Collaborative Privacy Policy Authoring in a Social Networking Context

Ryan Wishart, Domenico Corapi, Srdjan Marinovic, Morris Sloman
Department of Computing,
Imperial College London,
London, U.K.
Email: {r.wishart, d.corapi, srdjan, m.sloman}@imperial.ac.uk

***Abstract*—Recent years have seen a significant increase in the popularity of social networking services. These online services enable users to construct groups of contacts, referred to as friends, with which they can share digital content and communicate. This sharing is actively encouraged by the social networking services, with users' privacy often seen as a secondary concern. In this paper we first propose a privacy-aware social networking service and then introduce a collaborative approach to authoring privacy policies for the service. In addressing user privacy, our approach takes into account the needs of all parties affected by the disclosure of information and digital content.**

I. Introduction

Online social networking sites have enjoyed explosive growth recently with Facebook[1], the most popular site, claiming membership in excess of 400 million users[2]. Facebook, and social networking sites in general, enable users to define groups of friends (other users of the social networking service) with which they can share information and digital content such as photographs and videos.

As the business model for many social networking sites depends on access to large quantities of user data, users are actively encouraged to upload and share content and information, often with strangers. This has led to numerous incidents where personal data available on a social networking site has been exploited for uses other than it was originally intended. In several recent cases, people have lost their jobs due to the posting of inappropriate comments and photographs on Facebook [1].

To safeguard users' privacy online, privacy protection mechanisms are needed. These mechanisms limit the scope for attacks on user privacy by ensuring that accesses to a user's data are authorized by the user. Existing privacy protection mechanisms enable users to specify policies for their data. They do not support cases where disclosure of data affects several individuals. An example of this is with the sharing of a photograph of a group of people. Viewers of the photograph are able to learn information which any one of the individuals depicted in the photograph may not have wanted disclosed (for example the individual's location, activity and who they were with at a particular time).

In this paper we present a solution to this problem that employs a collaborative method for specifying privacy policies. In our approach, the *owner* of content is responsible for uploading it to the social networking site. This owner can then specify privacy policy for the content. This policy includes the activation conditions for the policy, the resource it applies to and also the permissions it grants. The owner can then nominate a number of friends on the social networking site to modify the policy activation conditions i.e. change the scope of the policy. In the remainder of this paper we refer to these individuals as co-owners of the content. We assume that co-owners are people that will be affected by the disclosure of the content.

The remainder of this paper is structured as follows. Related work in the field is presented in Section II. A model for a privacy-preserving social network is then presented in Section III. This is followed by a description of our privacy policy language in Section IV. We outline a motivational scenario for the work in Section V before discussing our prototype implementation in Section VI. A discussion of future work is then given in Section VII before the paper is concluded in Section VIII.

II. Related Work

In this section we present an overview of existing work in the field of privacy policy. We begin with an overview of industry approaches, then discuss research work in the field of social networking and finish with a summary.

A. EPAL

EPAL (the Enterprise Privacy Awareness Language) [2] is a privacy policy language developed by IBM. As the name suggests, it is primarily intended for use by enterprise-level organizations that deal with customer data. The policy language enables these organizations to specify access and usage restrictions on the customer data they have collected (like the time of day a request is made), the purpose to which the data will be put and any obligations that come with access to the data (e.g., accessed data must be deleted after a set time period).

[1] www.facebook.com

[2] http://www.facebook.com/press/info.php?statistics

B. XACML

XACML, the eXtensible Access Control Markup Language [3], is an XML-based general access control policy language. While not explicitly intended for privacy policy specification, the policy language is sufficiently general to be put to this purpose. Each XACML policy applies to a *target* (which may be subjects, resources, environments or system actions), contains a set of rules to determine when the policy is applicable, an algorithm for evaluating these rules and a list of obligations associated with accessing the target. As with EPAL, rules can be conditional (e.g., dependent on time of day, location etc).

