\ Hospital\}$ *wants to access Alice's EHRs, he will not be able to decrypt them using a CP-ABE scheme. This is because the CP-ABE works syntactically and cannot infer that a surgeon is a kind of medical doctor. In the proposed SABE-SEK scheme, however,*

[4] Lege in Norwegian means Medical Doctor in English

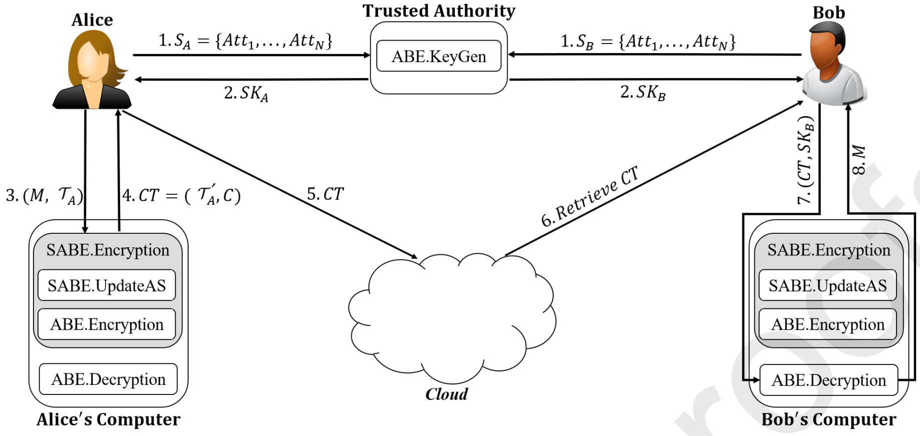

Figure 3: General architecture of the proposed SABE-SEAS

*the semantic component infers that Charlie (holding the Surgeon attribute) implicitly has Medical Doctor, Physician, Lege, and Person attributes. Hence, Charlie's private key will be generated according to the original and inferred attributes, i.e., {Surgeon, Medical Doctor, Physician, Lege, Person, Employer = Emergency Hospital}. Therefore, Charlie can decrypt and access Alice's EHRs using the proposed SABE-SEK scheme.*

To enable cross-domain interoperability, we need a common ontology for collaborating domains. For example, we can use a common ontology describing the attributes and relationships between attributes in three online social networks, e.g., Facebook, Twitter, and LinkedIn. When a user submits a *Friend* attribute, the semantic component (in SABE-SEK) infers and returns *Follower* and *Connection* attributes, which have the same meaning on Twitter and LinkedIn, respectively. Then, the user's private key will be generated based on *Friend*, *Follower*, and *Connection* attributes. This approach works properly if we have only one TA for all domains. However, in reality, Facebook, Twitter, and LinkedIn have their own TAs for generating private keys. Accordingly, the `KeyGen` algorithm of each domain uses a different master secret key, and thus, for example, Facebook and Twitter generate different private keys for the same attributes. Hence, it can be said that the proposed SABE-SEK scheme, only makes the ABE schemes semantic-aware and does not enable cross-domain interoperability. To provide cross-domain interoperability, we present a second approach called *Semantically-Enriched Access Structure (SEAS)*, which is described in the next subsection.

### *3.2. SEAS: Semantically-Enriched Access Structure*

Our second proposal (see Figure 3) is called SABE-SEAS, and is composed of seven algorithms: `ABE.Setup`, `ABE.KeyGen`, `ABE.Encryption`, `ABE.Decryption`,

**Algorithm 3:** Finding semantically relevant attributes in SABE-SEAS

```
Input: An attribute $Att_i$
Output: $L_i$: Attributes that are semantically relevant to $Att_i$
Load the ontology
Pre-compute classification (classifying the ontology)
Pre-compute instances for each named class in the ontology
$L_i \leftarrow$ Find all subclasses of the class $Att_i$ in the ontology
$L_i \leftarrow$ Find all named classes that are equivalent to the class $Att_i$ with
  respect to the set of reasoner axioms
Remove the repeated elements from $L_i$
Return $L_i$ as semantically relevant attributes to $Att_i$
```

`SABE.Setup`, `SABE.UpdateAS`, and `SABE.Encryption`; the first four being identical to those of a conventional CP-ABE scheme, while the last three are described below.

The idea of the SABE-SEAS scheme is to enable cross-domain interoperability by enriching the access structures utilizing semantic technologies. In SABE-SEAS, we add the semantic component to the `Encryption` algorithm of a CP-ABE scheme. When a user wants to encrypt data based on an access structure, the proposed SABE-SEAS scheme updates the provided access structure by including semantically relevant attributes (as defined in Definition 4, and based on Algorithm 3 and Algorithm 4) into the access structure and then encrypts the data based on the updated access structure.

As already mentioned, a common ontology describing the attributes and relationships between attributes in different domains is required to provide cross-domain interoperability. We employ a secure signature scheme [12] to guarantee the validity and authenticity of the ontology. Every TA signs the ontology for its own users using its master secret key and provides the ontology and the corresponding signature to the users as public parameters. The ontology will be validated before updating an access structure by verifying the signature. The goal is to detect unauthorized modifications in the ontology, i.e., the integrity of the ontology, and not the confidentiality of it. Different TAs in SABE-SEAS will generate private keys as usual. A TA, which works independently as in a conventional CP-ABE scheme, does not share any private information or computation with other TAs. In Example 2, Facebook, Twitter, and LinkedIn do not share any private information with each other (except the common ontology that they agreed upon) and they do not need to trust each other (i.e., external key generators). However, the public keys (or public attributes) of different domains should be publicly available.

`SABE.Setup(`$1^\lambda$`, `*Ontology*`)` is run by the TA and takes as input a security parameter and an ontology (which can be a common ontology for several domains). It calls `ABE.Setup` to generate a master secret key ($MK$) and a set of public parameters ($PP$), after which it signs the *Ontology* using

**Algorithm 4:** Finding attributes that are semantically relevant to an AND-statement in SABE-SEAS

```
Input: $L_{Input}$: Attributes existing in an AND-statement
Output: $L_{AND}$: Attributes that are semantically relevant to the
        AND-statement
Load the ontology
Create a dummy OWLNamedIndividual, $TestUser$
for every attribute $Att_i$ in $L_{Input}$ do
    if $Att_i$ is a property, e.g., $property = value$ then
        Create a property assertion axiom for $TestUser$ as ($Att_i$,
          $TestUser$, $value$), where $Att_i$ is a property.
        Add the created axiom to the ontology
    else
        Create a class assertion axiom for $TestUser$ as ($Att_i$, $TestUser$)
        Add the created axiom to the ontology
Synchronize the reasoner (inference engine) to obtain inferred axioms
Add inferred axioms to the ontology
Realize the ontology to run SWRL rules
$L_{AND} \leftarrow$ Find all the classes (concepts) in the ontology that $TestUser$
  belongs.
Remove superclasses and equivalent classes of the attributes in $L_{Input}$
  from $L_{AND}$
Return $L_{AND}$ as semantically relevant attributes to the AND-statement
```

a secure signature scheme [12] with the master secret key $MK$ and adds the *Ontology* and the produced signature $\sigma$ to the public parameters.

