Privacy Dynamics: Learning Privacy Norms for Social Software

Gul Calikli[1], Mark Law[2], Arosha K. Bandara[1], Alessandra Russo[2], Luke Dickens[3], Blaine A. Price[1], Avelie Stuart[4], Mark Levine[4], and Bashar Nuseibeh[1,5]

[1]Computing and Communications, The Open University, UK, {firstname.lastname}@open.ac.uk

[2]Department of Computing, Imperial College, UK, {mark.law09, a.russo}@imperial.ac.uk

[3]University College London, UK, {l.dickens}@ucl.ac.uk

[4]Psychology, University of Exeter, UK, {a.stuart, m.levine}@exeter.ac.uk

[5] Lero, University of Limerick, Ireland, UK, {bashar.nuseibeh}@lero.ie

ABSTRACT

Privacy violations in online social networks (OSNs) often arise as a result of users sharing information with unintended audiences. One reason for this is that, although OSN capabilities for creating and managing social groups can make it easier to be selective about recipients of a given post, they do not provide enough guidance to the users to make informed sharing decisions. In this paper we present *Privacy Dynamics*, an adaptive architecture that learns privacy norms for different audience groups based on users' sharing behaviours. Our architecture is underpinned by a formal model inspired by social identity theory, a social psychology framework for analysing group processes and intergroup relations. Our formal model comprises two main concepts, the group membership as a *Social Identity (SI) map* and privacy norms as a set of *conflict* rules. In our approach a privacy norm is specified in terms of the information objects that should be prevented from flowing between two conflicting social identity groups. We implement our formal model by using inductive logic programming (ILP), which automatically learns privacy norms. We evaluate the performance of our learning approach using synthesised data representing the sharing behaviour of social network users.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Privacy; I.2.6 [**Artificial Intelligence**]: Induction—*inductive logic programming*; D.2.11 [**Software Architectures**]: Domain-specific architectures—*adaptive privacy*

Keywords

Adaptive privacy, online social networks, inductive logic programming, social identity theory

Pre-publication Manuscript of paper accepted at SEAMS 2016 May 16-17, 2016, Austin, TX, USA

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

1. INTRODUCTION

The number of active users of Online Social Networks (OSNs) is now measured in millions [2]. As of November 2015, Facebook was ranked at the top with 1.55 billion active users. There has also been a significant increase in the users of LinkedIn, Twitter and Instagram since September 2014. Besides the growth in the number of users, the level of user engagement with Facebook has significantly increased compared to previous years [1, 2, 10]. As the use of OSNs has grown, privacy violations due to inappropriate sharing of information on OSNs have also become common. As a result of sharing posts with unintended audiences, many people have lost their jobs [12, 28, 34] and some have lost their health insurance [25], while some people experienced serious damage in their relationships with their spouses, friends or family members [14, 35].

Privacy violations are mostly due to the misalignment of users' "imagined audience" with the "actual audience" [24]. A user's "imagined audience" is the mental conceptualisation of people with whom he/she is communicating [3, 24], which is an essential feature of social media interactions because with social media technologies the audience is not necessarily physically co-located with the user at the time the information is shared. For instance, on Facebook a user might accidentally send a photo of himself drunk at a party to his boss as well as his friends, since his "imagined audience" does not contain his boss, while his actual audience does. In this paper, we propose a solution that aims to provide guidance to OSN users in order to minimise privacy violations resulting from such misalignment. One of the key concepts of our proposed solution is the notion of *conflicts* that is used to identify people who are in the actual audience and but not in the imagined audience (i.e. the unintended audience).

In order to model the co-presence of multiple groups within one context in OSNs irrespective of time and space, we employ *Social Identity (SI)* theory, which is a theoretical analysis of group processes and intergroup relations proposed in the social psychology literature. Social Identity refers to our sense of ourselves as members of a social group and the meaning that group has for us [18]. According to SI theory, people belong to multiple groups and social identities are created through group memberships. Being *British*, being a *researcher* or being a *student* are all examples of social identities. Lampinen et al. [22] empirically show that multiple groups are co-present on Facebook users' sites

and hence, multiple social identities corresponding to these groups are salient to the users. This empirical study also shows that Facebook users are aware of the co-presence of multiple groups on their sites and in order to prevent privacy violations users form implicit groups by mentally dividing individuals into subgroups. Therefore in addition to *conflicts*, another key concept of our proposed solution is the *SI map*, where each implicit group is modelled as a social identity group. Users can employ current OSN features (e.g., Facebook's Friend List or Google+ Circle features) to explicitly define groups so that certain posts can be shared within certain groups. However, to avoid privacy violations due to misalignment of imagined and actual audiences, users need help in maintaining the membership of the OSN groups. Our solution provides guidance to users, based on an analysis of their sharing behaviours, so that in the ideal case the defined OSN groups converge to the groups in the user's *SI map*.

In order to prevent privacy violations, which might arise due to the dynamic nature of groups, we propose an adaptive architecture, that can provide sharing recommendations to users as well as assisting them to re-configure the groups they explicitly defined using OSN settings. The core of our adaptive architecture uses a model based on social identity theory and is implemented by employing a logic based learning system. Our use of logic based learning complements other techniques that use statistical learning [8, 29] to achieve adaptive audience recommendations because it can provide explanations for why certain individuals should not be in the audience for a given information item. The learned privacy norms are the *conflicts* between social identity groups, say A and B, indicating that a given information type should not flow from group A to group B. The information types defined in these norms are specified using attributes of the information objects being shared. We explain how social identity theory informs the formal model that underpins the core of our adaptive architecture in this paper through a motivating scenario. We also explain how we implemented the core of our adaptive architecture by using a logic based learning system. The performance of this learning system is evaluated through experiments using synthesised data representing the sharing behaviour of social network users. The experimental results demonstrate that our approach can learn privacy norms with $50-70\%$ specificity, depending on the number of *conflicts* present, given at least 10 examples of user's sharing behaviour. This indicates the learnt norms cover the majority of instances of inappropriate sharing actions. Additionally, we find that if the user updates the group membership of their *SI map* to be more closely aligned to the social identity groups with whom they avoid sharing information, we can learn privacy norms with high specificity given fewer examples.

The rest of this paper is organised as follows: Section 2 describes a motivating scenario explaining how we structure the privacy violation problem in OSNs by using our key concepts, which are *conflicts* and *SI map*. Section 3 explains the *Privacy Dynamics Architecture*, which is the architecture of our proposed privacy-aware adaptive system. Section 4 describes the logic based learning system we employed in order to implement the core of our adaptive architecture. Section 5 is our evaluation of the performance of the core of our privacy-aware adaptive architecture. In Section 5, we also explain the design of our experiments and present the results. Section 6 covers related work and compares our approach with some of these studies. Finally, in Section 7 we discuss our conclusions and future work.

