Aegis: A Semantic Implementation of Privacy as Contextual Integrity in Social Ecosystems

Imrul Kayes
Computer Science and Engineering
University of South Florida
Tampa, Florida USA 33620
Email: imrul@mail.usf.edu

Adriana Iamnitchi
Computer Science and Engineering
University of South Florida
Tampa, Florida USA 33620
Email: anda@cse.usf.edu

***Abstract*—Combining and incorporating rich semantics of user social data, which is currently fragmented and managed by proprietary applications, has the potential to more accurately represent a user's social ecosystems. However, social ecosystems raise even more serious privacy concerns than today's social networks. This paper proposes to model privacy as contextual integrity by using semantic web tools and focuses on defining default privacy policies, as they have the highest impact. Through a real implementation and performance evaluation we show that such a framework is practical.**

***Index Terms*—Privacy as contextual integrity; social ecosystems; online privacy**

I. Introduction

Social ecosystems [1] refer to the collection of rich datasets of user-to-user interactions in support of social applications. This data is collected from Internet-mediated social interactions (such as declared relationships in online social networks or tagging/contributing content in user-generated content platforms), from public profiles (to infer homophily relationships), and from phone-recorded real life interactions (such as colocation sensing and activity identification). Social ecosystems have enabled a large set of social applications, such as recommender systems [2], [3], [4], email filtering [5], [6], defending against Sybils [7], [8] and against large-scale data crawls [9]. The novel scenarios activated by social ecosystems, however, raise serious concerns regarding user privacy.

User privacy in online activities is already a hot issue due to lack of formal framing [10]. The primary aspect of social ecosystems, that of aggregating data from various sources to provide it (possibly processed) to a diversity of applications, significantly amplify the privacy concern. First, aggregated data from different contexts of activity presents a more complete and possibly uncomfortable picture of a person's life. Second, data is to be exposed to a variety of applications, themselves from different contexts of activity, from personal to professional.

Numerous solutions addressed privacy in social ecosystems, typically in the context of a particular system [11], [12], [13], [14], [15] or for particular application scenarios [16], [17], [18]. Little addressed, however, is the setting of a *default* privacy policy that protects the user and, at the same time, allows the user to benefit from application functionality. While users are invited to change the default privacy settings, in reality very few do it. For example, more than 99% Twitter users retained the default privacy setting where their name, list of followers, location, website, and biographical information are visible [19]. Other studies [20], [21], [22] show that the majority of Facebook users have default or permissive privacy settings. More worrisome, when the default settings are not matched with user preferences, they almost always tend to be more open, exposing the content to more users than expected [23]. Users' unwillingness to change the default policy is sometimes aggravated by the complexity of the process; default privacy controls are too cumbersome to properly understand and use [24], [25], [26].

The privacy challenge is fundamentally due to the lack of a universal framework that establishes what is right and wrong [10]. Nissenbaum proposed such a framework in her formulation of privacy as contextual integrity [27]. To the best of our knowledge, one line of work approaches privacy as contextual integrity by proposing a formal language for expressing generic privacy expectations [28]. We take the problem one step further by focusing not only on policy specification, but also on designing and prototyping the enforcing of contextual integrity-based default policies for social ecosystems.

In this work we employ semantic web techniques to implement Nissenbaum's framework for defining privacy as contextual integrity, with a specific focus on defining application and platform-independent default privacy settings. In the context of the classification proposed in [29], this work addresses the problem of *social privacy*, that aims to protect user information from other users and applications running on other users' behalf. To this end, we propose Aegis, an extensible, fine-grained privacy model for social ecosystems based on semantic web technologies. The model implements the basic concepts of Nissenbaum's privacy framework: social contexts, norms of appropriateness, and norms of distribution. It builds on ontologies used to encode social data and implicitly represent social contexts, and on Resource Description Framework (RDF) statements/SPARQL queries to define and verify access to data. This work extends our previous efforts [30] with a refined data model, the implementation of the prototype and experimental evaluation.

The contributions of this work are:

- We propose an ontology-based social ecosystem data model to capture users aggregated social data from diverse sources (e.g., Facebook, LinkedIn etc.). This data model can be used to acquire information from an unrestricted set of social sources and export it to an ever-evolving collection of socially-aware applications and services.
- We employ semantic web technologies to generate default privacy policies based on Nissenbaum's contextual integrity theory. These policies are extensible, fine-grained and expressive enough to be changed by the user. Furthermore, the policy model is generic enough to be used in other systems.
- We provide an architecture and a prototype implementation of our privacy model that automatically enforces access control policies on a social ecosystem knowledge base. Our experimental evaluation on three real-world large networks demonstrates the applicability in practice of our solution.

The rest of the paper is organized as follows. Section II introduces the contextual integrity theory and discusses its relevance to social ecosystems. Section III describes the system and data models, and the system architecture. We present our policy specification in Section IV. Section V presents our prototype implementation and experimental evaluation. Section VI reviews related work and Section VII concludes.

II. Privacy as Contextual Integrity in Social Ecosystems

While notoriously difficult to define [31], privacy is understood as an individual's right to determine to what extent her data can be communicated to others. Privacy is typically seen not simply as the absence of information about us in the minds of others, but rather as the control we have over information about ourselves [32], [33].

Social ecosystems, which combine users' social information from diverse sources and incorporates richer semantics, pose a daunting task in terms of privacy enforcement. It has to exercise a more complex representation of users' social world, ranging from object-centric domains (e.g., common preferences) to people-centric domains (e.g., declared friendship relationships). Privacy-preserving default policy generation in such a complex system could be leveraged by contextual integrity, a social theory-based account of privacy proposed by Nissenbaum [27]. Instead of defining the term "privacy", Nissenbaum proposes a reasoning framework for privacy as contextual integrity, where privacy is seen as neither a right to secrecy nor a right to control, but a right to an appropriate flow of *information about an individual* (referred to as "personal information").

Nissenbaum's account of privacy as contextual integrity is based on two non-controversial facts. First, every transfer of personal information happens in a certain social context and all areas of life (and online activity makes no exception [10]) are governed by context-specific norms of information flow. Second, people move among a plurality of distinct contexts, thus altering their behavior to correspond with the norms of those contexts, aware to the fact that information appropriately shared in one context becomes inappropriately shared into a context with different norms. For example, it is appropriate to discuss romantic entanglements with friends, financial information with banks, and work-related goals with co-workers, but sharing romantic experiences with the bank is out of place.