C. P3P

The Platform for Privacy Preferences [4], (commonly referred to as P3P), is a privacy policy language developed by the W3C to address the privacy needs of web users. Websites that collect user data describe how the data is collected, what it will be used for, the retention policy of the website, and whether gathered data will be disclosed to third-parties. Once defined, the website's P3P policy is made publicly available.

Web users define their own data usage policies, using the APPEL [5] preference language, stating the conditions under which they are willing to disclose their data. When a user visits a website with a P3P policy, a P3P agent built into the user's web browser compares the website's P3P policy to the user's APPEL policy. The results are displayed to the user, who then must decide whether or not to release data to the website.

D. Or Best Offer (OBO)

The OBO approach was developed by Walker *et al.* [6] to automate privacy policy negotiation. Their work is intended to establish a privacy policy acceptable to both the client (user) and the data collecting service.

In the approach, each type of user data (e.g., telephone numbers) receives three tags describing: the recipients; the purposes that the data can be used for (such as marketing); and the server retention policy. For each of these three tags, the user defines 'ideal', 'acceptable' and 'unacceptable' values. The server similarly defines its privacy policy, but in terms of the data it wishes to collect.

During the negotiation phase, the user (represented by a negotiation client) and the server attempt to arrive at a privacy policy in which the recipient, purpose and retention tags for all user data types have values that are at least 'acceptable' to both parties.

E. Collaborative Access Control

Carminati *et al.* [7] developed an access control approach based on the relationships in the social network between the requester and the content/resource owner. Their work categorizes all relationships between users in the social network according to: type, referring to whether the relationship is direct, or indirect (e.g., parent or friend of a friend); the trust given to that relationship; and, the distance between the requester and the owner in the social network graph. This graph is a commonly used representation of a social network in which each user appears as a node and friendship relationships are represented as links between nodes.

Access policies associated with the owner's resource are specified using the 3 properties of relationships (type, trust and depth) described previously. Users of the network are given signed credentials by the owner certifying their depth, relationship type and trust level. To access a resource, the requesting party must demonstrate it holds the necessary relationship credentials from the resource owner.

F. Lockr

The developers of Lockr, Tootoonchian *et al.* [8], use relationships within the social network to specify privacy policy. In their approach, users store their data inside a Lockr server. Access to the data is based on authenticated access to the Lockr with authentication achieved using public key certificates. Permissions within the Lockr are associated with public keys. To access data, the requester must be in possession of a public key with the necessary permissions. Alternatively, access to data can be gained through attestations - signed statements from trusted parties proclaiming that the requester of the data is in a certain kind of relationship (e.g., parent) with the owner of the data in the Lockr.

G. Collective Privacy Management

Squicciarini *et al.* [9] developed a collaborative privacy policy approach specifically for social networks. In their work, a user specifies privacy policies for her data (e.g., videos and photographs) in terms of the maximum depth in the social network graph that the viewer can be. For instance, a maximum depth of 1 would equate to a direct friend, while a depth of 2 would mean a friend of a friend.

The approach also makes use of the concept of shared ownership of data. This is achieved by having the originator of the data, that is the user responsible for uploading the data, specify other potential owners of that data. The system then holds an auction on the possible privacy policy to apply to the data in which all the owners submit a vote for their desired policy. The system is designed to reward data originators that include other owners, and owners that correctly choose the winning privacy policy. To reduce the frequency of auctions, the approach includes a learning mechanism that applies privacy policy to data that are similar (based on a comparison of tags on the data).

H. Summary

In online social networking environments, where multiple parties can be affected by disclosure of content, privacy protection mechanisms are needed to enforce the privacy of all parties involved. Policy-based approaches offer the greatest flexibility in these environments but, to the best of our knowledge, no existing approach is able to establish policy for a shared resource that respects the privacy requirements of all users affected by the disclosure of the resource.

Tootoonchian *et al.* attempt to address the problem of access control within social networks. However, they focus on policy specification from the perspective of a single resource owner.