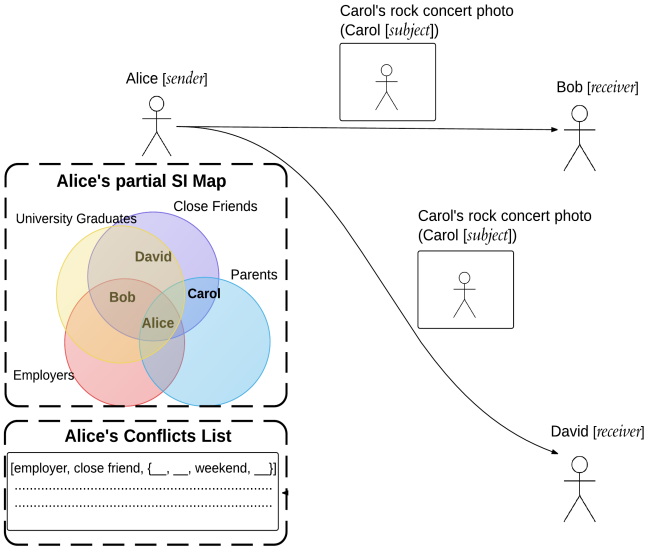

Figure 1: Structuring the privacy violation problem using Social Identity Theory.

2. MOTIVATING SCENARIO

In this section, we present a motivating scenario to explain how social identity theory can help us understand privacy problems in OSNs. In particular, this scenario focusses on how the misalignment between "actual" and "imagined" audience leads to privacy violations.

Our motivating scenario is as follows: Alice has Facebook friends Carol, Bob and David. Alice knows Bob and David from the university where they all graduated ten years ago and they are close friends. Alice first met Carol at a parents' evening at her son's school and they have become close friends since then. Alice later finds out that Carol and David are very close friends and Bob is Carol's boss. They are all aware of the mutual friendship through Facebook. While Alice and Carol are at a party on a weekend, Alice takes some photos and tells Carol that she is going to tag them and post them on Facebook. Carol tells her to make sure they are not seen by Bob because he is her boss and hence she does not want Bob to know what she does on the weekends. Alice duly removes Bob from the recipient audience for her post and then shares the photos on Facebook. For the next few weeks Alice remembers that Bob is Carol's boss and makes sure not to share posts about her weekend activities, which involve Carol with Bob.

Some months later one weekend Carol and Alice go to a rock concert together and Alice takes a photo of Carol diving off the stage and posts it to Facebook, forgetting that Bob is part of her default share group. The following week Bob meets Carol at work and refers to the photos from the concert. Carol is embarrassed by the exchange and is annoyed with Alice for not respecting her privacy.

In our proposed solution (Figure 1), each user would have a social identity map (SI map) and a set of social identity conflict rules (SI conflicts). This knowledge would be used to alert Alice of the potential privacy violation by learning that information about Carol's weekend activities should not have been shared with Bob.

A user's local SI map is their perception of social identities for themselves and the people they communicate with on the OSN. In our example, multiple groups are co-present on Alice's Facebook account and some of these groups correspond to Alice's "parent", "employer", "university graduate" and "close friend" social identities in her local SI map (Figure 1). This represents the knowledge that Alice thinks of herself as belonging to the "parents", "employers", "university graduates" and "close friends" social identity groups. The social identity group membership of Bob, David and Carol according to Alice's SI map can be seen in Figure 1. For instance, Alice thinks that Carol belongs to "close friends" and "parents" social identity groups, and that Bob belongs to the "employers" group. On the other hand, according to Alice's conflicts list that is also shown Figure 1, SI conflicts specify norms, such as the one that "employers" social identity group conflicts with the "close friend" social identity group, when the shared information was captured on the weekend. This is equivalent to stating that information that was captured on the weekend should not flow from the "close friends" social identity group to the "employers" social identity group.

In this section, we have explained how our proposed solution detects privacy violations by referring to the user's SI map and SI conflicts through a motivating scenario. In the next section, we explain details of the *Privacy Dynamics Architecture*, that is designed to integrate our solution with OSN platforms.

3. PRIVACY DYNAMICS ARCHITECTURE

Our goal is to support developers interested in integrating OSNs into an application, which we call a *SocialApp*, to provide effective privacy management capabilities that are adaptive to users' behaviour. To this end we propose the *Privacy Dynamics Architecture* as a layer of privacy enhancing technologies that can be integrated between the user interface of *SocialApp* and the OSN platform (Figure 2). The proposed architecture has at its core a *Privacy Inference (PI) engine*, which uses knowledge about the user's sharing history and social identity groups to learn privacy norms as rules about information types that should not be shared from one group to another. The focus of this paper is to explain and evaluate how this component uses inductive logic programming techniques to provide this adaptive privacy capability.

The background knowledge required by the PD architecture is stored in a *Privacy Dynamics (PD) repository*, having been acquired from the OSN platform through a *OSN platform monitor*; a *content analysis module* and the *SocialApp* itself. When the user wishes to use the OSN integration features of the *SocialApp* to share an information object, an *audience recommendation engine* queries the PD repository for the user's SI map and *conflicts*. The latter are learned privacy norms to determine if the actual audience contains recipients who are members of conflicting social identity groups. The actual audience members who are in such conflicting groups are reported to the *SocialApp* as potentially unintended recipients of the object being shared. Because the identification of these recipients is derived using a logic program, our solution can provide a rationale for its recommendation. This allows the user to decide whether or not to change the audience of the shared object thus making a final share decision and also updating their SI map.

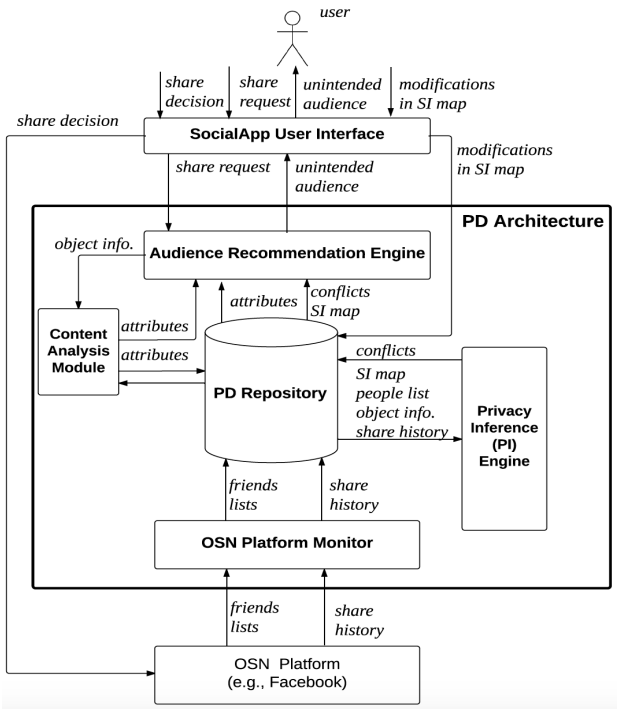

Figure 2: Privacy Dynamics (PD) architecture

The organisation of the PD architecture is based on the MAPE-K pattern for adaptive systems [21], and each component architecture is described further below.