On the basis of these facts, Nissenbaum suggests that contextual integrity is maintained when two types of norms are upheld: *Norms of appropriateness* and *Norms of distribution*. Norms of appropriateness circumscribe the type of information about persons that is appropriate to reveal in a particular context. So, it is appropriate to share medical information with doctors, but generally not appropriate to share it with employers. Implemented in social ecosystems, this type of norm specifies *where* context-specific data can be communicated. For example, if Alice is a colleague of Bob in the professional context, then requests from Alice regarding Bob's gaming context such as the games owned by Bob should be denied, as the requests do not comply with the norms of appropriateness.

Norms of distribution cover the transfer of a third party's personal information from one user to another. In a social ecosystem, the norm of distribution suggests a default policy that restricts the distribution of information. For example, if Alice and Bob have a shared content—e.g., Bob's picture that he shared with Alice—then a request from Charlie to Alice regarding the content should not succeed without Bob's consent, even if Alice owns Bob's picture now.

III. System Model and Architecture

Nissenbaum's framework is articulated for protecting the citizen from an overly curious government. For the digital context, her approach works best as a default privacy policy, which is precisely the focus of this paper. Thus, our proposed system, Aegis, enforces default policy as contextual integrity by modeling two assumptions from real world. First, information is always tagged with the context in which it is revealed. Second, the scope of privacy norms is always internal to a context. To implement this, Aegis implements the constructs of user roles and actions, resources, contexts, and privacy norms.

A. System Model

Similar to Dey et al.'s definition of a context [34], we define the social context of a user as the collection of social information that describes the user in a domain. For example, data about Bob's education, skills, and LinkedIn connections describe Bob's *Professional* context.

Our system model is defined by the following:

1) there is an unrestricted set of disjoint social contexts in the system;
2) a user belongs to only one social context at any time;
3) a user can have one or more roles in every social context s/he is part of;
4) each piece of data (resource) is assigned to only one context; however, users can share a resource with other

users, in which case the resource is replicated in each of the other users' current contexts;

5) a request for a resource is made on behalf of the requester's role in the particular context in which the requester is when the request is made;
6) a request specifies an action, which could be *read*, *write*, *delete* or *replicate* to another user's ownership.

Note that in real life users can be simultaneously part of multiple contexts: for example, Alice is both a friend and a colleague for Bob. However, at any given time, only one of these contexts will be considered, perhaps the prominent one given the physical environment (e.g., at work) or based on a system-wide ranking of contexts (e.g., work has higher priority over friendship, to limit sensitive data exposure).

Implementing contextual integrity in the default privacy policies is thus reduced to implementing the norms of appropriateness and distribution in this system model.

B. Modeling Social Contexts and Roles

We model social contexts, and therefore the entire social ecosystem (consisting of a set of social contexts), based on ontologies. An ontology is a set of entities, instances, functions, relations and axioms, and is used as a vocabulary for expressing the knowledge of a domain.

The traditional advantages of using ontologies apply in this case as well: first, an ontology provides a common vocabulary, thus it ensures formal and structured representation of users' contextual data. Second, using ontologies provide semantic interoperability, thus data can be used by a variety of applications. Third, high-level logic inferences are possible as data model have semantics associated with it. For example, if Bob has content in his professional context (*contents subClassOf ProfessionalContext*) and *recommendations* are content, then *recommendations* are inferred to belong to Bob's professional context. Finally, a social ecosystem can be built incrementally by adding new context ontologies. We can also reuse existing web ontologies from different domains to meet the demand of an exhaustive scale social ecosystems ontology. For example, an ontology [35] is already available to model bloggers' interest in a blogging community.

To represent the data model, we classify online social contexts into entity classes (e.g., friendship context, professional context, blogging context). The context classification is inherent in the modeling process because context definition depends on underlying entities and relationships among the entities.

Our context ontology is divided into an upper ontology and domain-specific ontologies. The upper ontology is a high-level ontology which captures general contexts (e.g., Friendship, Professional, Gaming). The domain-specific ontologies are collections of low-level ontologies which define the details of general contexts and their properties in each sub-domain.

Figure 1 shows a class hierarchy of the entities considering some online contexts of a user, where the top level class (root) is the context itself. All generic contexts are subclasses of the root context entity and all domain-dependent descriptors (classes, properties) have some common properties to inherit from the root. The lower level sub-classification expresses the domain dependence of the contexts.

The addition of new social contexts to the ecosystem happens naturally, with the implementation of new sensors for new social signals (see next section): the developers of social sensors have to be aware of the ontology of the social contexts to which the sensors report, in order to maintain structural data representation. Another way to extend the social ecosystem is by extending a social context itself when new relevant social signals become available: for example, Facebook recently added a service called Gifts which allows users to buy presents for their friends. Consequently, the social context model needs to be adaptive to accommodate additions of new contexts. Ontologies help in designing a scalable context model.

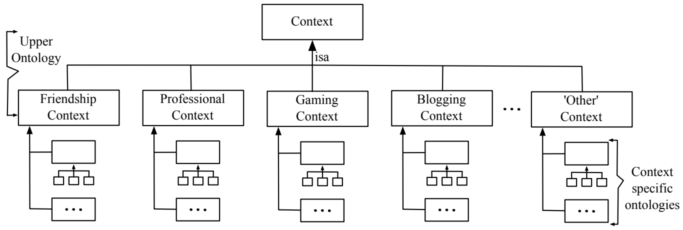


Fig. 1: Context entities and their domain dependent elements.

Figure 2 gives an example of three contexts of a user's digital world: Professional, Friendship and Gaming. The representation of the ontology is *person*-centric which gives a user-oriented viewpoint of the data model. The three large circles in the model are the contexts; each circle encodes context-specific knowledge and they are subclasses of Context.

Roles are modeled as relationships: for example, *isColleagueOf* in Alice's ecosystem specifies that Bob has the role of a colleague in her professional context. Roles, as relationships, are thus asymmetrical: Charlie might be a follower in Alice's followers ecosystem but Alice might not be Charlie's follower.

We use OWL [36] to model social contexts. OWL is more expressive than other ontology languages such as RDFS [36]. Moreover, W3C Web Ontology Working Group has defined OWL from an existing rich language DAML+OIL [1].