`SABE.Encryption(`$M$`, `$\mathcal{T}$`, `*Ontology*`, `$\sigma$`, `$PP$`)` is run by users, connecting the `ABE.Encryption` algorithm to the semantic component, taking as input a message, $M$, an access structure, $\mathcal{T}$, a common ontology, *Ontology*, and the ontology's signature generated by the TA. The ontology and the corresponding signature are provided to users as public parameters. As demonstrated in Algorithm 5, the `SABE.Encryption` algorithm first calls `SABE.UpdateAS` to update the provided access structure based on the semantic relationships defined in the ontology, which is then provided to `ABE.Encryption`($M$, $\mathcal{T}'$). Finally, it returns a ciphertext $CT = (\mathcal{T}', C)$, where $C$ is the encrypted version of $M$.

`SABE.UpdateAS(`$\mathcal{T}$`, `*Ontology*`, `$\sigma$`)` is run by users, taking as input an access structure, and the signed ontology. `SABE.UpdateAS` first checks the signature and then returns an updated access structure, $\mathcal{T}'$, using semantic relationships inferred using the ontology.

For example, suppose that in SABE-SEAS, a user wants to encrypt data based on the following access structure: $(Att_a \vee Att_b) \wedge (Att_c \vee Att_d)$.

**Algorithm 5:** Pseudocode of SABE-SEAS encryption

```
Input: Plaintext($M$), Access structure ($\mathcal{T}$), $Ontology$, Signature of the
       Ontology ($\sigma$), Public parameters ($PP$)
Output: Ciphertext ($CT$)
// Updating the access structure ($\mathcal{T}$) using a domain ontology (after
   validating the authenticity of $Ontology$ using its signature, $\sigma$). $\mathcal{T}'$
   denotes the updated access structure.
Call SABE.UpdateAS,
  $\mathcal{T}' \leftarrow$ SABE.UpdateAS($\mathcal{T}$, $Ontology$, $\sigma$)
// Encrypting a plaintext based on the updated access structure using the
   Encryption algorithm of a CP-ABE scheme.
Call ABE encryption,
  $CT \leftarrow$ ABE.Encryption($M$, $\mathcal{T}'$)
Return $CT$
```

Assume that in the common ontology $Att_{a1}$ and $Att_{a2}$ attributes are semantically relevant to $Att_a$ attribute and $Att_{b1}$, $Att_{c1}$, and $Att_{d1}$ are semantically relevant to $Att_b$, $Att_c$, and $Att_d$ attributes, respectively (according to Definition 4). Furthermore, there is a SWRL rule stating that $(Att_a \wedge Att_d)$ is the same as $Att_e$.

The provided access structure will be updated as follows: $((Att_a \vee Att_{a1} \vee Att_{a2} \vee Att_b \vee Att_{b1}) \wedge (Att_c \vee Att_{c1} \vee Att_d \vee Att_{d1})) \vee Att_e$.

Then, the data will be encrypted based on the updated access structure. Note that $Att_{a1}$, $Att_{a2}$, $Att_{b1}$, $Att_{c1}$, and $Att_{d1}$ could be the attributes of different domains, which are publicly available. It should be noted that for encryption, we only need the name of attributes (or their public keys) and not any private key related to the attributes.

The proposed SABE-SEAS scheme updates access structures based on Algorithm 6.

The following example, which is based on Figure 4, demonstrates how the proposed SABE-SEAS scheme enables cross-domain interoperability. Alice is a user who has an account in three social networks, e.g., Facebook, Twitter, and LinkedIn. Bob and Charlie are close friend and friend, respectively, of Alice on Facebook. David and Alice follow each other on Twitter and Emily and Alice are connected on LinkedIn.

**Example 2.** *Suppose Alice wants to share a post on Twitter with her followers. Using a CP-ABE scheme, her post will be encrypted based on the following AS:* $(Follower = Alice)$. *It is obvious that Bob, Charlie, and Emily, who are connected to Alice on Facebook and LinkedIn, cannot decrypt and access Alice's post on Twitter as they do not have the related private key. However, in the proposed SABE-SEAS scheme, the provided AS:* $(Follower = Alice)$ *will be updated as* $((Follower = Alice) \vee (Connection = Alice) \vee (Intimate\ Fried = Alice)) \vee (Friend = Alice) \vee (Close\ Friend = Alice)$ *with the help of the semantic compo-*

**Algorithm 6:** Pseudocode for updating an access structure

```
Input: $\mathcal{T}$, the access structure to be updated
Output: $\mathcal{T}'$, updated access structure
for every attribute $Att_i$ in $\mathcal{T}$ do
    $List_A \leftarrow$ Find semantically relevant attributes according to Algorithm 3
    if $List_A \neq \emptyset$ then
        Construct an OR-statement of $Att_i$ and all attributes in $List_A$
        Replace $Att_i$ with the constructed OR-statement in $\mathcal{T}$
    else
        Keep the attribute $Att_i$ in $\mathcal{T}$ as it is
for every AND-statement do
    $List_B \leftarrow$ Find attributes that are semantically relevant to the AND-statement according to Algorithm 4
    if $List_B \neq \emptyset$ then
        Construct an OR-statement of the AND-statement and all attributes in $List_B$
        Replace the AND-statement with the constructed statement in $\mathcal{T}$
    else
        Keep the AND-statement in $\mathcal{T}$ as it is
for every OR-statement do
    Remove the repeated attributes
Return the result as the updated access structure, $\mathcal{T}'$
```

*nent, where* ($Connection = Alice$) *is an attribute on LinkedIn and* ($Close\ Friend = Alice$), ($Intimate\ Friend = Alice$), *and* ($Friend = Alice$) *are attributes on Facebook. Then, Alice's post on Twitter will be encrypted based on the updated AS, which means those who are connected to Alice on LinkedIn and Facebook can also decrypt and access Alice's post on Twitter as they have the private keys related to* ($Connection = Alice$), ($Close\ Friend = Alice$), ($Intimate\ Friend = Alice$), *and/or* ($Friend = Alice$) *attributes.*

Example 2 shows how SABE-SEAS not only makes the ABE schemes semantic-aware but also enables cross-domain interoperability.

## 4. Security Analysis

### *4.1. Security Assumptions*

We consider the following assumptions:

- In conventional CP-ABE schemes, a TA holds a master secret key and generates private keys. In our proposals a TA has the same trust level; particularly, in SABE-SEK, the TA keeps also the ontology secret.

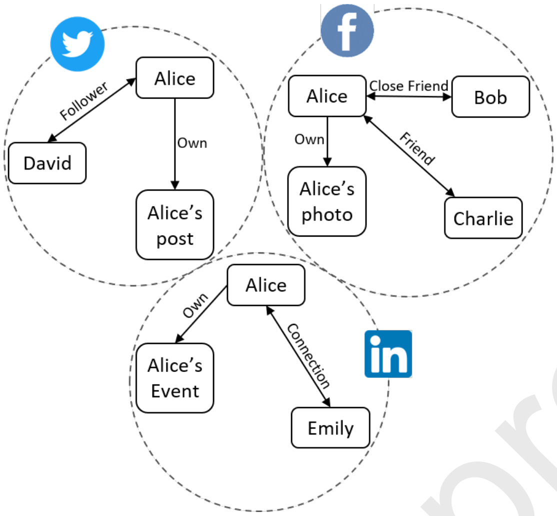

Figure 4: An example for interoperability

- We assume that the ontology that is provided to a TA during system initialization can be considered trusted in the sense that it demonstrates the correct relationships between attributes (concepts).
- In SABE-SEAS, when different domains form a federation, all the parties, i.e., TAs, should agree on a common ontology. Thus, one TA cannot change the common ontology alone and without the consent of other TAs.