A simple approach to supporting multi-owner policy specification would be to merge the individual policies of the resource owners pertaining to the resource to decide if a disclosure should be permitted. This would then enable existing policy languages, such as EPAL and P3P/APPEL, to express the privacy policy. XACML, while not specifically intended for privacy policy, is sufficiently general that it can also be put to this purpose. However, merging policies leaves no room for negotiation [6]. The need for negotiation is supported by Spiekermann *et al.* [10] who found that user's privacy requirements are not fixed, and can be downgraded in different situations or to achieve a desired outcome.

In the Collaborative Access Control approach developed by Squicciarini *et al.* co-owners individually propose privacy policy for their shared resource and then vote on their preferred policy. The approach acknowledges the need for authors to collectively decide on the policy for the resource, but offers limited capacity for negotiation.

The concept of policy negotiation is further explored in OBO, developed by Walker *et al.*, which enables two parties to generate a privacy policy that is mutually acceptable. In a social networking scenario, where a resource potentially has multiple owners, performing such a complex, multi-stage negotiation is likely to be infeasible.

III. Social Network Model

In this paper we have not restricted our analysis of the privacy issues regarding data dissemination to a specific social networking platform. We have rather opted to specify a set of features that need to be present in a social network for our approach to be applicable. These features are:

1) Users are able to add/remove their content on the service. Here content refers to video, photographs, status updates, comments and links.
2) Ownership of content is allocated to the user that uploads it to the service.
3) Users can specify a privacy policy for each of the content items they own.
4) Users can add/remove friends on the service.
5) Users that are friends can view one another's content as well as modify it (by adding tags, comments and links).
6) Owners can undo/remove modifications (such as comments) added by friends to their content.

It is interesting to observe that Facebook meets all these requirements. Therefore, to apply our policy approach to specifying privacy concerns, Facebook would need to implement our policy evaluation mechanism and provide a user interface to author the policies.

IV. Privacy Policy Language

In this section we present a simple privacy policy language that we use as a vehicle for explaining our collaborative policy authoring approach. As our contributions lie in the method by which policies are authored, other languages (such as XACML) could theoretically have been used given suitable extensions.

In our approach, privacy policies are specified as logic rules that define a set of *permitted* actions on a *resource* for a given *request*. The policy language is used for brevity and should not be considered a contribution of the paper.

Policies are created by an *owner* (responsible for the resource) who manages the policies along with a set of *trusted* co-owners. In the remainder of this paper we shall refer to the owner and the set of co-owners collectively as *owners*.

Within the policy language, policy conditions are expressed as first order predicates and can refer to properties of the request, the resource and the owners. All conditions are specified as either *weak* or *strong*. They are semantically equivalent when used in the policy evaluation, but they differ in the way that they can be authored by the owners.

Definition 1. *A policy p is defined as a tuple* (r, A, SC, WC) *where* $r \in \mathcal{R}$ *identifies the resource the policy refers to;* $A \in \mathcal{A}$ *is a set of permitted actions on the resource* r*;* SC *is a set of strong conditions and* WC *is a set of weak conditions both expressed as logic literals.*

Definition 2. *A weak or strong condition is a tuple* (u, c) *where* $u \in \mathcal{O}$ *is a user identifier and is called the* author *of the condition and* c *is a logic literal.*

The semantics of the policy authoring are clarified in Sec. IV-B.

Definition 3. *The ownership function* o *is defined as* $o : \mathcal{R} \rightarrow 2^{\mathcal{O}}$ *and associates a resource to a set of owners.*

For the sake of readability, we show policy $p = (r, \{a_1, ...a_m\}, \{sc_1, ..., sc_n\}, \{wc_1, ..., wc_p\})$ in the following syntactical representation:

```
policy p {
    strong-conditions:
         sc1, ..., scn
     weak-conditions:
         wc1, ..., wcp
    resource: r
    can-do: a1, ..., am
  }
```

A. Policy Evaluation

In our approach, all requests to access a resource are passed to the Policy Decision Point (PDP). This PDP evaluates the policy for the resource using a knowledge base. This knowledge base contains properties of the request as well as information about the social network (such as user groups).

In this paper, we use the Datalog language and its semantics [11] to specify the policy evaluation semantics for a request. The use of stratified Datalog semantics enables us to introduce negation in the policy body.