C. Aegis Architecture

Our general architecture fits the Social Hourglass infrastructure introduced in [37]. The focus of this work is the Privacy Management Layer, presented in Figure 3. It receives input from users' personal aggregators and outputs privacy-compliant social data to applications. The input from each user's personal aggregator is a labeled, directed ego net, that represents the user's recorded social interactions with other users, with each type of interaction semantically tagged.

A brief explanation of the *Social Data Acquisition and Aggregation Layer* at the bottom of Figure 3 (that belongs to the social hourglass infrastructure) is presented next. (For

[1] http://www.daml.org/2000/12/daml+oil-index.html

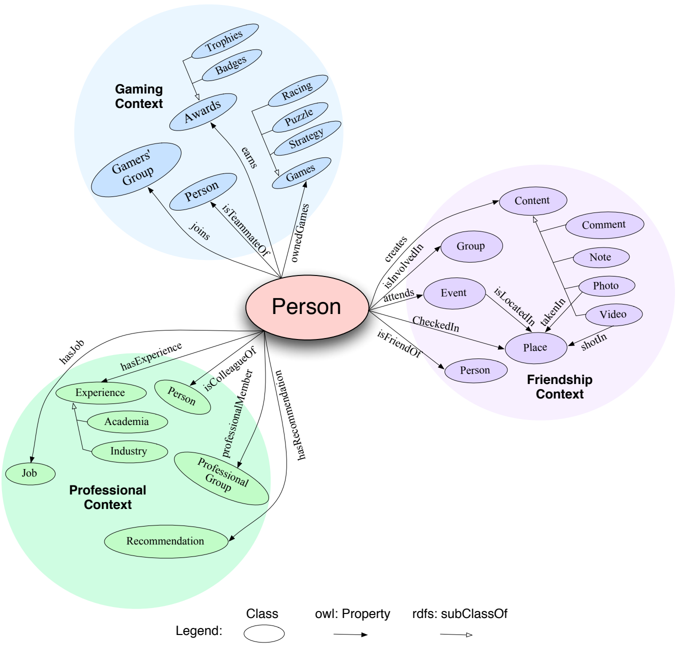


Fig. 2: A partial definition of social ecosystems ontology considering Professional, Friendship and Gaming contexts. A circle represents a context, which contains context specific classes (e.g., declared social groups).

more details we refer the reader to [37].) A social sensor is an application running on behalf of a user on a user's device as an independent application or in the browser) and observing one or more social signals (for example, Facebook interactions of the user with other users). It reports processed social data in the form of <contact ID, type of interaction, strength of interaction> to the user's aggregator. The aggregator runs on the user's trusted device (e.g., mobile phone or home computer, but not on a shared computer or a commercial service). It processes all such information received from the social sensors deployed on behalf of its user and reports an aggregated and personalized social edge to the *Social Data Management Layer* and to the *Contextual Policy Definer*. For user *ego*, this social edge is of the form <ego, alter, context, weight>, where *alter* is a user *ego* interacted with in *context* with the interaction strength *weight*. Social data management can be implemented by various solutions; to provide *surveillance privacy* protection [29], distributed solutions such as Prometheus [38] can be used.

Social sensor design and implementation are context specific: for example, a LinkedIn sensor observes its user's professional data and a Facebook sensor observes the user's friendship data based on the ontology shown in Section III-B. In addition to requirements related to sensor accuracy and performance, sensor design should address the following. First, a particular sensor can target one context only (thus, report one label only), but is capable of collecting data from different social signals. For example, a gaming sensor could collect gaming related data from multiple services, e.g., Stream[2] and Giantbomb[3]. Moreover, by using the ontology vocabulary, sensors should be able to distinguish context-specific data from the wealth of social data existing in a service. Second, sensors should be able to cope with changes in ontology and act immediately.

The Privacy Management Layer in Figure 3 is responsible for managing and enforcing privacy policies, and thus for extracting and applying the default policies as well. This component communicates with the Social Data Management Layer which implements social contexts and roles.

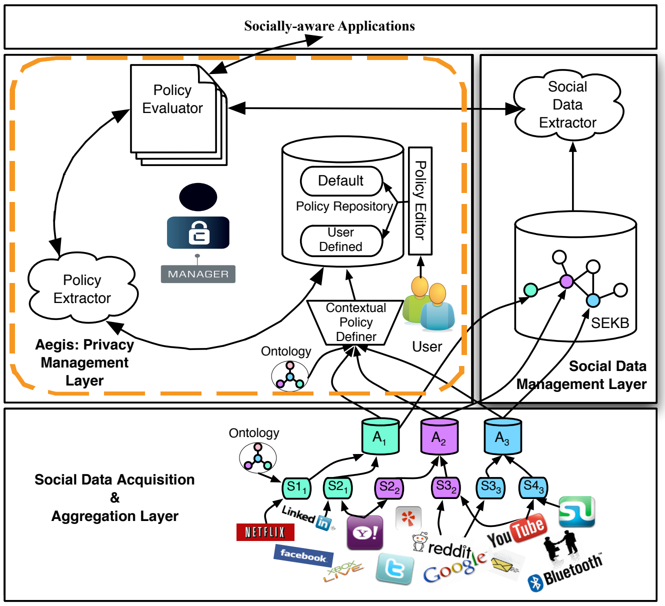


Fig. 3: A layered architecture of social data collection, personalization and management for socially aware applications along with Aegis, as a form of privacy management layer.

The *Contextual Policy Definer* generates default access control policies based on a social ontology and the contextual integrity norms and stores them in the *Policy Repository*. Policies in the *Policy Repository* can be edited with the GUI-based *Policy Editor*. The *Contextual Policy Definer* generates default policies based on the following rule: only roles in a user's social context are allowed access to the user's data associated with that particular context. An example of a default policy extracted with this rule is the following: all users with a *Colleague* role in Bob's *Professional* context can access (all) his data associated to the *Professional* context. Our policy model is granular; it defines a policy for every resource covering all the contexts a user could belong to.

The *Policy Manager* consists of extractor and evaluator for handling access requests. In particular, any access request is intercepted by the *Policy Evaluator*, which evaluates the policy. Permitted access requests are finally fulfilled by returning data from the social ecosystem knowledge base (SEKB) through

[2] http://www.steamcommunity.com/
[3] http://www.giantbomb.com/

social data extractor. The policies are stored in the policy repository and the policy extractor extracts policies from the policy repository.