Knowledge	PD repository data
Person	unique identifiers $\mathtt{p_i}$ of the user and user's friends
SI Map	SI group name, $\mathtt{s_i}$, and list of people, $\mathtt{p_1}, \ldots, \mathtt{p_n}$, in group
Object	unique identifiers $\mathtt{o_i}$ of objects being shared
Attribute	attribute names and values associated with each object
Share History	identifier $\mathtt{o_i}$ of objects and people, $\mathtt{p_i}$ with whom the object is shared
Conflict	conflicting SI group names, $\mathtt{s_x}, \mathtt{s_y}$, and attribute values of objects for which conflict applies

Table 1: PD repository data description

PI engine: This is the core of our proposed architecture; it uses inductive logic programming techniques to learn the conflicts in SI conflicts list. In order to learn the conflicts, the PI engine takes as input the SI map, people list, list of social identities, objects list and share history that are stored in PD repository. The learning process will be triggered whenever the user makes a sharing decision that is inconsistent with the current conflict rules that have been learned by the PI engine. The learning procedure used by the PI engine is detailed in Section 4. *PD repository*: This is another core component of the architecture, serving as the knowledge base for the PI engine and audience recommendation engine. The repository stores observed attributes such as *people list*, *objects list*, the history of the user's sharing actions and *social identity groups list*, which are described in Table 1.

Content analysis module: This module takes the object to be shared as input and outputs object attribute types and values. The content analysis module executes at periodic intervals by taking as input the objects in the share history from the PD repository. It can also be triggered by the audience recommendation engine, when the engine receives a share request including an object with unknown attributes. In such situations, it produces results at run time. The content analysis module could either use machine learning techniques to identify the attribute types and values for shared information objects, or use attributes assigned to the objects by the OSN platform itself. In our current implementation we have assumed the latter, where the PI engine only uses attributes that can be extracted by using Facebook's Graph API.

OSN platform monitor: This module periodically fetches user's friends lists (e.g., customised friends lists on Facebook, circles in Google+) and share history. This component is also used to bootstrap the system by using the retrieved friends lists to populate the initial SI map in PD repository.

Audience recommendation engine: Our audience recommendation engine generates the sharing recommendation informing the user about potentially unintended receivers. The recommendation engine is triggered by the user's share request, which consists of information about the object to be shared and the potential receivers of the object. In order to generate a recommendation, the engine performs the following steps:

- Extracts the information about the potential receivers and the attributes of the object to be shared, including the subject, from the share request;
- Queries the SI map in the PD repository to retrieve the social identity groups that the subject belongs to;
- Uses the attribute values of the object being shared to query the repository to determine if there are any conflicts associated with the subject's social identities and the object. This results in a list of potentially conflicting social identity groups;
- For each of the potentially conflicting social identities, check if there is an overlap between members of the social identity groups and the potential recipients selected by the user. Any recipients who are in a conflicting social identity group are added to the set of unintended recipients;
- Return the set of unintended recipients to the SocialApp user interface.

The set of unintended recipients can be reported to the user through the SocialApp user interface. The developer of the application can allow the user to use the unintended recipient information to modify the audience of their share request before posting the object to the social media platform. Additionally this user interface can be used to update the SI map used by the PD architecture.

4. LEARNING PRIVACY NORMS

In this section, we describe our approach for learning privacy norms, based on conflicts between social identities, from examples of shared and not shared information. These norms are assumed to be expressed in a declarative language, which can be more easily accessible to the user. We therefore make use of Inductive Logic Programming (ILP) [30]. In ILP, the goal is to learn a logic program H, called a *hypothesis*, which explains a set of positive and negative examples (E^+ and E^-) in the context of another logic program B called the *background knowledge*.

Several recent frameworks for ILP [7, 4, 23] have encoded the search for hypotheses in ASP (Answer Set Programming) [17], which is an efficient declarative logic programming paradigm. The programs we present in this paper use only a subset of the ASP language; specifically, the programs are *normal logic programs*. A *normal rule* is of the form $\mathtt{h}$:- $\mathtt{b}_1, \ldots, \mathtt{b}_m, \mathtt{not}\ \mathtt{b}_{m+1}, \ldots, \mathtt{not}\ \mathtt{b}_n$, where $\mathtt{h}$ and $\mathtt{b}_1, \ldots, \mathtt{b}_n$ are atoms. $\mathtt{h}$ is called the *head* of the rule and $\mathtt{b}_1, \ldots, \mathtt{b}_m, \mathtt{not}\ \mathtt{b}_{m+1}, \ldots, \mathtt{not}\ \mathtt{b}_n$ is called the *body*. The meaning of the rule is that if $\mathtt{b}_1, \ldots, \mathtt{b}_m$ are true and $\mathtt{b}_{m+1}, \ldots, \mathtt{b}_n$ are false then $\mathtt{h}$ must be true. A rule with an empty body is called a *fact*. In this paper, the ASP programs we consider consist entirely of normal rules. Given an ASP program, each solution to the program is a full instance of the model, called an *answer set*[1].

We now describe the underlying decision model for determining the people with whom an object should be shared, if social identities and conflicts among social identities were known. We will then present how our learning approach can automatically learn the conflicts between social identities from user's sharing behaviours.

Our model assumes a set of social identities (which are sets of people), a set of objects, with given attributes, and a set of rules defining the conflicts between social identities on each of these objects. We formalise the sharing behaviours of a set of social identities as an ASP program whose single answer set specifies which objects should be shared with whom, taking into account the conflicting social identities. This program infers that an object can be shared with everyone who is in at least one of the sharer's social identities unless there is a reason not to share the object with them (e.g., a conflict between a social identity of the subject of the object and a social identity of the receiver). We assume an object to be characterised by a set of attributes, given in table 2, and that for each object, an attribute has a single value. We can therefore write an object as a unique tuple of attributes $\langle \mathtt{subject}, \mathtt{location}, \mathtt{time}, \mathtt{day} \rangle$.