### *4.2. Security Model*

The security model for both SABE-SEK and SABE-SEAS is defined using the following game, which is based on the classical indistinguishable encryption against chosen-plaintext attacks (IND-CPA) and mainly considers the confidentiality of the ciphertext.

- **Init phase**: An adversary $\mathcal{A}$ chooses an access structure $\mathcal{T}$ and sends it to a challenger $\mathcal{C}$.
- **Setup phase**: $\mathcal{C}$ generates the master secret key $MK$ and public parameters $PP$ by running the `Setup` algorithm. Then, $\mathcal{C}$ sends the public parameters $PP$ to $\mathcal{A}$ and keeps the master secret key $MK$ secret, whereas particularly for SABE-SEK $\mathcal{C}$ keeps also the ontology secret. However, in the proposed SABE-SEAS scheme, $\mathcal{C}$ generates a signature for the provided ontology and gives the ontology and the corresponding signature to $\mathcal{A}$ as part of the public parameters $PP$.
- **Phase 1**: $\mathcal{A}$ asks (like any user) from $\mathcal{C}$ private keys related to any sets of attributes. In SABE-SEK, $\mathcal{C}$ runs the `SABE.KeyGen` to generate private

keys using updated attributes, whereas in SABE-SEAS it only runs the ABE.KeyGen.

- **Challenge phase**: $\mathcal{A}$ submits two messages $M_0$ and $M_1$ of equal length. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, encrypts $M_b$ under the access structure $\mathcal{T}$, and returns the produced ciphertext $CT^*$. Note that in SABE-SEAS, $\mathcal{C}$ encrypts $M_b$ under an updated access structure generated by SABE.UpdateAS.
- **Phase 2**: $\mathcal{A}$ repeats **Phase 1** multiple times.
- **Guess phase**: $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$.

$\mathcal{A}$ wins the game if (i) $b = b'$ and (ii) none of the sets of attributes that were requested by $\mathcal{A}$ (including, in the case of SABE-SEK, also attributes inferred through SABE.UpdateAtt) satisfy the access structure that was used for encryption (where in the case of SABE-SEAS is the one updated through SABE.UpdateAS). The advantage of the adversary $\mathcal{A}$ is defined as the quantity

$$Adv_{\mathcal{A}}^{IND-CPA} = |Pr[b = b'] - \frac{1}{2}|.$$

**Definition 5** (IND-CPA Secure)**.** *An SABE framework (both SABE-SEK and SABE-SEAS schemes) is* IND-CPA secure *iff $Adv_{\mathcal{A}}^{IND-CPA}$ is negligible for any probabilistic polynomial time (PPT) adversary.*

For the SABE-SEAS scheme we need something more because here an adversary may also modify the common ontology. Since we employ a secure signature scheme [12], for SABE-SEAS we consider, in addition to the IND-CPA game, also the following game that is based on the Existential Unforgeability under Chosen Message Attacks (EUF-CMA) [14, 20].

- **Setup phase**: exactly as in the previous game.
- **Queries phase**: $\mathcal{A}$ repeatedly requests signatures for chosen messages $(M_1, \ldots, M_j)$, and receives corresponding signatures $(\sigma_1, \ldots, \sigma_j)$ from $\mathcal{C}$. Here we treat ontologies as messages.
- **Forgery phase**: In the end, $\mathcal{A}$ outputs a message $M^*$ (i.e., an ontology) and a signature $\sigma^*$.

$\mathcal{A}$ *wins* the game if (i) $M^*$ was not among those messages requested by $\mathcal{A}$ in the **Queries phase**, and (ii) the signature $\sigma^*$ can be verified correctly using the public key of the trusted authority. The advantage of the adversary in this game is defined as the quantity

$$Adv_{\mathcal{A}}^{EUF-CMA} = Pr[\mathcal{A}\ wins].$$

**Definition 6** (EUF-CMA Secure)**.** *An SABE-SEAS scheme is EUF-CMA secure iff $Adv_{\mathcal{A}}^{EUF-CMA}$ is negligible for any PPT adversary.*

### *4.3. Security Proofs*

The security of our SABE-SEK and SABE-SEAS schemes can be proved by reduction to the underlying CP-ABE [11] and signature [12] schemes, i.e., if there is an attack against SABE, then the same attack can be used to break the underlying CP-ABE and/or signature schemes (but these have already been proven to be secure).

**Theorem 1.** *Both SABE-SEK and SABE-SEAS schemes are IND-CPA secure provided that the underlying CP-ABE scheme is IND-CPA secure.*

*Proof.* Assume that there exists a PPT adversary $\mathcal{A}$ that can break the proposed SABE with advantage $\epsilon$. We can construct a simulator $\mathcal{B}$ to break the underlying CP-ABE scheme with the same advantage $\epsilon$ as follows, where $\mathcal{B}$ will play two roles at the same time: (1) the challenger for the adversary $\mathcal{A}$ in the IND-CPA game for SABE; and (2) the adversary for the challenger in the IND-CPA game for the underlying CP-ABE scheme.

- **Init phase**: $\mathcal{B}$ receives an access structure $\mathcal{T}$ from $\mathcal{A}$ and sends it to $\mathcal{C}$ (in the CP-ABE scheme).

- **Setup phase**: $\mathcal{C}$ generates the master secret key $MK$ and public parameters $PP$ by running the `Setup` algorithm of the underlying CP-ABE scheme. Then, $\mathcal{C}$ sends $PP$ to $\mathcal{B}$ and keeps $MK$ secret. In SABE-SEK, $\mathcal{C}$ keeps the provided ontology secret as well. Then, $\mathcal{B}$ forwards $PP$ to $\mathcal{A}$. However, in SABE-SEAS, $\mathcal{C}$ generates also a signature for the provided common ontology and gives both to $\mathcal{B}$ as part of the public parameters. Therefore, in SABE-SEAS, $\mathcal{B}$ removes the received common ontology and its signature from the public parameters and sends only the rest to $\mathcal{A}$.

- **Phase 1**: When $\mathcal{B}$ receives a private key query for a set of attributes from $\mathcal{A}$, in SABE-SEAS, it forwards the received set of attributes to $\mathcal{C}$ to get the corresponding private keys from the underlying CP-ABE scheme. However, in SABE-SEK, $\mathcal{B}$ first updates the received set of attributes by calling the `SABE.UpdateAtt` algorithm, then sends the updated set of attributes to $\mathcal{C}$. In response, $\mathcal{C}$ generates the corresponding private keys using the `KeyGen` algorithm of the underlying CP-ABE scheme and returns the generated private keys to $\mathcal{B}$. Then, $\mathcal{B}$ forwards the received private keys to $\mathcal{A}$ in response to $\mathcal{A}$'s original query.