IV. Policy Specification

A policy is defined as a set of RDF statements. As shown in the architecture, the contextual policy definer generates policies that obey the two information norms of contextual integrity: norms of appropriateness and distribution.

Let us take as example a policy generated by the policy definer for the resource *groups* in the *Professional* context: Bob's colleagues can read his professional group involvement in the *Professional* context. The policy can be formalized as the following SPARQL query, <Policy>, where the prefixes `p:` and `se:` represent the namespace of the policy model and of the social ecosystems model, respectively:

```
<Policy>
ASK
where {
 ?req rdf:type p:requestor.
 ?req p:allowed p:read.
 p:read p:performedOn Bob.
 ?req se:isColleagueOf Bob.
 Bob se:professionalMember ?group.}

<Augmented Policy>
ASK
where {
  Alice rdf:type p:requestor.
  Alice p:allowed p:read.
  p:read p:performedOn Bob.
  Alice se:isColleagueOf Bob.
  Bob se:professionalMember ?group.}
```

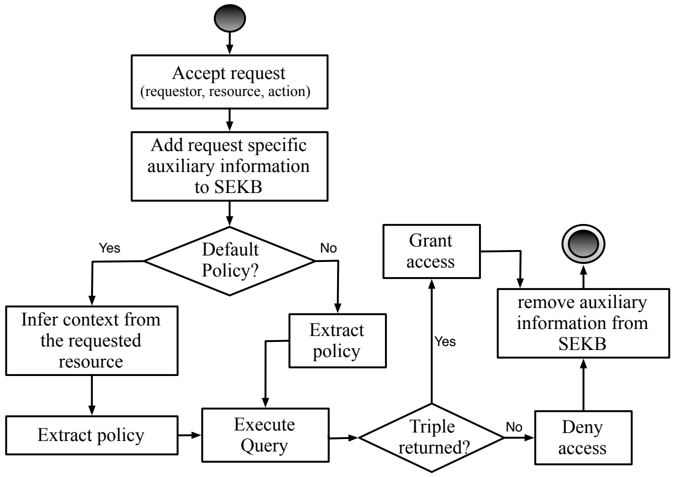

Accept request
(requestor, resource, action)
Add request specific auxiliary information to SEKB
Yes
Default Policy?
No
Infer context from the requested resource
Extract policy
Extract policy
Execute Query
Triple returned?
Yes
No
Grant access
Deny access
remove auxiliary information from SEKB

Fig. 4: Request handling process.

A basic access request is a triple *<rstr, rsc, act>*, where *rstr* is a user who requests the access (e.g., an instance of *se:Person*), *rsc* is the resource requested (e.g., *se: Photo*), and *act* is the type of action (read/insert/delete).

When a request such as "Alice wants to see Bob's professional group involvement" comes to the policy manager, the predefined policy variable `?req` will be replaced by Alice as shown by the augmented policy. The policy evaluator will temporarily insert policy-related auxiliary RDF statements to the knowledge base, such as the first three statements of the augmented SPARQL query, and executes the query over the modified knowledge base. The above policy representation states that the access request will be granted if Alice and Bob are colleagues. The same access request from Bob's teammate in the *Gaming* context will be denied because of lack of appropriate triples in the SEKB, thus implementing the norm of appropriateness.

Similarly, the system will disallow access to a resource that is shared or co-owned with someone, upholding the norms of distribution. For example, the following policy restricts Charlie's access to Bob's photos that he previously shared with Alice.

```
<Policy>
ASK
where {
 ?req rdf:type p:requestor.
 ?req p:allowed p:read.
 p:read p:performedOn Bob.
 ?req se:isFriendOf Bob.
 Bob se:hasPhoto ?photo.
 ?photo se:status se:notShared}
```

Our policy representation is granular, as it allows policies for each resource. For the request, the policy manager will infer the context and will decide whether the default or personalized policy will be enforced (see policy evaluation flow chart from Figure 4). As in the data model we have hierarchy among classes (which eventually define resources) and a group of classes belong to a context, we can infer the context from requested resource. Note that a default auto generated policy could be personalized by the user and in this case, the personalized policy will be evaluated. For example, from a requested resource *recommendation*, a context inference is possible from the following SPARQL query to knowledge base:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-
rdf-syntax-ns#
PREFIX rdfs:<http://www.w3.org/2000/01/
rdf-schema#
PREFIX se:<http://www.dsg.cse.usf.com/se>