Datalog has been extensively investigated and used in the field of access control policies [12], [13] and trust management [14], [15]. The important property that the policy evaluation gains is the tractable decidability of the request evaluation. A consequence of this is that the policy language cannot use functions.

For the sake of completeness of the presentation, we recall basic notions of Datalog. The reader is referred to [16] for more detailed coverage.

A Datalog program P consists of a set of relations R and a set of rules Q. A relation $r \in R$ is a set of n-ary tuples with $n > 1$. A *ground (positive) literal* $r(k_1, ..., k_n)$ is true iff the tuple $(k_1, ..., k_n)$ is in the relation r. A ground *(negative) literal* $\neg r(k_1, ..., k_n)$ is true iff the tuple $(k_1, ..., k_n)$ is not in the relation r, which implements the Closed World Assumption (CWA) implied by the stratified semantics. A *(non-ground) literal* can contain variables.

CWA is not usually adopted in trust management languages [14], [15] due to the problem of credential gathering. Since our approach follows the *typical* access control model where all information needed to make a decision is contained in the system (in our case the social network information) we find it appropriate to adopt the CWA.

Rules are of the type $h \leftarrow b_1, ..., b_n$ that intuitively means that h is *true* if the literals $b_1, ..., b_n$ are true. h is called the *head* of the rule and is a positive literal, while $b_1, ..., b_n$, are the *body* of the rule. These body terms are positive or negative literals or relations over natural numbers (such as $=, >$).

All variables appearing in the rule are universally quantified with scope the entire definition and are denoted as upper case letters. With abuse of notation we use Q to denote the conjunction of the logic rules in Q. Similarly, R denotes the conjunction of the relations in R as logic atomic (with empty body) rules. A *query* g is true iff it is true in the iterated fixed point Herbrand model [17] of $Q \wedge R$, denoted in this paper as $Q \cup R \models g$.

We translate policies into rules that together with a (possibly empty) set of domain rules B are included in Q. B can be used to derive views from the relations. For example, rules in B can define useful abstractions like the definition of active user (a user that posted at least one comment) or the definition of common friend. The definition of rules in B and the control and update of R are under the authority of the social network management. E includes all the relations relevant to evaluate the policy.

Given a set of policies S we derive a set of rules Q_S that contains for each policy $(r, \{a_1, ...a_m\}, \{sc_1, ..., sc_n\}, \{wc_1, ..., wc_p\})$ a set of rules:

$$
\begin{aligned}
&p'_1 \leftarrow sc'_1 \wedge ... \wedge sc'_n \wedge wc'_1 \wedge ... \wedge wc'_p \\
&... \\
&p'_m \leftarrow sc'_1 \wedge ... \wedge sc'_n \wedge wc'_1 \wedge ... \wedge wc'_p
\end{aligned}
$$

Atoms in the head correspond to permissions and are defined as follows $p'_i = perm(a_i, X)$, where X is a logic variable that refers to requests and that does not appear in the rest of the rule. Conditions refer to predicates defined in R or B. A subset of these predicates REQ has as first argument the request variable X. Each weak or strong condition $c_i(k_1, ...k_n)$ is transformed into the logic condition $c'_i = c_i(X, k_1, ...k_n)$ if $c_i \in REQ$; otherwise $c'_i = c_i(k_1, ...k_n)$.

Definition 4. *Given a set of policies S, a set of domain rules B and a set of relations R, an action t is permitted for a request q if and only if $Q_S \cup B \cup R \models perm(t, q)$. The action is not permitted otherwise. Denied decision is semantically equivalent to a not permitted decision.*

As an example of how policies are transformed into rules and then evaluated, consider policy p below. The policy states that the album ‘Birthday party’ can be viewed by users that are (1) in the ‘Family’ group for Alice and (2) are not “friends in touch” with Bob (friends that have received at least one message from Bob).

```
policy p{

   strong-conditions:
      request_by(Y),
      group(‘Alice’, Y, ‘Family’)

   weak-conditions:
      not friend_in_touch(‘Bob’, ‘Y’)

   resource: ‘Birthday party’

   can-do: view
  }
```