Attribute	Possible Values
subject	$\mathtt{p}_1, \ldots, \mathtt{p}_n$
location	office, beach, gym, shop, bar, night_club, cinema
time	day_time, night_time, evening, work_time
day	weekday, weekend

Table 2: The attributes of objects. Note that in the case of subject, $\mathtt{p}_1, \ldots, \mathtt{p}_n$ refer to the people in the user's network

Example 1 shows, for a small set of social indentities, the meaning of a simple conflict rule.

EXAMPLE 1. *Consider a simple social identity map with two social identities* $\mathtt{s}_1 = \{\mathtt{alice}, \mathtt{bob}, \mathtt{charlie}\}$ *and* $\mathtt{s}_2 = \{\mathtt{charlie}, \mathtt{david}\}$. *Also consider the single conflict rule:*

$$\mathtt{conflict\_si}(\mathtt{O}, \mathtt{s}_2, \mathtt{s}_1) \text{ :- } \mathtt{location}(\mathtt{O}, \mathtt{night\_club}).$$

[1]For the full semantics of ASP see [16].

The above rule states that any object with location night_club *causes a conflict if it is shared with a person in social identity* s_2 *and the subject of the object is someone in* s_1*. According to this rule, the object* ⟨alice, night_club, night_time, weekday⟩ *can be shared with* alice *and* bob*, but not with* charlie *or* david*.*

The ASP program that formalises our model is made of four components: SI, $Conf$, Obj and $Share$, representing the social identities, conflict rules, objects and a general protocol for sharing, respectively. SI is the program composed of facts of the form $\texttt{in\_si}(p_{id}, si_{id})$. Each of these facts states that person p, with unique identifier p_{id}, is in social identity si with unique identifier si_{id}. $Conf$ is the program containing the ASP conflict rules. For each object $\texttt{obj} = \langle s, l, t, d\rangle$, Obj contains the facts $\texttt{subject}(obj_{id}, s)$, $\texttt{location}(obj_{id}, l)$, $\texttt{time}(obj_{id}, t)$ and $\texttt{day}(obj_{id}, d)$ (where obj_{id} is a unique identifier for obj). $Share$ is a program containing the following two rules:

```
conflict(O, P) :-
  subject(O, P2),
  in_si(P, S1), in_si(P2, S2),
  conflict_si(O, S1, S2).

share(O, P) :-
  person(P),
  object(O),
  not conflict(O, P).
```

The first rule means that sharing an object O with a person P, who is in a social identity S1, could cause a conflict if the subject of O is in another social identity S2 which is in conflict with S1 for the object O. The second rule means all objects O are shared with all people P, unless sharing O with P could cause a conflict. Note that conflict rules do not contain the subject attribute, as the definition of conflict(O,P) depends on subject.

Given any social identity map, a set of conflicts and a set of objects, the corresponding program $SI \cup Conf \cup Obj \cup Share$ has exactly one answer set. This answer set contains the atom $\texttt{share}(obj_{id}, p_{id})$ for exactly those objects obj and people p such that obj could be shared with p given the known conflicts and social identity map.

So far, we have presented a method for determining a user's sharing given a social identity map, a set of objects and a set of known conflicts. In this paper, however, we begin with the conflict rules unknown. The task is to learn them from positive examples of objects being shared and negative examples (those objects which are not shared with certain people). Given a set of objects and a social identity map, the background knowledge B in our task is the program $SI \cup Obj \cup Share$. Note that we do not assume the social identity map represented by SI to be complete (as if it is given by a user, it is unlikely to be). The goal is then to learn a hypothesis containing conflict rules which explain as many of the (positive and negative) examples of sharing as possible (in the case where the social identity map is incomplete, it is possible that not all examples can be explained by the same hypothesis).

Hypotheses must be a subset of a search space S_M defined by a given *language bias*. This is a common feature of ILP systems and can be used to guide the search towards interesting hypotheses. In our case, the search space contains any rule for conflict_si whose body contains conditions on the attributes of a single object (for example, the single conflict rule in example 1).

For each object obj, we encode each of the people p who obj was shared with as a positive example $\texttt{share}(obj_{id}, p_{id})$; similarly, we encode the people who shouldn't be shared with as a negative example. The goal of our learning task is to find a hypothesis $H \subseteq S_M$ such that $B \cup H$ has at least one answer set[2] which contains every positive example and contains no negative example. In the ILP literature, this is known as *brave induction* [32].

In the case that not all examples can be explained by any hypothesis, we aim to find a hypothesis which maximises the number of examples which are explained. As is common in ILP, we compute the smallest such hypothesis (where the length of a hypothesis is calculated by counting the number of object attributes that appear in hypothesis). We call a hypothesis *optimal* if no hypothesis explains more examples, and no smaller hypothesis exists that explains the same number of examples.

Example 2 shows, for the small set of social identities in example 1, how a conflict rule can be learned from examples of sharing.

Example 2. *Recall the social identity map from example 1. Consider the object* o_1 = ⟨alice, night_club, night_time, weekday⟩ *and* o_2 = ⟨alice, office, day_time, weekday⟩ *such that* o_1 *was shared with* alice *and* bob *and* o_2 *was shared with everyone.*

The background knowledge in this case would be the program Share, augmented with the following facts:

$\texttt{in\_si}(\texttt{alice}, s_1)$. $\texttt{in\_si}(\texttt{bob}, s_2)$. $\texttt{in\_si}(\texttt{charlie}, s_2)$.
$\texttt{in\_si}(\texttt{charlie}, s_2)$. $\texttt{in\_si}(\texttt{david}, s_2)$.
$\texttt{subject}(o_1, \texttt{alice})$. $\texttt{location}(o_1, \texttt{night\_club})$.
$\texttt{time}(o_1, \texttt{night\_time})$. $\texttt{day}(o_1, \texttt{weekday})$.
$\texttt{subject}(o_2, \texttt{alice})$. $\texttt{location}(o_2, \texttt{office})$.
$\texttt{time}(o_1, \texttt{day\_time})$. $\texttt{day}(o_1, \texttt{weekday})$.

The positive and negative examples would be as follows:

$$E^+ = \left\{ \begin{array}{l} \texttt{share}(o_1, \texttt{alice}), \\ \texttt{share}(o_1, \texttt{bob}), \\ \texttt{share}(o_2, \texttt{alice}), \\ \texttt{share}(o_2, \texttt{bob}), \\ \texttt{share}(o_2, \texttt{charlie}), \\ \texttt{share}(o_2, \texttt{david}) \end{array} \right\} \quad E^- = \left\{ \begin{array}{l} \texttt{share}(o_1, \texttt{charlie}), \\ \texttt{share}(o_1, \texttt{david}) \end{array} \right\}$$

The positive examples each correspond to an object o *and person* p *such that* o *was shared with* p*. Similarly, negative examples each correspond to an object* o *and a person* p *such that* o *was not shared with* p*.*

In this case, one optimal solution would be the single conflict rule:

$\texttt{conflict\_si}(O, s_1, s_2)$:- $\texttt{location}(O, \texttt{night\_club})$.