SELECT ?superClass
Where {
    se:requestedResource rdfs:subClassOf
?superClass .}
```

The policy engine will extract the policy of the inferred context and execute it.

V. Experimental Evaluation

We have implemented a prototype of Aegis in Java Platform Standard Edition 6 (Java SE 6). We used the capabilities offered by Jena[4] to implement both the knowledge base and the policy manager. Jena is a framework for building semantic web applications, and provides a collection of tools and Java libraries to develop semantic web and linked-data applications, tools and servers. Jena is currently the most comprehensive framework to manage RDF and Web Ontology Language (OWL) data in Java applications as it provides APIs for RDF data management, an ontology API for handling OWL and RDFS ontologies and a query engine compliant with the SPARQL specification. We leverage TDB[5] for persistent storage of knowledge base.

Aegis was deployed and evaluated on a machine equipped with 2 GHz Intel Core i7 processor, 4 GB 1333 MHz DDR3 RAM, Mac OS X Lion 10.7.5 operating system, and Java 1.6 runtime environment.

Our evaluation of the prototype implementation of Aegis had the following objectives. First, we wanted to evaluate the performance of the policy engine in executing default policies for realistic workloads with a realistically large number of users. For this, we chose three real social network datasets from different domains and experimented with default policy enforcement. Second, we wanted to investigate the scalability of the policy engine in executing default policies. Ideally, the policy engine should scale well with the size of the social ecosystem knowledge base. Finally, we wanted to measure the overhead induced by default policies.

A. Experimental Setup

We constructed social ecosystems knowledge base from three different real networks. Table I presents a summary of these datasets.

- soc-Slashdot0811 (from [39]): a network of friend/foe links between the users of Slashdot, a news website which features user-submitted and editor-evaluated technology-oriented news. Using Slashdot Zoo feature, users can tag each other as friends or foes.
- BlogCatalog (from [40]): a blogging website where registered users can create online profiles, post blogs, and automatically receive blogging updates from the users with whom they have declared "friend" relationships.
- Facebook (from [41]): a highly popular online social network. The dataset contains friend links of the users.

To provide test cases, we selected 13 sizes ranging from 100 to 70,000 users from the above networks. To create a sub-graph of each size, we randomly picked a node as a seed in a network and applied snowball sampling algorithm. Although snowball sampling is biased toward high-degree nodes, it preserves the topological structure of a graph [42]. For each sampled sub-graph, we created a SEKB containing nodes of type *Person* and the relationships among them. More specifically, a ego's (user) connections are randomly labeled according to the data model (*se: isFriendOf, se: isColleagueOf, se: isTeammateOf*) to abstract a user's social ecosystem and contexts (*Friendship, Professional* and *Gaming*). Also, we added users *FriendshipGroup, ProfessionalGroup* and *GamingGroup* using a random string generator as resources in relevant contexts to invoke different test cases.

We considered two types of responses: (type1) positive authorization access control response and (type 2) negative authorization access control response. Type 1 accesses are allowed, while type 2 are denied by the default policies. To this end, we generated two types of access requests. They are as follows: User U1 belongs to the context C1 of user U2 and she requests U2's resource R1 from the same context C1. And, User U1 belongs to the context C1 of user U2 and she requests U2's resource R2 from different context C2. For each sample size of each dataset we evaluated both requests 10 times and report the average evaluation time. Moreover, we performed the same experiments with no policy enforcement to measure the policy enforcement overhead.

TABLE I: Summary of the real networks used.

Network	Num. of Users	Num. of Edges
soc-Slashdot0811(Slashdot)	77,360	905,468
BlogCatalog	88,784	4,186,390
Facebook	63,731	1,545,686

B. Results

The performance results of the policy engine in executing default policies are shown in Figure 5. It shows positive and negative authorization access time and the number of requests answered by the policy engine per second for random authorizations. Our observations are as follows:

First, for all datasets and both types of authorizations, the time needed to fulfill access requests increases linearly with the size of the social ecosystem knowledge base (SEKB). As such, inference time of the default policies vary according to the size of the SEKB.

Second, for the same size of SEKB, positive and negative authorization take about the same time. Intuitively, a positive authorization should take less time than a denied request due to less scanning in the knowledge base. To asses the significance of the time difference, we ran a two sample *t*-test in which we compared the time taken for positive and negative authorization time for all sizes of SEKB. A *t*-test determines if two sets of data are significantly different from each other [43]. We obtained a *p*-value of 0.96, thereby confirming that the difference is not statistically significant. This is due to the implementation of the semantic data store, TDB: data structures in TDB use TDB B+Trees, a custom implementation of threaded B+Trees. The threaded nature implies that long scans such as negative authorizations of indexes (it uses triple and quad indexes) proceeds without needing to traverse the branches of the tree.

[4] http://jena.apache.org/index.html

[5] http://jena.apache.org/documentation/tdb/index.html

Third, for up to $10,000$ users in the SEKB, both accepted and denied access request execute fast on our tested machine (tens of milliseconds). However, as the knowledge base increases with the number of users, performance decreases. This is more visible for the BlogCatalog dataset, which has about three times more edges per node than the other datasets (see Table I). This behavior is due to the stress SEKB puts on memory: a denser graph requires more memory, thus with the increase in the number of users represented, penalties related to swapping will take place. Obvious solutions to this performance limitation include 1) increasing system memory to realistic capacity for an in-production server and 2) employing distributed solutions for SEKB data management.

To evaluate the overhead introduced by the policy engine for executing default policies, we tested the time needed to execute request with and without default policies in place. For each sampled size, we took the average access time for positive and negative authorizations both with and without default policies. Figure 6 shows the comparison. The difference between access requests with and without policies ranges from $3.17ms$ to $12.06ms$. To assess the significance of this overhead, we ran a two sample *t*-test in which we compared the access time with and without default policies. The *p*-value of this *t*-test is 0.81, which implies the overhead is statistically insignificant. So, we conclude that the action of default policy enforcement does not impose a significant burden on social ecosystems.