As this is the only policy, $S = \{p\}$. At some later point in time, Eva, Alice’s aunt, issues a request to view the Birthday party album. Eva is close friends with Bob and regularly sends messages to him. In this case R includes:

$$
\begin{aligned}
&\mathit{request\_by}(r, \text{‘}Eva').\\
&group(\text{‘}Alice', \text{‘}Eva', \text{‘}Family').\\
&friend(\text{‘}Bob', \text{‘}Eva').\\
&\mathit{message\_sent}(\text{‘}Bob', \text{‘}Eva', m).
\end{aligned}
$$

B includes the following rule:

$$
\begin{aligned}
&\mathit{friend\_in\_touch}(X, Y) \leftarrow\\
&friend(X, Y), \mathit{message\_sent}(X, Y, M).
\end{aligned}
$$

One of the predicate symbols in the policy refers implicitly to the request, i.e. $\mathit{request\_by} \in REQ$. Q_S is defined as:

$$
\begin{aligned}
&perm(view, X) \leftarrow\\
&\mathit{request\_by}(X, Y),\\
&group(\text{‘}Alice', Y, \text{‘}Family'),\\
&\neg \mathit{friend\_in\_touch}(\text{‘}Bob', Y),
\end{aligned}
$$

Thus, the request r’s action $view$ is not permitted as the condition $\mathit{friend\_in\_touch}(\text{‘}Bob', \text{‘}Eva')$ is true where the policy requires it to be false. This means $Q_S \cup B \cup R \models \neg perm(view, r)$.

Note that the weak condition in the example only restricts the set of users allowed to view the resource. This is a general feature of the model.

B. Policy Authoring

The authoring of policies addresses two main requirements: owners have to be able to collaborate on the definition of the policy; and, each owner of the resource has to be able to arbitrarily restrict the permissions on the resource.

The policy authoring mechanism is regulated by the following rules. Given a policy $p = (r, P, SC, WC)$:

1) *A user u can add a weak or strong condition to p iff $u \in o(R)$*
2) *A user u can delete a weak condition iff $u \in o(R)$*
3) *A user u can delete a strong condition (u', c) iff $u' = u$*

Owners can freely collaborate on weak conditions but can also express non-negotiable restrictions on the resource using strong conditions. A straightforward extension would permit two different kinds of ownership: one for users entitled to instantiate strong conditions and another for users only able to use weak conditions.

C. Condition Conflict Resolution and Malicious Owners

When collaboratively writing a policy, owners may specify conflicting conditions for the policy. Other work within our group [18][19] has focussed on policy conflict analysis and we will use this to detect policy conflicts. We assume that the conditions are regularly evaluated during the authoring process to detect such conflicts. Once detected, the authoring process is halted and all owners involved notified.

If the conflict is due to a co-owner that placed overly restrictive conditions over content, the other co-owners should either respect that co-owner's wishes or modify the content so that the co-owner is no longer affected e.g., for a photograph one could blur the co-owner's face or crop them from the picture.

Alternatively, the conflict may be caused by a malicious co-owner purposely sabotaging the policy authoring with unreasonable conditions. In this case, we assume such behaviour can be detected by the resource owner and other co-owners. This will require support from the policy authoring tool. Once notified of this malicious behaviour, the owner and co-owners can then vote to exclude the malicious co-owner from the policy authoring process. The owner can then restart the policy authoring protocol and not invite the malicious co-owner to participate. In following this approach, we assume that (1) a co-owner's reasonable concerns for her privacy will not be interpreted as malicious and (2) the majority of co-owners are not themselves malicious.

V. Scenario

In this section of the paper we present two scenarios. The first is a motivational scenario highlighting the deficiencies of existing privacy controls in online social networks. In the second scenario we introduce our approach and demonstrate how it can be used to overcome the problems identified in the first scenario.