To compute conflict rules we have made use of a learning approach based on the ASPAL algorithm for brave induction [7]. The learning task is translated into a meta-level ASP program whose optimal answer sets can be mapped back to the optimal inductive solutions of the given task. In our experiments, we use the answer set solver clingo [15] to solve the meta level representation.

[2]In fact, for any of our hypotheses H, $B \cup H$ has exactly one answer set.

5. EVALUATION

In this section, we evaluate our PI engine by testing how well it learns conflicts from synthetically generated examples of sharing. For each experiment, we randomly generated 100 social identity maps, each with a set of conflicts from our hypothesis space and used these to generate examples of sharing (described below). We then used the approach described in section 4 to learn a set of conflicts which explained the examples, given the (sometimes partial) social identity map and the shared objects' attributes as background knowledge, and evaluated the predictive performance of the learned conflicts against the true conflicts.

5.1 Generation of the Synthetic Data

Our synthetic data consists of three components: the social identity map, the conflicts and the examples.

We based the structure of our social identity maps on results from the literature. As the number of people in a typical user's social network, we used "Dunbar's number" which is the cognitive limit to the number of people with whom one can maintain stable social relationships [11]. In our experiments, we use the commonly used value, which is 150 [19]. On the other hand, the parameters we used to generate social identity groups are the number of social identity groups in a typical user's social network, social identity group size and finally pattern of a typical social network. The values we used for these parameters are the results of a survey conducted to social network users by McAuley and Leskovic [26]. In their study, the authors obtained network data of each user in order to construct user's corresponding social network graph and they asked each user to manually identify all the circles his/her friends belonged on his/her own social network graph. In this empirical study, on average users identified 19 circles in their social networks, the average size of a circle being 22. We observed that although users perceive circles as different social identity groups (e.g., school friends/university friends), the way they treat them in terms of sharing is fairly similar. Therefore, during our synthetic data generation, we set a range for the number of social identity groups, so that minimum value is 2 and the maximum value is 10. Taking the average number of friends in a social identity group as 22 and given that minimum group size can be 1, we set the range for a social identity group size to be $[1, 43]$. Regarding the typical pattern of social network, we also used findings of McAuley and Leskovic who found that approximately, 25% of groups are completely contained in another group, 50% overlap with another group and 25% have no members in common with any other group.

As the notion of sharing rules represented by conflicts is not studied in the literature, it is not possible to give a justification for how many conflicts a user is likely to have. We therefore tested our approach with sets of 10, 20 and 40 conflicts. The conflicts were restricted to those that contained 1 or 2 attributes in the body, as conflicts with 3 or 4 attributes in the body tend to have little effect (as they apply to fewer objects).

Once we had generated the social identity map and the conflicts, we could use this to decide, for each object, who should and shouldn't be shared with. We randomly selected a set of objects (from the full set of objects described by Table 2) and gave as positive and negative examples the people whom the objects should and should not be shared with, given the social identity map and conflicts.

5.2 Estimating the Learning Performance

In order to measure the performance of our approach, we tested with varying numbers of examples $(1, \ldots, 20)$ and using different numbers of synthetically generated conflicts $(10, 20, 40)$. In this section, we will refer to the synthetically generated conflicts, which were used to generate the examples, as H_{actual} and the conflicts which are learned by our system as $H_{learned}$.

		Learned Sharing Behavior	
		share	not share
Actual Sharing Behavior	share	TP	FN
	not share	FP	TN

Table 3: A ROC sheet for the assessment of learning performance.

Using the synthetically generated social identity map, we used the ASP approach defined in section 4 to compute the group of people that should be shared with for both H_{actual} and $H_{learned}$, over the full set of objects defined by Table 2. We then compared these two sets of "shares" to give a measure of the performance of the learned hypothesis $H_{learned}$.

In order to compare actual sharing behaviour with the learned shared behaviour, we use three different performance measures, which are *specificity*, *sensitivity* and *accuracy*. *Specificity* (i.e., true negative rate) measures the proportion of not share instances (given by H_{actual}) that are correctly identified as such (by $H_{learned}$). *Sensitivity* (i.e., true positive rate or recall) measures the proportion of positive shares instances that are correctly identified. Finally, *accuracy* measures the proportion of correctly classified both positive and negative share instances. Table 3 presents a ROC (Receiver Operating Characteristic) sheet, showing how the actual and learned sharing behaviours are mapped against each other to evaluate the performance of our inductive logic programming approach. Each column in the ROC sheet represents learned sharing behaviors, while each row represents the actual sharing behavior. Each cell {TP, FN, FP, TN} shows the number of examples that fall into each cell of this ROC sheet. TP, FN, FP and TN stand for true positives, false negatives, false positives and true negatives, respectively. The formulation of each performance measure is given in the following equations:

$$specificity = \frac{TN}{TN + FP} \tag{1}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

5.3 Results

We conducted three sets of experiments, each specifying the number of conflict rules in the H_{actual} set to be 10, 20 and 40. In each experiment set, we synthetically generated 100 social identity maps and sets of conflicts (SI and H_{actual}, respectively). We tested the performance of our learning approach with $0, \ldots, 20$ examples. As in reality the

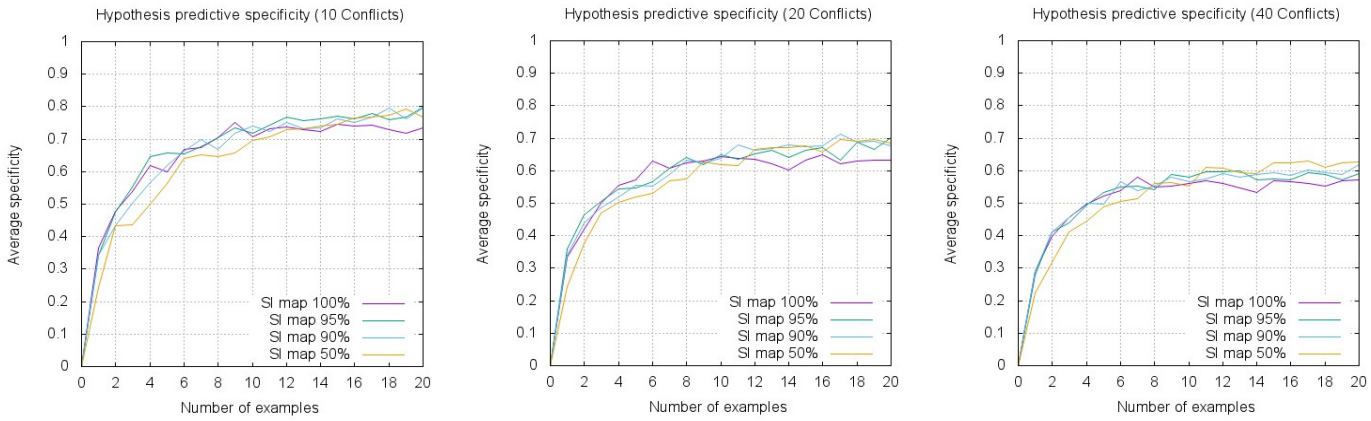

Figure 3: Specificity values for hypotheses generated with 10, 20 and 40 conflicts. Each point on the graph is the average value from 100 different synthetically generated SI maps and conflict sets.