One of the limitations of the workloads is that they only contain ego-nets and social groups. However, a social ecosystem ontology is a more diverse collection of entity types and relations (as shown in Figure 2). A long chain of context inferences, such as "X is a photo, which is content, and the content belongs to the friendship context" will likely take longer time. Moreover, overlapping contexts (such as professional and friendship and gaming) will create denser ego-nets, hence more memory required per user. Consequently, the scalability plots shown in Figure 6 will change, also function of the available physical memory. However, the limited availability of appropriate real-world traces prevented us from doing more sophisticated performance analysis.

VI. Related Work

Different solutions have been proposed to control access to users data on social networking applications in response to increasing popularity in this type of applications. In this section we discuss privacy models and frameworks that are targeted to social systems.

Trust-based access control policies are inspired by research and development in trust and reputation in social networks. Kruk [44] proposes Friend-of-a-friend (FOAF)-Realm, an ontology-based access control mechanism. FOAF uses RDF to describe relations among users. The D-FOAF system [45] is a FOAF ontology-based distributed identity management system for social networks, where information inherent in social networks is utilized to provide community-driven access rights delegation. Both systems use a generic definition of relationships ("knows") as a trust metric and generates rules that control a friend's access to resources based on degree of separation in the social network. This approach that uses the degree of separation as the only way to quantify the level of relationship between two users ignores the relationship type. Choi et al. [46] consider named relationships (e.g., worksWith, isFriendOf, knowsOf) in modeling trust. A more nuanced trust-related access control model is proposed by Carminati et al. [47] based on relationship type, degree of separation, and trust level between users in the network.

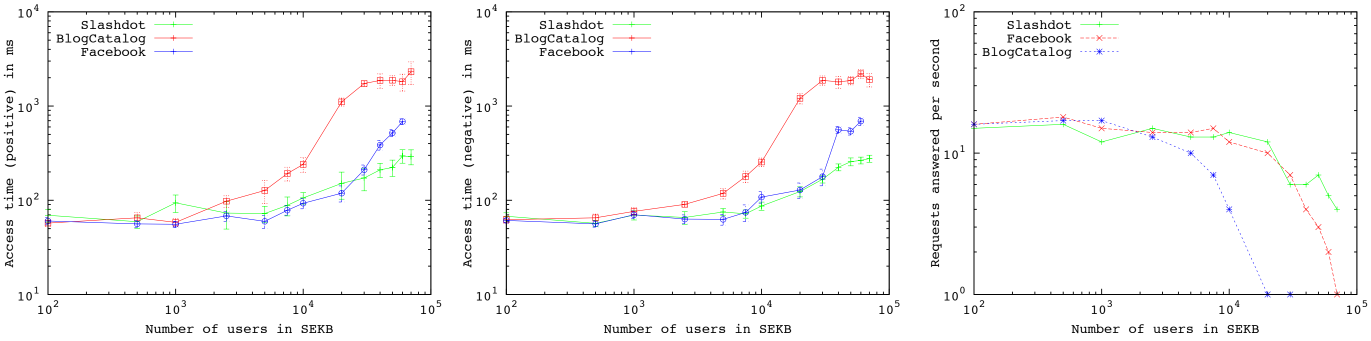
An inherent problem in trust-based privacy models is that the trust threshold values should be smoothed as much as possible. In practice, it is difficult to comprehend and specify appropriate trust thresholds without prior threshold value tuning experiments. Our approach avoids this problem by not using trust (always difficult to define), but by capturing instead the information semantics using an ontology-based access control policy .

Semantic rule-based policies have also emerged as a promising choice to control access to users social data. Rule-based policies represent the social knowledge base in an ontology (e.g., OWL) and define policies as Semantic Web Rule Language (SWRL) rules[6]. Access request related authorization is provided by reasoning on the social knowledge base. Systems that leverage OWL and SWRL to provide rule-based access control framework are [48], [49]. Although conceptually similar, [49] provides richer OWL ontology and different types of policies; access control policy, admin policy and filtering policy. The practicality of these solutions is difficult to evaluate in the absence of a proof-of-concept implementation. A more detailed semantic rule-based model is [50], which also provides a proof-of-concept implementation.

Rule-based privacy models have several limitations. First, authorization is provided by forward reasoning on the whole knowledge base, challenging scalability with the size of the knowledge base. Second, all authorizations must be recomputed if a change occurs in the social knowledge base. And finally, rule management is complex and requires a team of expert administrators [51]. In our approach the social knowledge base can be easily distributed, such that a user's trusted peer handles the user-related social data requests (like in [38]). Furthermore, re-computation of all policies is not required in case of knowledge base changes.

Role and Relationship-Based Access Control (ReBAC) are other types of privacy models that employ roles and relationships in defining privacy policies. Fong [52] proposes a ReBAC model based on the context-dependent nature of relationships in social networks. This model targets social networks that are poly-relational (e.g., teacher-student relationships are distinct from child-parent relationships), directed (e.g., teacher-student relationships are distinct from student-teacher relationships) and tracks multiple access contexts that are organized into a tree-shaped hierarchy. When access is requested in a context, the relationships from all the ancestor contexts are combined with the relationships in the target

[6] http://www.w3.org/Submission/SWRL/



(a) (b) (c)

Fig. 5: (a) Access time for positive authorization (b) Access time for negative authorization (c) Number of requests answered per second.

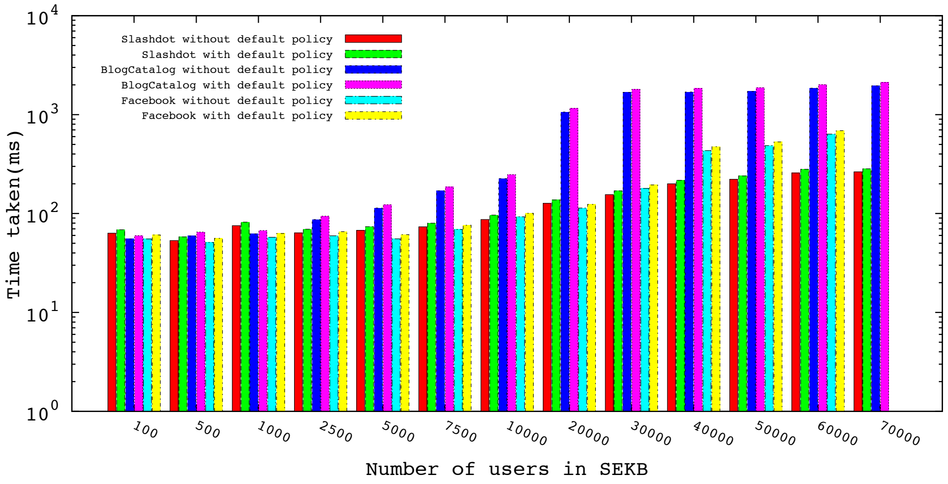


Fig. 6: Performance overhead of the policy engine with and without default policy enforcement.

access context to construct a network on which authorization decisions are made. Our work is similar in that we also model relationships in a social context as the means to access and distribute social data. But our objective is different, as we do not target particular social networks, but generate default policies for aggregated social data that could be accessed by diverge social applications.

Giunchiglia et al. [53] propose RelBac, another relation-based access control model to support sharing of data among large groups of users. The model defines permissions as relations between users and data, thus separating them from roles. The formalization of the RelBac model as an entity-relationship model allows for its direct translation into description logics, which also allows reasoning. The model, however, does not provide any precise social aspect and lacks auto generation of default policies.

The work conceptually closest to our paper is PriMa [54]. PriMa also auto generates access control policies for users, acknowledging the fact that it is perhaps not wise to rely on regular users to manually set up their access control policies because of the growing complexity of the social network and diversity of user contents. The policies in PriMa are generated based on intuitive factors such as average privacy preference of similar and related users, accessibility of similar items in similar and related users, closeness of owner and accessor (measured by the number of common friends), popularity of the owner (i.e., popular users have sensitive profile items), etc. Access control policies for profile items are finally generated aggregating these factors. This approach is vulnerable to highly volatile policies due to changes in these factors. Moreover, a large number of factors and their parametrized tuning contribute to longer policy generation and enforcement time. Unfortunately, these limitations are not addressed, so it is difficult to judge their impact in practice. Another auto-generated policy framework is PolicyMgr [55], based on a supervised learning mechanism. PolicyMgr leverages user-provided example policy settings as training sets and build classifiers that are the basis for auto-generated policies to regulate access to user profile objects. Again, its practicality in terms of response time has not yet been shown.