- **Challenge phase**: $\mathcal{A}$ sends two messages of equal length, $M_0$ and $M_1$, to $\mathcal{B}$ who forwards them to $\mathcal{C}$. Then, $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, encrypts $M_b$ under the access structure that was provided in the **Init phase**, and returns the produced ciphertext $CT^*$ (the output of the `Encryption` algorithm of the underlying CP-ABE scheme) to $\mathcal{B}$ who forwards it to $\mathcal{A}$. Note that in SABE-SEAS, $\mathcal{B}$ asks $\mathcal{C}$ to perform the encryption based on the updated access structure (the result of calling the `SABE.UpdateAS` algorithm).

- **Phase 2**: The same as **Phase 1** multiple times.
- **Guess phase**: $\mathcal{A}$ outputs a guess $c' \in \{0, 1\}$, and then $\mathcal{B}$ sends $c'$ to $\mathcal{C}$.

Based on this simulation game, it is clear that if $\mathcal{A}$ has an advantage $\epsilon$ in the IND-CPA game against the proposed SABE schemes, then $\mathcal{B}$ can attack the underlying CP-ABE scheme with the same advantage $\epsilon$. However, the underlying CP-ABE has been proven to be IND-CPA secure [11].

To explain more, recall that the proposed SABE schemes directly use the algorithms of the underlying CP-ABE scheme, i.e., we do not change the functionality of `Setup`, `KeyGen`, `Encryption`, and `Decryption` algorithms of the underlying CP-ABE scheme in any way. Indeed, only the input of the `KeyGen` and `Encryption` algorithms of the underlying CP-ABE may be changed as some more attributes may be added to the set of attributes submitted by the user for the key generation (in SABE-SEK) or to the access structure for the encryption (in SABE-SEAS). However, the type of input is still the same, and thus these two algorithms would function in the same way with the same security guarantees. □

Even if IND-CPA secure, the SABE-SEAS scheme depends also on the security of the signature scheme [12] employed to defeat ontology modification attacks. Recall that in SABE-SEAS the provided access structure may be updated based on a common ontology. An adversary may launch ontology modification attacks by adding new concepts (or relationships) in such a way that makes the attributes of the attacker be semantically related to (possibly all) other concepts, which in turn would allow to decrypt (possibly any) ciphertext.

**Theorem 2.** *The proposed SABE-SEAS scheme is secure provided that the underlying CP-ABE scheme is IND-CPA secure and the employed signature scheme is EUF-CMA secure.*

*Proof.* In SABE-SEAS, the access structure that is updated based on the common ontology is used only after the authenticity and validity of the ontology is checked using the employed signature scheme. If an adversary modifies the common ontology, then it can be detected before using the access structure. Suppose there exists a PPT adversary $\mathcal{A}$ that can attack the proposed SABE-SEAS scheme by launching ontology modification attacks. We can build a simulator $\mathcal{B}$ that can break the employed signature scheme by using the same actions as $\mathcal{A}$. In other words, the security of the proposed SABE-SEAS can be reduced to the security of the employed signature scheme, which has been proven to be EUF-CMA secure in [12]. □

## 5. Implementation and Evaluation

We implemented the proposed SABE framework based on both approaches, i.e., Semantically-Enriched Key and Semantically-Enriched Access Structure, in Java. To extend a set of attributes (in SABE-SEK) or update access structures

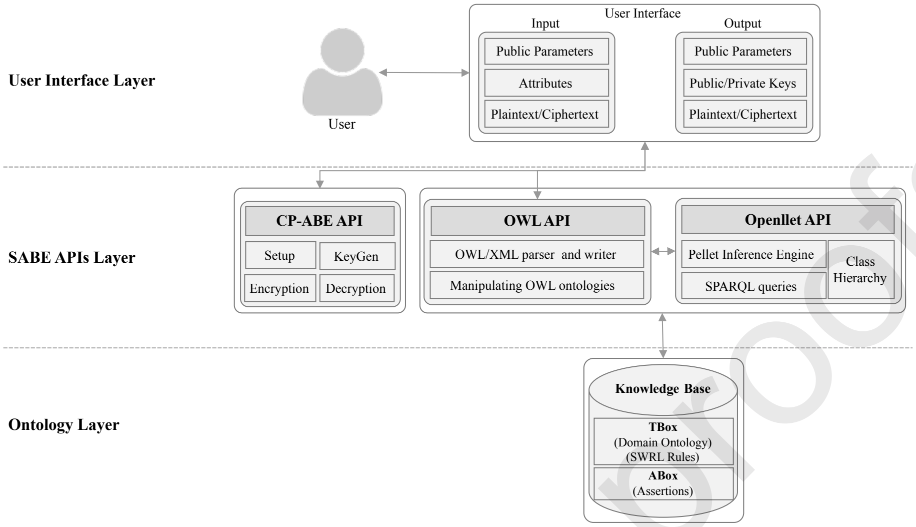

Figure 5: SABE architecture.

(in SABE-SEAS) using semantic technologies, two simple ontologies are created using Protégé [37] editor (version 5.5.0) and based on the OWL 2 data modeling language. The OWL API [24] (version 5.1.17) is used to deal with the created ontology, e.g., loading the ontology, adding assertions into the ontology, and inferring extra knowledge (implicit knowledge). It means that the semantic component in the proposed framework provides an API by which the set of submitted attributes (for the generation of a private key) or the access structures (for encryption of a data item) may be updated (based on Definition 3 and Definition 4).

The default reasoner of the OWL API is the HermiT [19] reasoner. However, the HermiT reasoner does not support SWRL built-in atoms, e.g., *swrlb:greaterThan*. Hence, the Pellet [43] reasoner provided by the Openllet library [18] (version 2.6.1) is used as the inference engine because we used some SWRL rules including SWRL built-in atoms.

As shown in Figure 5, we implemented SABE in a modular way, being possible to easily replace a module or extend it to any CP-ABE scheme. We made the source code of our implementation publicly available at `https://github.com/haamedarshad/SABE-code`.

The performance of the proposed SABE framework is evaluated by running the implementation on an Intel Core i7-8550U CPU at 1.80GHz with 32 GB RAM and Windows 10 (64-bit) computer. We added the semantic component to the `KeyGen` algorithm (in SABE-SEK) and `Encryption` algorithm (in SABE-SEAS) of a classical CP-ABE scheme [11, 48] (the CP-ABE scheme presented in [11] is selected because its source code is publicly accessible). We executed each of the `KeyGen`, `Encryption`, and `Decryption` algorithms of both the proposed

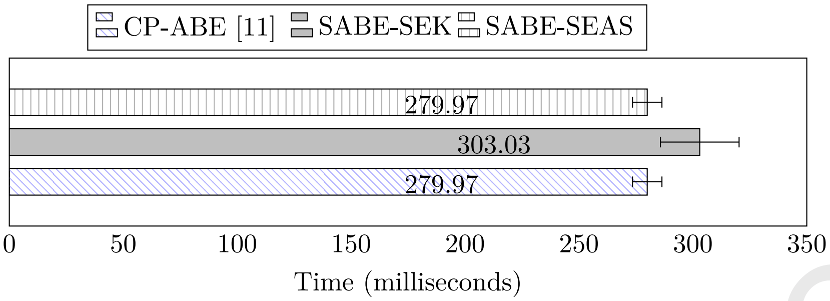

Figure 6: Execution time for key generation (with 95% confidence intervals).