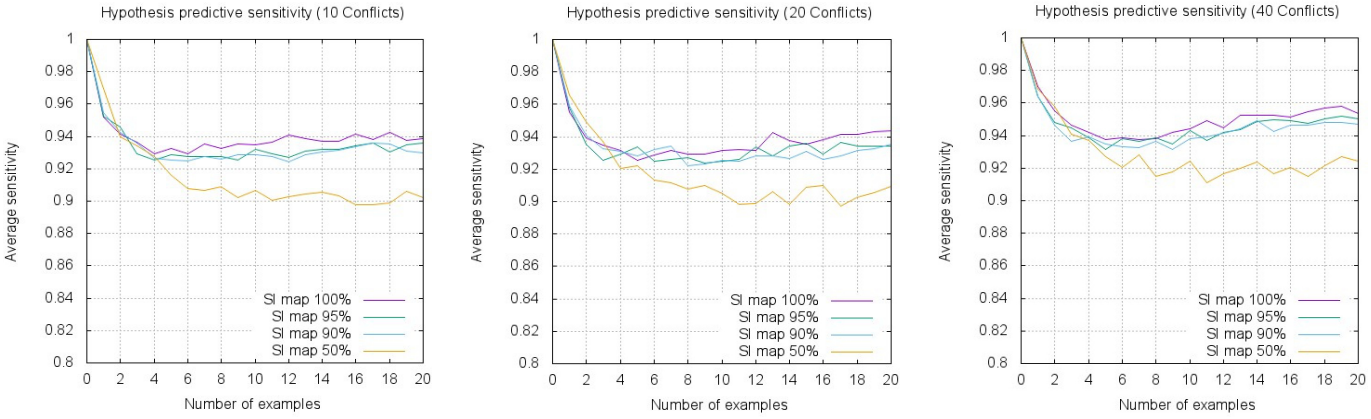

Figure 4: Sensitivity values for hypotheses generated with 10, 20 and 40 conflicts. Each point on the graph is the average value from 100 different synthetically generated SI maps and conflict sets.

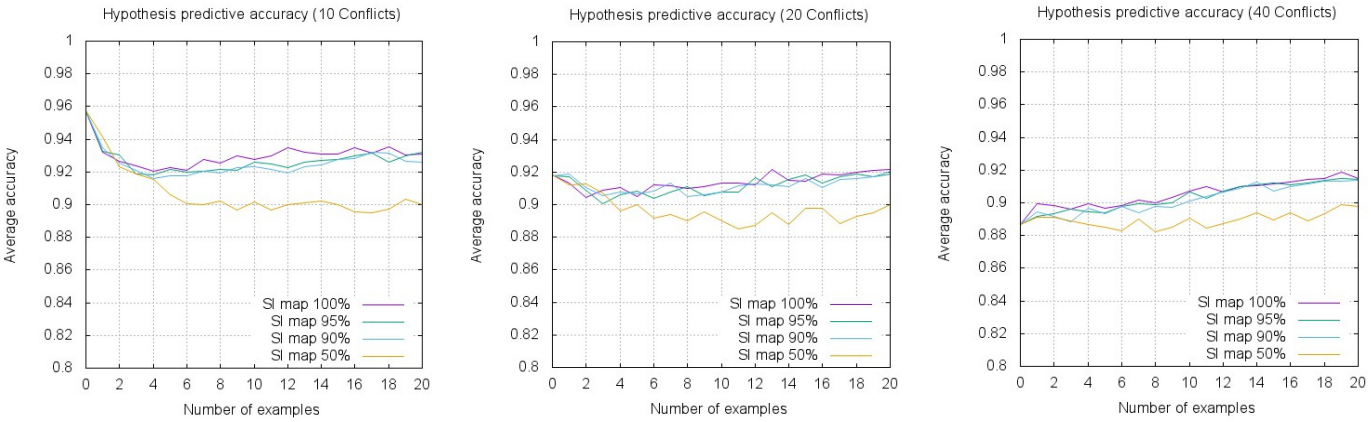

Figure 5: Accuracy values for hypotheses generated with 10, 20 and 40 conflicts. Each point on the graph is the average value from 100 different synthetically generated SI maps and conflict sets.

social identity map is unlikely to be fully known, we also tested the effect of giving a partial social identity map to the learner (where SI was 50%, 90%, 95% and 100% complete). The specificity, sensitivity and accuracy results for each experiment are shown in Figures 3, 4 and 5, respectively.

All three sets of experiments (with varying conflicts) show a similar pattern. For 0 examples, sensitivity is perfect and specificity is low - this is because our learned hypothesis will contain no conflicts, meaning that everyone is shared with, regardless of the attributes of the object; as the number of examples increases, the specificity increases rapidly at first and sensitivity drops slightly. The drop in sensitivity is caused by the learner being too conservative with the conflicts it learns, but sensitivity increases as the number of examples increases.

The final specificity values are better for fewer numbers of conflicts, as greater numbers of examples are needed to be observed learn the more complicated sets of conflicts. Interestingly, in every experiment, the specificity values are slightly higher for the less complete social identity maps. We believe this is because with an incomplete map, the learner can learn conflicts which are too conservative, because some of the people in the affected social identity who should be shared with are missing, and hence, unaffected (as far as the learner is concerned) by the learned conflict. This is also reflected by a significant drop in the sensitivity for experiments with the less complete social identity maps.

These results suggest that although groups that are explicitly defined by users on OSNs (e.g., Facebook's friend lists or Google+ circles) represent only a portion of users' actual SI maps, our learning system can still perform well at catching the instances of objects, which should *not* be shared. The idea is that the slight drop in detecting the instances, which should be shared should cause the user to revise their social identity map. This is safer than the alternative, where the conflicts are less conservative and the object is first shared with too many people.