Our privacy model differs from the solutions above by the focus on generating default policies for a social ecosystem that deals with users' aggregated social data from different domains; the existing solutions target single application scenarios. Moreover, most of those solutions do not take target default policy generation as a primary goal. Furthermore, to the best of our knowledge, we are the first to consider a privacy framework proposed by social theorists and translate it into an architecture and proof-of-concept implementation.

VII. Summary and Discussion

In this paper, we have proposed a privacy model for social ecosystems based on the semantic web standard. The privacy model leverages contextual integrity for generating default policies that protect user's information from other users. We designed an architecture in support of the proposed privacy model, demonstrated its feasibility by building Aegis, a prototype implementation, and evaluated its performance and scalability using three large real networks. The experimental evaluation shows that our system scales well, and policy enforcement does not impose significant overhead.

Aegis addresses "social privacy" [29] problems such as those that emerge through the necessary renegotiation of boundaries as social interactions get mediated by OSN services. Social privacy problems occur when access to data is inappropriately protected due to wrong default or personalized settings. Often the default settings serve the business model of the service provider rather than the user's interests, following the "opt out" model. Aegis mitigates social privacy threats by generating default privacy policies that restrict user information to be shared or transferred inappropriately. At the same time, Aegis does not restrict users from choosing personalized and maybe relaxed privacy settings.

Although our privacy model is designed for targeting user's aggregated social data, the model is generic enough to be used in existing online social networks. For example, Google Plus and Facebook allow users to select the type of relationship with another user. This information can be leveraged to provide higher granularity in social privacy and to implement privacy as contextual integrity for default privacy settings.

While Aegis addresses social privacy, it may aggravate *surveillance* and *institutional* privacy. Surveillance privacy threats arise when users' personal information and social interactions are leveraged by authorities or service providers. Institutional privacy [56] refers to those privacy problems related to users losing control and oversight over the aggregation, processing and mining of their online social information. The aggregation of social data in social ecosystems and the ontology-based labeling (thus, the addition of processed information) creates new sensitive data that would not have been directly available. For example, users' context-specific data (such as work, personal, etc.) would increase the accuracy of user profiling to an overly curious, possibly hostile political regime. These problems are alleviated by implementing the social ecosystem as a distributed architecture, as in [38]. A distributed architecture eliminates the need for a central, omniscient authority that is in a privileged position to observe all the activity in the system.

One of the limitations of our work is that we could not experiment with a real social ecosystem due to the unavailability of users' data from multiple sources. Ideally, a social ecosystem should be an aggregation of social data from various social network platforms (e.g., Facebook, LinkedIn, Steam). Instead, we took three large networks and constructed social ecosystems from those networks. Experiments on a real social ecosystem would give more insights on our system.

Our future work includes the validation of the policy framework in emerging application scenarios, more specifically targeting applications that are built on aggregated social data. These social applications will run on users trusted devices and their access to social data store will be managed by Aegis. Furthermore, in order to experiment with Aegis in a real social ecosystem, we plan to create an "intelligent" mapping of the users in different social network datasets. This mapping will create a unification of identities from datasets and abstract a single user on multiple data sources. We also want to understand the system in different platform settings, such as peer-to-peer and mobile computing.

Acknowledgements The US National Science Foundation supported this research under grants CNS 0952420. We also thank the anonymous reviewers for their input. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] A. Toninelli, A. Pathak, A. Seyedi, R. S. Cardoso, and V. Issarny, "Middleware support for mobile social ecosystems," in *COMPSAC Workshops*, 2010, pp. 293–298.

[2] P. Avesani, P. Massa, and R. Tiella, "Moleskiing.it: a trust-aware recommender system for ski mountaineering," *International Journal for Infonomics*, 2005.

[3] O. Celma, "Foang the music: Bridging the semantic gap in music recommendation," in *Proceedings of the International Semantic Web Conference*, 2006, pp. 927–93.

[4] J. Golbeck and M. M. Wasser, "Socialbrowsing: integrating social networks and web browsing," in *CHI '07 extended abstracts on Human factors in computing systems*, 2007, pp. 2381–2386.

[5] J. Kong, B. Rezaei, N. Sarshar, V. Roychowdhury, and P. Boykin, "Collaborative spam filtering using e-mail networks," *Computer*, vol. 39, no. 8, pp. 67 –73, aug. 2006.

[6] J. Golbeck and J. Hendler, "Reputation network analysis for email ltering," in *Proceedings of the First Conference on Email and Anti-Spam*, 2004.

[7] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '06. New York, NY, USA: ACM, 2006, pp. 267–278. [Online]. Available: http://doi.acm.org/10.1145/1159913.1159945

[8] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, ser. NSDI'12, 2012.

[9] M. Mondal, B. Viswanath, A. Clement, P. Druschel, K. P. Gummadi, A. Mislove, and A. Post, "Defending against large-scale crawls in online social networks," in *Proceedings of the 8th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT'12)*, Nice, France, December 2012.

[10] H. Nissenbaum, "A contextual approach to privacy online," *Daedalus*, vol. 140, no. 4, pp. 32–48, 2011.

[11] L. Banks and S. Wu, "All friends are not created equal: An interaction intensity based approach to privacy in online social networks," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, vol. 4, 2009, pp. 970 –974.

[12] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 135–146.

[13] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 351–360. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772727