SABE framework (both SABE-SEK and SABE-SEAS) and the underlying CP-ABE scheme [11, 48] 100 times to get the average execution times of the mentioned algorithms. The same input data (with the size of 1 MB, 100 MB, and 1 GB) and access structure are used for encryption and decryption experiments.

Table 1 shows the average execution times (in milliseconds) of SABE (both SABE-SEK and SABE-SEAS) and the classical CP-ABE scheme. The differences between the execution times (in milliseconds and with 95% confidence intervals) of the `KeyGen` and `Encryption` algorithms of SABE and those of the classical CP-ABE scheme are demonstrated in Figures 6 and 7, respectively. Note that the execution time of the `Decryption` algorithm is almost the same for both SABE and CP-ABE.

As can be seen in Table 1 and Figures 6 and 7, the `KeyGen` algorithm in SABE-SEK takes about 5 milliseconds (on average) more than those of the underlying CP-ABE scheme and SABE-SEAS. This difference is because of using the semantic technologies when generating a private key. As explained before, in SABE-SEK, when a request for generating a private key arrives, the set of submitted attributes will be sent to the semantic component, which loads an ontology, adds a few assertions to the ontology (all the attributes submitted

Table 1: Execution time (in milliseconds) of SABE (SEK and SEAS) vs CP-ABE

| Algorithm | Input Size | CP-ABE [11] | SABE | |
|---|---|---|---|---|
| | | | SEK | SEAS |
| Key Generation | - | 279.97 | 303.03 | 279.97 |
| Encryption | 1 MB | 94.12 | 94.12 | 140.49 |
| | 100 MB | 28.76 | 228.76 | 275.13 |
| | 1 GB | 1595.18 | 1595.18 | 1641.55 |
| Decryption | 1 MB | 33.72 | 33.83 | 33.76 |
| | 100 MB | 381.85 | 383.42 | 382.27 |
| | 1 GB | 3224.38 | 3266.48 | 3241.57 |

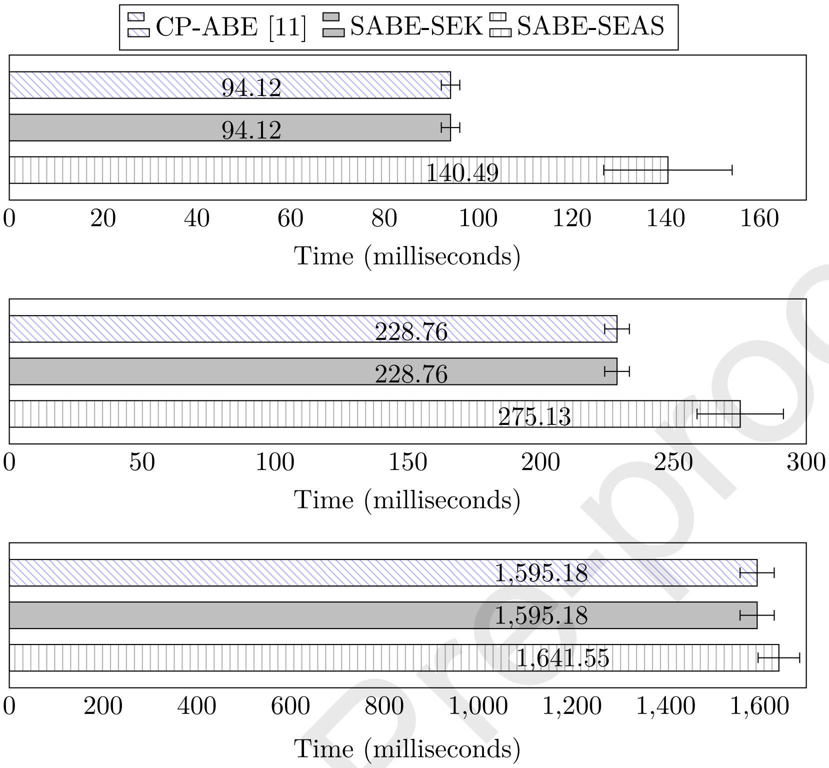

Figure 7: Execution time for encryption (with 95% confidence intervals). The input size for the top, middle, and bottom charts is 1 MB, 100 MB, and 1 GB, respectively.

by a user will be temporarily added as assertions for a dummy individual into the ABox of the knowledge base), and then performs the semantic reasoning by means of an inference engine to obtain all other attributes that are semantically relevant (according to Definition 3) to the submitted ones. Next, the semantically relevant attributes will be added to the list of attributes submitted by the user and then the `KeyGen` algorithm generates a private key based on the updated set of attributes. Therefore, the key generation process in SABE-SEK takes 5 more milliseconds, which is negligible.

The execution time of the `Encryption` algorithm in both SABE-SEK and the underlying CP-ABE is the same (see Table 1). This is because SABE-SEK uses the `Encryption` algorithm of the underlying CP-ABE scheme without any changes. Besides, the input of the `Encryption` algorithm in SABE-SEK is also the same as that in the underlying CP-ABE. However, the `Encryption` algorithm of SABE-SEAS takes in average 46 milliseconds more than those of SABE-SEK and the underlying CP-ABE. This is because in SABE-SEAS, the semantic component will be called to update the provided access structure for each encryption.

In our experiments, there were six attributes in the updated access structure that we used for encryption (in SABE-SEAS). In real scenarios and with large ontologies, more semantically relevant attributes may be inferred from the ontology, and accordingly, the updated access structure may include more attributes. Hence, we performed the encryption (in SABE-SEAS) using an access structure including 50 attributes. The average encryption time for a 1 GB input was 4.2 seconds, which is almost one second more than that with an access structure including six attributes. Therefore, large ontologies may result in bigger access structures (in SABE-SEAS), which in turn the encryption time will be increased. However, the overhead is not very high. Besides, it may be less probable to have more than 50 attributes in an updated access structure.

The size of ciphertexts may increase a little bit (only a few kilobytes as the size of an attribute is a few bytes) in SABE-SEAS as the updated access structures may include more attributes. The size of private keys in SABE-SEK may also increase in the same way.

Table 1 also demonstrates that the `Decryption` algorithm of the proposed framework (both SABE-SEK and SABE-SEAS) takes a few milliseconds more than that of the underlying CP-ABE scheme. That might be due to the fact that the private keys (in SABE-SEK) and access structures (in SABE-SEAS) may contain more attributes, and thus checking the compliance between attributes in the private key and the access structure of the ciphertext takes a few more milliseconds. Nevertheless, the difference between the execution time of the `Decryption` algorithm in the proposed framework (both SABE-SEK and SABE-SEAS) and the underlying CP-ABE scheme is negligible.

As a conclusion, it can be said that the overheads associated with adding semantic technologies to the CP-ABE scheme are reasonable. In other words, the overall experiment results are encouraging as such overheads are almost negligible.