5.4 Discussion

In this section, we discuss some of the limitations of our current approach, and how these might be addressed in the future.

Our current approach depends on the user providing us with an accurate social identity map. We have shown that our system still performs well with an incomplete map, but further experiments are needed to investigate how it performs with a noisy map (with some people incorrectly identified as being in a particular social identity). In future work we could also consider extending our learning approach to suggest revisions to the social identity map based on the sharing examples.

In our experimental setup, we also used a timeout of 5 minutes on the learner. The learner returned the best hypothesis that it found within the 5 minutes, but it is possible that a larger timeout could lead to a more accurate result. It was not feasible to run such large scale experiments with large timeouts, but in reality, our system would not be expected to run in real time as there is often a large gap between instances of sharing by the user. If we were to deploy the system in a real setting, we would therefore use a larger timeout. The number of Facebook friends for a given user will also have a bearing on the execution time of the learning process, but the impact of this can be minimised by preprocessing the sharing examples to treat groups of friends with whom identical objects are shared as a single individual.

Another assumption made in our experiments is that the user's sharing decisions were always consistent with his/her privacy norms, i.e. there is no noise in their sharing behaviour. Our learning approach would have been able to handle such noise, by learning hypotheses that explained an optimal number of examples. However, we did not include noise in our experimental data due to the difficulty of estimating a realistic amount of noise to include when generating our examples. Therefore, we plan to conduct experiments using real data in order to further test the performance of our current approach.

In the cases where some examples are not explained by the learned hypothesis, we can report the set of examples which have not been explained to the user to point out the inconsistencies in their sharing. The idea is that the user can then decide whether this was really their usual behaviour (and not just an exceptional case) and if so, modify their social identity map to explain the examples. Additionally, users can also update the knowledge to indicate which examples are irrelevant to help resolve inconsistencies and the learning will take this into account on next iteration.

6. RELATED WORK

In the literature, there are a number of privacy management solutions proposed for OSN platforms. Kafali et al. developed a tool for the run-time detection of privacy violations on online social networks [20]. The tool proposed by the authors employ model checking as a computational method taking as input the privacy agreements of the users, privacy properties, relations and content, which need to be *predefined*. Mester et al. employ a negotiation protocol for multi-agent systems in order to handle and resolve privacy violations that occur on OSNs [27]. The authors evaluate their solution, which aims multi-party privacy management, against realistic privacy violation scenarios on Facebook. In their proposed solution, privacy rules also need to be *predefined* using Semantic Web Rule Language (SWRL). Our solution also addresses multi-party conflict resolution; however, in our approach, privacy rules are learned over time by observing the user's sharing behaviour. Rules about the conflicting social identities facilitate automatic detection of privacy leakage and guides the user to ensure that items are not accidentally shared by unintended audience that are outside the groups of interest. Another solution proposed for multi-party privacy conflict resolution is by Such and Criado [33]. Here, the authors propose a computational mechanism that inspects individual privacy policies of all users who are subjects of the object to be shared, in order to figure out whether there are any contradictory access control decisions. Their proposed mechanism proposes a solution for each conflict detected based on the sensitivity of the shared object for each subject by taking into account each subject's willingness for negotiation. This is based on a quantitative negotiation model, in contrast to our logic based model. Although our experiments used one subject per shared object, our model also supports multiple subjects for a shared object and this is possible by addition of multiple subject facts for the object.

Barth et al. propose a logical framework for expressing and reasoning about norms of transmission of personal in-

formation [5]. Their work is based on Helen Nissenbaum's "contextual theory", which is a privacy theory conceptualising the appropriateness of information sharing based on the contexts in which the information is shared [31]. This work has inspired the development of quantitative computational model of "contextual integrity", where agents can infer users' privacy norms from their sharing behaviour [9]. The authors model the "context" in the form of a group of users, which have similarities with our definition of social identity groups. One important aspect of their model is that the agents can infer privacy norms in the presence of malicious attackers. Our logic-based computational model also learns users' privacy norms through observing their sharing behaviour and as future work, we plan to incorporate noise in user's sharing behaviour to simulate malicious attackers in our model.

There are also solutions proposed in order to decrease the user burden on privacy management [8, 29, 35]. Wang et al. propose privacy nudges in order to prevent users from making online disclosures that they will later regret [35]. In their study they focus on two types of nudges, one that reminds users about the audience in their post and another that encourages users to use and think before posting an item on social media. However, there is not an intelligent mechanism at the background producing these nudges and the proposed solution does not have any adaptive features. Our system, being logic based, has the capability to provide informative recommendations to the users, while guiding them in their sharing decisions and as well as in the modification and creation of their defined OSN groups. This is also an advantage of our approach over Criado and Such's computational model of contextual integrity [9]. The other techniques proposed by Mugan et al. and Crenshaw et al. employ machine learning techniques to enhance adjustment of complex privacy preferences. Mugan et al. employ statistical machine learning techniques to decrease the complexity of privacy settings for location sharing applications by generating user-understandable privacy option [29]. Crenshaw et al. use a classifier based on multi-variate Gaussian mixtures in order to represent users' location sharing preferences [8]. The solution proposed by these authors can learn privacy policies in a user-controllable setting. However, these models have only been demonstrated with location sharing applications, whereas our approach is designed to be more general.

Among other statistical machine learning based systems is the adaptive information sharing system developed by Bilogrevic et al [6] for mobile social networks. Similar to our approach, Bilogrevic et al. also assume that users are privacy-aware (i.e., their sharing decisions always comply with their privacy norms). Another study, which assumes that all users are privacy-aware is by Fang and LaFevre [13]. In their work, the authors design a template of a Facebook privacy wizard, which asks some users to label their friends who were randomly selected as "share" and "not share". The wizard uses these labels to construct a classifier that can be used to assign similar privacy privileges to users' unlabelled friends who are in the same communities. The authors employ existing techniques to extract communities from users' social network data. Fang and LaFevre claim that users in same communities share similar privacy preferences. This claim is in line with our claim that people in similar social identity groups have common privacy norms. However, we need to empirically investigate whether social identity groups overlap with communities in users' social network data.

7. CONCLUSION AND FUTURE WORK

In this paper we presented a Privacy Dynamics architecture inspired by social identity theory. This is based on a formal model with two key concepts: firstly, group membership information, represented as social identity maps; and secondly, privacy norms, represented as a set of conflicts. We have shown that we can use ILP to learn a user's privacy norms through examples of their sharing behaviour. The key benefit of using ILP, rather than a statistical machine learning approach, is that our output is human readable, and can therefore be explained to the user.

We have evaluated our approach with synthetic examples of sharing, and shown that we are able to learn sharing norms which can accurately predict the people with whom an object can be shared, even with very few examples and an incomplete social identity map.