[14] P. Fong, M. Anwar, Z. Zhao, M. Backes, and P. Ning, *A Privacy Preservation Model for Facebook-Style Social Network Systems*. Springer Berlin / Heidelberg, 2009, pp. 303–320.

[15] F. Stutzman and J. Kramer-Duffield, "Friends only: examining a privacy-enhancing behavior in facebook," in *Proceedings of the 28th international conference on Human factors in computing systems*, 2010, pp. 1553–1562.

[16] J. Finnis, N. Saigal, A. Iamnitchi, and J. Ligatti, "A location-based policy-specification language for mobile devices," *Pervasive and Mobile Computing*, vol. 8, pp. 402–414, 2010.

[17] K. P. N. Puttaswamy and B. Y. Zhao, "Preserving privacy in location-based mobile social applications," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, 2010, pp. 1–6.

[18] E. De Cristofaro, A. Durussel, and I. Aad, "Reclaiming privacy for smartphone applications," in *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, march 2011, pp. 84 –92.

[19] B. Krishnamurthy, P. Gill, and M. Arlitt, "A few chirps about twitter," in *Proceedings of the first workshop on Online social networks*, 2008, pp. 19–24.

[20] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, 2005, pp. 71–80.

[21] A. Acquisti, R. Gross, G. Danezis, and P. Golle, *Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook*. Springer Berlin / Heidelberg, 2006, vol. 4258, pp. 36–58.

[22] B. Krishnamurthy and C. E. Wills, "Characterizing privacy in online social networks," in *Proceedings of the first workshop on Online social networks*, 2008, pp. 37–42.

[23] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Analyzing facebook privacy settings: user expectations vs. reality," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011, pp. 61–70.

[24] C. Lampe, N. B. Ellison, and C. Steinfield, "Changes in use and perception of facebook," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008, pp. 721–730.

[25] H. R. Lipford, A. Besmer, and J. Watson, "Understanding privacy settings in facebook with an audience view," in *Proceedings of the 1st Conference on Usability, Psychology, and Security*, 2008, pp. 2:1–2:8.

[26] K. Strater and H. R. Lipford, "Strategies and struggles with privacy in an online social networking community," in *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, 2008, pp. 111–119.

[27] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Review*, vol. 79, no. 1, pp. 119–158, 2004.

[28] A. Barth, A. Datta, J. Mitchell, and H. Nissenbaum, "Privacy and contextual integrity: framework and applications," in *IEEE Symposium on Security and Privacy*, 2006, pp. 184–198.

[29] S. Gürses and C. Diaz, "Two tales of privacy in online social networks," *Security Privacy, IEEE*, vol. PP, no. 99, pp. 1–1, 2013.

[30] I. Kayes and A. Iamnitchi, "Out of the wild: On generating default policies in social ecosystems," in *IEEE ICC'13 - Workshop on Beyond Social Networks: Collective Awareness*, June 2013.

[31] H. Nissenbaum, *Privacy in context: Technology, policy, and the integrity of social life*. Stanford Law Books, 2010.

[32] C. Fried, "Privacy," *Yale Law Journal*, vol. 77, no. 3, pp. 475–483, 1968.

[33] P. M. Regan, *Legislating Privacy: Technology, Social Values, and Public Policy*. University of North Carolina Press, 1995.

[34] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Hum.-Comput. Interact.*, vol. 16, no. 2, pp. 97–166, Dec. 2001.

[35] M. Nakatsuji, Y. Miyoshi, and Y. Otsuka, "Innovation detection based on user-interest ontology of blog community," in *Proceedings of the 5th International Semantic Web Conference*, 2006, pp. 515–528.

[36] M.Smith, C. Welty, and D. McGuinness, "Web ontology language (owl) guide," http://www.w3.org/TR/owl-guide/, 2004.

[37] A. Iamnitchi, J. Blackburn, and N. Kourtellis, "The social hourglass: An infrastructure for socially aware applications and services," *IEEE Internet Computing*, vol. 16, pp. 13–23, 2012.

[38] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi, "Prometheus: User-controlled p2p social data management for socially-aware applications," in *11th International Middleware Conference*, November 2010.

[39] J. Leskovec, "Social computing data repository at ASU," http://snap.stanford.edu/data/, 2008. [Online]. Available: http://socialcomputing.asu.edu

[40] R. Zafarani and H. Liu, "Stanford large dataset collection," http://socialcomputing.asu.edu, 2009.

[41] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[42] J. Illenberger and G. Fltterdb, "Estimating network properties from snowball sampled data," *Social Networks*, vol. 34, no. 4, pp. 701 – 711, 2012.

[43] D. W. Zimmerman, "Teachers corner: A note on interpretation of the paired-samples t test," *Journal of Educational and Behavioral Statistics*, vol. 22, no. 3, pp. 349–360, 1997.

[44] S. Kruk, "Foaf-realm: control your friends access to the resource," in *In Proceedings of the 1st Workshop on Friend of a Friend*, 2004.

[45] S. Kruk, S. Grzonkowski, H. Choi, T. Woroniecki, and A. Gzella, "D-foaf: Distributed identity management with access rights delegation," in *Proceedings of the 1st Asian Semantic Web Conference (ASWC 2006)*, 2006, p. 140154.

[46] H. C. Choi, S. R. Kruk, S. Grzonkowski, K. Stankiewicz, B. Davis, and J. Breslin, "Trust models for community aware identity management," in *Proceedings of the Identity, Reference and Web Workshop, in conjunction with WWW 2006*, 2006, p. 140154.

[47] B. Carminati, E. Ferrari, and A. Perego, "Rule-based access control for social networks," in *Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems*, 2006, pp. 1734–1744.

[48] N. Elahi, M. Chowdhury, and J. Noll, "Semantic access control in web based communities," in *Proceedings of the 2008 The Third International Multi-Conference on Computing in the GlobalInformation Technology*, 27 2008-aug. 1 2008, pp. 131 –136.

[49] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "A semantic web based framework for social network access control," in *Proceedings of the 14th ACM symposium on Access control models and technologies*, ser. SACMAT '09, 2009.

[50] A. Masoumzadeh and J. Joshi, "Ontology-based access control for social network systems." *IJIPSI*, vol. 1, no. 1, pp. 59–78, 2011.

[51] R. Engelmore, Ed., *Readings from the AI magazine*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1988.

[52] P. W. Fong, "Relationship-based access control: protection model and policy language," in *Proceedings of the first ACM conference on Data and application security and privacy*, 2011, pp. 191–202.

[53] F. Giunchiglia, R. Zhang, and B. Crispo, "Relbac: Relation based access control," in *Fourth International Conference on Semantics, Knowledge and Grid*, 2008, pp. 3 –11.

[54] A. Squicciarini, F. Paci, and S. Sundareswaran, "Prima: an effective privacy protection mechanism for social networks," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 320–323.

[55] M. Shehab, G. Cheek, H. Touati, A. Squicciarini, and P.-C. Cheng, "User centric policy management in online social networks," in *IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*, 2010, pp. 9 –13.

[56] K. S. Raynes-Goldie, *Privacy in the Age of Facebook: Discourse, Architecture, Consequences*. Curtin University., 2012.