## 6. Discussion

In this paper, we proposed two different approaches for augmenting ABE schemes with semantic technologies, i.e., Semantically-Enriched Key (SABE-SEK) and Semantically-Enriched Access Structure (SABE-SEAS). The SABE-SEK scheme makes the ABE schemes semantic-aware; but, it does not facilitate cross-domain interoperability. However, the proposed SABE-SEAS scheme provides both. SABE-SEK increases the time required for generating private keys (as the semantic component will be called for updating a submitted set of attributes) whereas SABE-SEK affects the encryption time (as a provided access structure will be updated utilizing the semantic component). SABE-SEK could be considered more advantageous since: 1) in real-life scenarios, we perform encryption much more frequently than key generation; 2) a TA, which has more resources than a user, runs the `KeyGen` algorithm, whereas the users run the `Encryption` algorithm; 3) in some applications, we may not need cross-domain interoperability so SABE-SEK would be enough (instead of the SABE-SEAS).

In the proposed SABE framework, if the common ontology changes, which is rather infrequent since an ontology describes the relationships between attributes and not users, then the affected private keys (in SABE-SEK) can be revoked [1, 41, 52] and new private keys should be generated according to the new ontology. Besides, ontology changes in SABE-SEAS may affect the existing ciphertexts that can be managed by re-encryption of ciphertexts or other methods [9, 10, 30]. Therefore, aspects regarding the *dynamicity* of the ontology have not been treated in this paper as we consider that they would easily be solved by existing techniques.

The proposed SABE framework (both SABE-SEK and SABE-SEAS) is ”CP-ABE agnostic”, which means different CP-ABE schemes can be used instead of the one that we used in our implementations. As represented in Figure 2, SABE-SEK includes `SABE.UpdateAtt` in addition to `ABE.KeyGen` (in the key generation). As demonstrated in Algorithm 1, `SABE.UpdateAtt` is added for updating the provided set of attributes (based on the semantic relationships between attributes) before calling the `KeyGen` algorithm of a classical CP-ABE scheme (i.e., `ABE.KeyGen`). Figure 3 and Algorithm 5 demonstrate that SABE-SEAS includes `SABE.UpdateAS`, which is added for updating access structures before calling the `Encryption` algorithm of a classical CP-ABE scheme (i.e., `ABE.Encryption`). The functionality of `ABE.KeyGen` and `ABE.Encryption` is not changed in any way in the proposed SABE framework. `ABE.KeyGen` and `ABE.Encryption`, respectively, in Algorithm 1 and Algorithm 5 are generic constructs that can be realized using different CP-ABE schemes. Besides, as explained throughout the paper, `SABE.UpdateAtt` and `SABE.UpdateAS` are independent of the underlying CP-ABE scheme. They only may update the provided set of attributes, which is an input of `ABE.KeyGen`, and the provided access structure, which is an input of `ABE.Encryption`, by including (in most cases) more attributes based on semantic relationships between attributes. Therefore, any CP-ABE scheme can be extended to a SABE.

### *6.1. Further System Properties*

**Property 1.** *The proposed framework is modular.*

Here we refer to the modularity of the software implementation and architecture for the SABE. As illustrated in Figure 5, the different modules of SABE are: CP-ABE API, OWL API, Reasoner (Openllet API), User Interface, and a domain ontology. This allows to replace, e.g., the underlying CP-ABE scheme, for reasons of security or performance, without changing other modules. However, we talk about static modularity, for otherwise, if we change the underlying CP-ABE scheme when the framework is in use, i.e., at runtime, then we may not be able to use the updated framework (with a new underlying CP-ABE scheme) for the decryption of the existing ciphertexts because every CP-ABE scheme generates different private keys for the same set of attributes. Nevertheless, this holds for every ABE schemes and that is not a limitation of our proposed framework.

**Property 2.** *The proposed framework is scalable.*

As described in Section 3, a common ontology is used to facilitate the interoperability between Facebook, Twitter, and LinkedIn. More online social networks can be added to the scenario by updating the common ontology. The only thing that needs to be done is creating a new ontology, which defines the semantic relationships between attributes of new domains in addition to what exists in the current ontology. Then, thanks to the modularity of the proposed framework, the current ontology can be replaced with the new ontology easily. Therefore, it can be said that the proposed framework is scalable in terms of the number of organizations/domains (in our case, online social networks) collaborating with each other. However, an issue with practical scalability could be the increase in the number of attributes existing in the updated access structures that may increase the encryption time as discussed in Section 5.

**Property 3.** *The proposed framework is extensible.*

As explained in Property 1, the proposed framework is modular and every single module can be easily changed. Hence, it is possible to extend the functionalities of the proposed framework by changing the modules. For example, the underlying CP-ABE scheme can be replaced with a new CP-ABE scheme, which may offer extra features like accountable decryption or enforceable obligations. Besides, the semantic component (OWL API and Openllet API) can be extended to provide more implicit knowledge when generating private keys or updating access structures. For instance, more SWRL rules and more advanced relationships can be defined.

**Property 4.** *The proposed framework is generic.*

SABE can be used in different environments and domains, e.g., eHealth, education, eGovernment, hardware security, cloud computing, etc., other than online social networks. This can be done by changing only the common ontology that is used in the proposed framework.

## 7. Related Work

In 2005, Sahai and Waters [42] introduced the concept of Attribute Based Encryption. They proposed a new type of Identity-Based Encryption (IBE) through which one can encrypt a piece of data for a group of recipients enabling multicast encryption [44]. After a year, Goyal et al. [22] proposed a Key-Policy Attribute Based Encryption (KP-ABE) in which ciphertexts are associated with a set of attributes and private keys are generated based on access structures. Hence, a ciphertext can be decrypted if the access structure of a private key satisfies the attributes required by a ciphertext. Bethencourt et al. [11] proposed the first CP-ABE scheme in which private keys are associated with a set of attributes and the ciphertexts are produced based on access structures. Till now, a considerable number of KP-ABE [29, 38] and CP-ABE [23, 32, 49] schemes have been proposed.

A combination of KP-ABE and CP-ABE, to have both types of ABE at the same time, was proposed by Attrapadung and Imai [3]. Müller et al. [36] proposed a CP-ABE scheme for distributed environments, where several authorities manage attributes and generate private keys. Yu et al. [51] employed proxy re-encryption and lazy re-encryption techniques to improve the efficiency of KP-ABE.

ABE schemes rely on a trusted authority in generating private keys for attributes. The trusted authority, which has full power on private keys, may behave maliciously. Thus, ABE schemes suffer from the key escrow problem. There are a huge number of research studies in the literature [26, 53] addressing the key escrow problem in ABE schemes. For instance, in [16], the key escrow problem was addressed by incorporating several TAs cooperating to generate private keys. However, such a multi-authorities ABE scheme may be susceptible to the collusion of TAs. Hu et al. [26] proposed a multi-authorities CP-ABE scheme addressing the key escrow problem and collusion attacks (i.e., the collusion of the authorities). Zhang et al. [53] proposed a multi-authorities KP-ABE scheme addressing collusion attacks and user privacy. Recently, Zhang et al. [54] proposed a novel CP-ABE scheme addressing the key escrow problem and user revocation. Li et al. [33] proposed a CP-ABE scheme that provides accountability in white-box model and addresses the privacy issues through policy hiding.

Tang and Ji [45] added a verification property to both single-authority and multi-authorities versions of KP-ABE, by which users can verify the correctness of the received private keys as errors may occur during creation or transmission of the keys. Wang et al. [46, 47] combined a hierarchical IBE scheme and a CP-ABE scheme to address the revocation problem in ABE schemes (revoking access rights from users who are no longer legitimate).