Although in this paper we have made the assumption that the user's sharing behavior perfectly reflects their sharing norms (i.e. their behaviour is not noisy), this may not be the case in reality. Therefore, in future work we intend to evaluate the approach with examples of real users' sharing behaviours.

8. ACKNOWLEDGMENTS

This research is partially supported by the EPSRC grants EP/K033522/1, EP/K033425/1, EP/K033522/1 (Privacy Dynamics), as well the ERC Advanced Grant - Adaptive Security and Privacy (291652 - ASAP), SFI grant 10/CE/I1855 and SFI grant 3/RC/2094.

9. REFERENCES

[1] Facebook User Statistics. http://newsroom.fb.com/company-info/, 2014 (accessed December 1, 2015).

[2] Leading Social Networks Worldwide as of November 2015, Ranked by Number of Active Users (in millions). http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/, 2014 (accessed December 1, 2015).

[3] D. B. Alice E. Marwick. I tweet honestly, i tweet passionately: Twitter users, context collapse and the imagined audience. *New Media and Society*, 13(1):114–133, 2010.

[4] D. Athakravi, D. Corapi, K. Broda, and A. Russo. Learning through hypothesis refinement using answer set programming. In *Inductive Logic Programming*, pages 31–46. Springer, 2014.

[5] A. Barth, A. Datta, J. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: framework and applications. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15 pp.–198, May 2006.

[6] I. Bilogrevic, K. Huguenin, B. Agir, M. Jadliwala, and J.-P. Hubaux. Adaptive information-sharing for privacy-aware mobile social networks. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 657–666, New York, NY, USA, 2013. ACM.

[7] D. Corapi, A. Russo, and E. Lupu. Inductive logic programming in answer set programming. In *Inductive Logic Programming*, pages 91–97. Springer, 2012.

[8] J. Cranshaw, J. Mugan, and N. Sadeh.

User-controllable learning of location privacy policies with gaussian mixture models. 2011.

[9] N. Criado and J. Such. Implicit contextual integrity in online social networks. *Information Sciences*, 325:48–69, 12 2015.

[10] M. Duggan, N. B. Ellison, C. Lampe, A. Lenhart, and M. Madden. Social Media Update 2014. http://www.pewinternet.org/2015/01/09/social-media-update-2014/, 2015 (accessed December 1, 2015).

[11] R. I. M. Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, June 1992.

[12] R. Emerson. 13 Controversial Facebook Firings: Palace Guards, Doctors, Teachers and More. http://bit.ly/Controversial-FB-Firings, 2015 (accessed December 1, 2015).

[13] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 351–360, New York, NY, USA, 2010. ACM.

[14] G. A. Fowler. When the Most Personal Secrets Get Outed on Facebook. http://www.wsj.com/articles/SB10000872396390444165804578008740578200224, 2012 (accessed December 1, 2015).

[15] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 20131.

[16] M. Gebser, R. Kaminsky, B. Kaufmann, and T. Schaub. *answer Set Solving in Practice (Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan and Claypool Publishers, 2012.

[17] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, pages 1070–1080, 1988.

[18] J. T. Henri Tajfel. *Intergroup relations: Essential readings. Key readings in social psychology* ., chapter An Integrative Theory of Intergroup Conflict. Psychology Press„ New York, NY, US, 2001.

[19] A. Hernando, D. Viluendas, C. Vesperinas, M. Abad, and A. Plastron. Unravelling the size distribution of social groups with information theory in complex networks. *The European Physical Journal B*, 76(1):87–97, 2010.

[20] O. Kafali, A. Gunay, and P. Yolum. Protoss: A run time tool for detecting privacy violations in online social networks. In *Advances in Social Networks Analysis and Mining (ASONAM),2012 IEEE/ACM International Conference on*, pages 429–433, Aug 2012.

[21] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, Jan 2003.

[22] A. Lampinen, S. Tamminen, and A. Oulasvirta. All my people right here, right now: Management of group co-presence on a social networking site. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work*, GROUP '09, pages 281–290, New York, NY, USA, 2009. ACM.

[23] M. Law, A. Russo, and K. Broda. Inductive learning of answer set programs. In *Logics in Artificial Intelligence (JELIA 2014)*. Springer, 2014.

[24] E. Litt. Knock, knock. who's there? the imagined audience. *Journal of Broadcasting and Electronic Media*, 56(3):330–345, 2012.

[25] A. Luft. Quebec Woman Loses Benefits over Facebook Photo. http://www.thestar.com/news/canada/2009/11/23quebec_woman_loses_benefits_over_facebook_photo.html, 2015 (accessed December 1, 2015).

[26] J. Mcauley and J. Leskovec. Discovering social circles in ego networks. *ACM Trans. Knowl. Discov. Data*, 8(1):4:1–4:28, Feb. 2014.

[27] Y. Mester, N. Kokciyan, and P. Yolum. Negotiating privacy constraints in online social networks. In *Advances in Social Computing and Multiagent System, Second International Workshop on Multiagent Foundations of Social Computing*, pages 112–128, 2015.

[28] D. Mosbergen. Single Mom Fired from Daycare Center for Facebook Post Saying She Hates being around a Lot of Kids. http://www.huffingtonpost.com/2015/05/05/daycare-worker-fired-facebook-kaitlyn-walls_n_7210122.html, 2015 (accessed December 1, 2015).

[29] J. Mugan, T. Sharma, and N. Sadeh. *Understandable learning of privacy preferences through default personas and suggestions*. Carnegie Mellon University, School of Computer Science Technical Report CMU-ISR-11-112, http://reports-archive. adm. cs. cmu. edu/anon/isr2011/CMU-ISR-11-112. pdf, 2011.

[30] S. Muggleton. Inductive logic programming. *New generation computing*, 8(4):295–318, 1991.

[31] H. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 22(6):101–139, 2004.

[32] C. Sakama and K. Inoue. Brave induction: a logical framework for learning from incomplete information. *Machine Learning*, 76(1):3–35, 2009.

[33] J. M. Such and N. Criado. Adaptive conflict resolution mechanism for multi-party privacy management in social media. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, pages 69–72, New York, NY, USA, 2014. ACM.

[34] P. Walker. Octavia Nasr fired by CNN over Tweet Praising Late Ayatollah. http://www.theguardian.com/media/2010/jul/08/octavia-nasr-cnn-tweet-fired, 2015 (accessed December 1, 2015).

[35] Y. Wang, P. G. Leon, X. Chen, S. Komanduri, G. Norcie, A. Acquisti, L. F. Cranor, and N. Sadeh. From facebook regrets to facebook privacy nudges. *Ohio State Law Journal*, 74:1307–1335, 2013.