There are other research studies [17, 31] reducing the decryption overhead by means of decryption sharing and outsourcing.

## 8. Conclusions

In this paper, we have proposed the first semantic-aware attribute-based encryption framework called SABE, described in Section 3. We have proposed two different schemes: Semantically-Enriched Key (SEK) and Semantically-Enriched Access Structure (SEAS) using CP-ABE as a baseline scheme for both SABE-SEK and SABE-SEAS. In SABE-SEK, we have modified the key generation process by adding the support for semantic reasoning. The goal was to make CP-ABE schemes semantic-aware by taking into account the semantics of attributes. Algorithm 1 and Figure 2 demonstrate that any CP-ABE scheme can be extended to a SABE-SEK scheme by calling `SABE.UpdateAtt`, which updates the provided set of attributes based on the semantic relationships between attributes as defined in a domain ontology, before calling the `KeyGen` algorithm of a classical CP-ABE scheme. The proposed SABE-SEK scheme makes CP-ABE schemes semantic-aware as demonstrated in Example 1;

however, it does not enable cross-domain interoperability as different domains have different trusted authorities with different master secret keys. To provide cross-domain interoperability, we have presented the SABE-SEAS scheme in Section 3.2. In SABE-SEAS, we have used semantic technologies to update the access structures by including semantically relevant attributes in the access structures as illustrated in Algorithm 6. In other words, any CP-ABE scheme can be extended to a SABE-SEAS scheme by updating access structures (through `SABE.UpdateAS`) before encryption using the `Encryption` algorithm of a classical CP-ABE scheme, as demonstrated in Figure 3 and Algorithm 5. Example 2 demonstrates that SABE-SEAS not only makes CP-ABE schemes semantic-aware but also facilitates cross-domain interoperability.

We have formally verified the security of the proposed SABE-SEK and SABE-SEAS schemes. We have also implemented a prototype of both schemes by extending a classical CP-ABE scheme in a modular way as demonstrated in Figure 5. The source codes have been made publicly available for further research.

We have evaluated the effectiveness of the proposed SABE framework by comparing the performance of the proposed SABE-SEK and SABE-SEAS schemes and the underlying CP-ABE scheme in Section 5. The results of our experiments, as shown in Table 1, demonstrate that the key generation in SABE-SEK and encryption in SABE-SEAS take a few milliseconds (on average) more than those in the underlying CP-ABE. To assess the effect of the number of attributes on the performance, we have performed the encryption in SABE-SEAS using an access structure including 50 attributes. The results show that the average encryption time for a 1 GB input with a 50-attribute access structure is almost one second more than that with an access structure including six attributes. The results show also that the size of ciphertexts in SABE-SEAS and the size of private keys in SABE-SEK are increased (only a few kilobytes as the size of an attribute is a few bytes) compared to those in the underlying CP-ABE. Therefore, the overall experiment results confirm that SABE improves interoperability and functionality with negligible overheads.

For future work, we plan to incorporate the social network graphs in the domain ontology. Relationships between users change dynamically and sometimes quickly; thus, taking into account the dynamicity of the relationships between users in the social networks improves the quality of the cryptographic access control systems. Besides, we will include the contextual information surrounding users and resources in the social networks to provide a more fine-grained protection.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was partially supported by the Swedish Foundation for Strategic Research (SSF) and the Swedish Research Council (VR).

## References

## References

[1] Ruqayah R. Al-Dahhan, Qi Shi, Gyu Myoung Lee, and Kashif Kifayat. Survey on revocation in ciphertext-policy attribute-based encryption. *Sensors*, 19(7):1695, 2019. doi: 10.3390/s19071695. URL https://doi.org/10.3390/s19071695.

[2] Grigoris Antoniou and Frank van Harmelen. *A semantic web primer*. MIT Press, 2004. ISBN 978-0-262-01210-2.

[3] Nuttapong Attrapadung and Hideki Imai. Dual-Policy Attribute Based Encryption. In *International Conference on Applied Cryptography and Network Security*, pages 168–185. Springer, 2009. doi: 10.1007/978-3-642-01957-9_11.

[4] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003. ISBN 0-521-78176-0.

[5] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017. ISBN 978-0-521-69542-8.

[6] Randolph Baden, Adam Bender, Neil Spring, Bobby Bhattacharjee, and Daniel Starin. Persona: An Online Social Network with User-defined Privacy. In *SIGCOMM*, pages 135–146. ACM, 2009. doi: 10.1145/1594977.1592585.

[7] Mrinmoy Barua, Rongxing Lu, and Xuemin Shen. SPS: Secure personal health information sharing with patient-centric access control in cloud computing. In *IEEE Global Communications Conference (GLOBECOM)*, pages 647–652, 2013. doi: 10.1109/GLOCOM.2013.6831145.

[8] Sean Bechhofer, Frank Van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein, Guus Schreiber, and Mike Dean. Owl web ontology language reference. *W3C Recommendation*, 10(02):1–53, 2004.

[9] Sana Belguith, Nesrine Kaaniche, and Giovanni Russello. PU-ABE: Lightweight Attribute-Based Encryption Supporting Access Policy Update for Cloud Assisted IoT. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 924–927. IEEE Computer Society, 2018. doi: 10.1109/CLOUD.2018.00137.

[10] Sana Belguith, Nesrine Kaaniche, Mohammad Hammoudeh, and Tooska Dargahi. PROUD: Verifiable Privacy-preserving Outsourced Attribute Based SignCryption supporting access policy Update for cloud assisted IoT applications. *Future Generation Computer Systems*, 111:899–918, 2020. doi: 10.1016/j.future.2019.11.012.

[11] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-Policy Attribute-Based Encryption. In *IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334. IEEE Computer Society, 2007. doi: 10.1109/SP.2007.11.

[12] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. *Journal of Cryptology*, 21 (2):149–177, 2008. doi: 10.1007/s00145-007-9005-7.

[13] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 213–229. Springer Berlin Heidelberg, 2001. doi: 10.1007/3-540-44647-8_13.

[14] Dan Boneh, Emily Shen, and Brent Waters. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 229–240. Springer, 2006. doi: 10.1007/11745853_15.

[15] Dan Brickley. RDF vocabulary description language 1.0: RDF Schema. *W3C Recommendation*, 2004. URL `http://www.w3.org/TR/rdf-schema/`.

[16] Melissa Chase. Multi-authority Attribute Based Encryption. In Salil P. Vadhan, editor, *Theory of Cryptography*, pages 515–534. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-70936-7_28.

[17] Ningyu Chen, Jiguo Li, Yichen Zhang, and Yuyan Guo. Efficient CP-ABE Scheme With Shared Decryption in Cloud Storage. *IEEE Transactions on Computers*, 71(1):175–184, 2022. doi: 10.1109/TC.2020.3043950.

[18] A. Galigator. Openllet: An Open Source OWL DL reasoner for Java. `https://github.com/Galigator/openllet`, 2020.

[19] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL2 Reasoner. *Journal of Automated Reasoning*, 53(3): 245–269, 2014. doi: 10.1007/s10817-014-9305-1.

[20] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A ”Paradoxical” Solution To The Signature Problem. In *25th Annual Symposium on Foundations of Computer Science*, pages 441–448, 1984. doi: 10.1109/SFCS.1984.715946.

[21] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-Based Encryption for Circuits. *Journal of the ACM (JACM)*, 62(6):1–33, 2015. doi: 10.1145/2824233.

[22] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS ’06)*, pages 89–98, 2006. doi: 10.1145/1180405.1180418.

[23] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded Ciphertext Policy Attribute Based Encryption. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, pages 579–591. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-70583-3_47.

[24] Matthew Horridge and Sean Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011. doi: 10.3233/SW-2011-0025.

[25] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member submission*, 21:79, 2004. URL https://www.w3.org/Submission/SWRL/.

[26] Shengzhou Hu, Jiguo Li, and Yichen Zhang. Improving Security and Privacy-Preserving in Multi-Authorities Ciphertext-Policy Attribute-Based Encryption. *KSII Transactions on Internet & Information Systems*, 12(10), 2018. doi: 10.3837/tiis.2018.10.025.

[27] Vincent C Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. *NIST Special Publication*, 800(162):1–47, 2014. doi: 10.6028/NIST.SP.800-162.

[28] Yinhao Jiang, Willy Susilo, Yi Mu, and Fuchun Guo. Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. *Future Generation Computer Systems*, 78:720 – 729, 2018. doi: 10.1016/j.future.2017.01.026.

[29] Allison Lewko, Amit Sahai, and Brent Waters. Revocation Systems with Very Small Private Keys. In *IEEE Symposium on Security and Privacy*, pages 273–285, 2010. doi: 10.1109/SP.2010.23.

[30] Jianqiang Li, Shulan Wang, Yuan Li, Haiyan Wang, Huiwen Wang, Huihui Wang, Jianyong Chen, and Zhuhong You. An Efficient Attribute-Based Encryption Scheme With Policy Update and File Update in Cloud Computing. *IEEE Transactions on Industrial Informatics*, 15(12):6500–6509, 2019. doi: 10.1109/TII.2019.2931156.

[31] Jiguo Li, Yao Wang, Yichen Zhang, and Jinguang Han. Full Verifiability for Outsourced Decryption in Attribute Based Encryption. *IEEE Transaction on Services Computing*, 13(3):478–487, 2020. doi: 10.1109/TSC.2017.2710190.

[32] Jiguo Li, Ningyu Chen, and Yichen Zhang. Extended File Hierarchy Access Control Scheme with Attribute-Based Encryption in Cloud Computing. *IEEE Transactions on Emerging Topics in Computing*, 9(2):983–993, 2021. doi: 10.1109/TETC.2019.2904637.

[33] Jiguo Li, Yichen Zhang, Jianting Ning, Xinyi Huang, Geong Sen Poh, and Debang Wang. Attribute Based Encryption with Privacy Protection and Accountability for CloudIoT. *IEEE Transactions on Cloud Computing*, 10 (2):762–773, 2022. doi: 10.1109/TCC.2020.2975184.

[34] Jin Li, Yinghui Zhang, Xiaofeng Chen, and Yang Xiang. Secure attribute-based data sharing for resource-limited users in cloud computing. *Computers & Security*, 72:1–12, 2018. doi: 10.1016/j.cose.2017.08.007.

[35] Jianghua Liu, Xinyi Huang, and Joseph K. Liu. Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption. *Future Generation Computer Systems*, 52:67–76, 2015. doi: 10.1016/j.future.2014.10.014.

[36] Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed Attribute-Based Encryption. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology – ICISC 2008*, volume 5461 of *Lecture Notes in Computer Science*, pages 20–36. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00730-9_2.

[37] Mark A. Musen and The Protégé Team. *Protégé Ontology Editor*, pages 1763–1765. Springer New York, 2013. doi: 10.1007/978-1-4419-9863-7_1104.

[38] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-Based Encryption with Non-Monotonic Access Structures. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 195–203. Association for Computing Machinery, 2007. doi: 10.1145/1315245.1315270.

[39] Bill Parducci, Hal Lockhart, and Erik Rissanen. eXtensible Access Control Markup Language (XACML) version 3.0. *OASIS Standard*, pages 1–154, 2013. URL `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html`.

[40] Pablo Picazo-Sanchez, Raúl Pardo, and Gerardo Schneider. Secure Photo Sharing in Social Networks. In Sabrina De Capitani di Vimercati and Fabio Martinelli, editors, *ICT Systems Security and Privacy Protection*, pages 79–92. Springer International Publishing, 2017. doi: 10.1007/978-3-319-58469-0_6.

[41] Praveen Kumar Premkamal, Syam Kumar Pasupuleti, and PJA Alphonse. Dynamic traceable CP-ABE with revocation for outsourced big data in cloud storage. *International Journal of Communication Systems*, 34(2): e4351, 2021.

[42] Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 457–473. Springer Berlin Heidelberg, 2005. doi: 10.1007/11426639_27.

[43] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007. doi: 10.1016/j.websem.2007.03.004.

[44] Mehdi Sookhak, F Richard Yu, Muhammad Khurram Khan, Yang Xiang, and Rajkumar Buyya. Attribute-based data access control in mobile cloud computing: Taxonomy and open issues. *Future Generation Computer Systems*, 72:273–287, 2017. doi: 10.1016/j.future.2016.08.018.

[45] Qiang Tang and Dongyao Ji. Verifiable Attribute Based Encryption. *International Journal of Network Security*, 10(2):114–120, 2010. doi: 10.1.1.478.7208.

[46] Guojun Wang, Qin Liu, and Jie Wu. Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 735–737, 2010. doi: 10.1145/1866307.1866414.

[47] Guojun Wang, Qin Liu, Jie Wu, and Minyi Guo. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Computers & Security*, 30(5):320–331, 2011. doi: 10.1016/j.cose.2011.05.006.

[48] Junwei Wang. Java Realization for Ciphertext-Policy Attribute-Based Encryption. 2012. URL `\url{https://github.com/junwei-wang/cpabe/}`.

[49] Brent Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography – PKC 2011*, pages 53–70. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-19379-8_4.

[50] Lo-Yao Yeh, Pei-Yu Chiang, Yi-Lang Tsai, and Jiun-Long Huang. Cloud-Based Fine-Grained Health Information Access Control Framework for

Lightweight IoT Devices with Dynamic Auditing and Attribute Revocation. *IEEE Transactions on Cloud Computing*, 6(2):532–544, 2018. doi: 10.1109/TCC.2015.2485199.

[51] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. In *INFOCOM*, pages 534–542. IEEE, 2010. doi: 10.1109/INFCOM.2010.5462174.

[52] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 261–270. Association for Computing Machinery, 2010. doi: 10.1145/1755688.1755720.

[53] Leyou Zhang, Pengfei Liang, and Yi Mu. Improving Privacy-Preserving and Security for Decentralized Key-Policy Attributed-Based Encryption. *IEEE Access*, 6:12736–12745, 2018. doi: 10.1109/ACCESS.2018.2810810.

[54] Ruyuan Zhang, Jiguo Li, Yang Lu, Jinguang Han, and Yichen Zhang. Key Escrow-free Attribute Based Encryption with User Revocation. *Information Sciences*, 600:59–72, 2022. doi: 10.1016/j.ins.2022.03.081.