A. Motivational Scenario

In the present arrangement used by online social networking services, users upload digital content which is then available to other users of the social networking service. Typically, social networking sites offer only limited restrictions on the viewers of the content. For example, in Facebook, a user can restrict viewership to (in order of increasing generality) a sub group of their friends, all their friends or everyone on the social networking service.

Currently it is possibly for a person, Alice, to take photographs of people at a party. She can then upload the photographs to a social networking service like Facebook, making them available to everyone. In the process, Alice violates the privacy of all the people appearing in the photographs. For this example, we assume that Bob is one of the people in the photographs. He shares many friends with Alice, and finds the photographs particularly embarrassing.

B. Scenario with Collaborative Policy Authoring

The application of our technology allows Alice, once she has uploaded her photographs, to specify privacy policy for the photographs - in the process, giving her fine-grained control over who can access the photograph album. She can also create conditions on access that use context information within the knowledge base of the PDP, such as a user's location or time of a request.

Alice specifies the policy below that states only viewers in her 'Friends' group can view, tag and comment on the 'Party Album' containing the photographs in question.

```
policy policy1{

   strong-conditions:
      request_by(Y),
      group('Alice', Y, 'Friends')

   weak-conditions:

   resource: 'Party Album'

   can-do: view, comment, tag
  }
```

Alice then nominates Bob and Carol as co-owners of the album as they appear in most of the photographs. On flicking through the album, Bob notices several embarrassing pictures. He adds conditions to the policy to prevent his family members viewing the album. He also prevents two of his friends, Errol and Filipo, viewing the album (but only uses weak conditions as he would prefer them not to see the album, but is willing to concede access if the other owners think it socially difficult to block them e.g., if Carol already told them about the party). As an extra precaution he prevents access to the album when the requester is inside his house (where his family are likely to see the photos).

```
 policy policy1{

    strong-conditions:
      request_by(Y),
      group(‘Alice’, Y, ‘Friends’),
      not group(‘Bob’, Y, ‘Family’),
      request_time(T),
      not located_at(Y, ‘Bob’s House’, T)

    weak-conditions:
      not request_by(’Errol’),
      not request_by(’Filipo’)

    resource: ‘Party Album’

    can-do: view, comment, tag
   }
```

The ‘request_time’ predicate in the policy above is needed to separate out other instances where the requester may have visited Bob’s house.

Carol edits the policy. She definitely does not want Errol or Filipo to see the album and so upgrades the corresponding weak conditions to strong conditions.

```
 policy policy1{

   strong-conditions:
      request_by(Y),
      group(‘Alice’, Y, ‘Friends’),
      not group(’Bob’, Y, ‘Family’),
      request_time(T),
      not located_at(Y, ‘Bob’s House’, T)
      not request_by(’Errol’),
      not request_by(’Filipo’)

   weak-conditions:

   resource: ‘Party Album’

   can-do: view, comment, tag
    }
```

This new policy is enforced by the social networking service on all accesses to Alice’s ‘Party Album’.

It should be noted that, while our example is simplistic, our approach lends itself to the creation of arbitrarily complex policies. These policies can also refer to domain rules, (such as the friend_in_touch rule used in Section IV-A), as well as information within the knowledge base used by the PDP, such as time and requester location context information.

The contribution of the work is that the policy was collaboratively authored by several people for the purpose of controlling the disclosure of content on the social network.

VI. Implementation

To demonstrate the efficacy of the approach we developed a privacy-aware content sharing application for the Facebook social networking service, referred to as *PRiMMA-Viewer*. Our prototype application enables users to nominate other users of PRiMMA-Viewer that they consider to be friends. Users can upload digital content (currently restricted to photographs or photograph albums) as well as tag and comment on content belonging to friends.

Uploaded content is not stored on the Facebook server, but rather on the PRiMMA-Viewer server. Using a server external to Facebook with our own PDP and PEP enables support for the potentially complex policies indicated in the paper. In comparison, Facebook only offers comparatively simple access control policies. Additionally, use of an external server circumvents concerns that service providers, such as Facebook, will use stored content for commercial purposes in the future and not adequately enforce user-specified access controls (e.g., [20]).

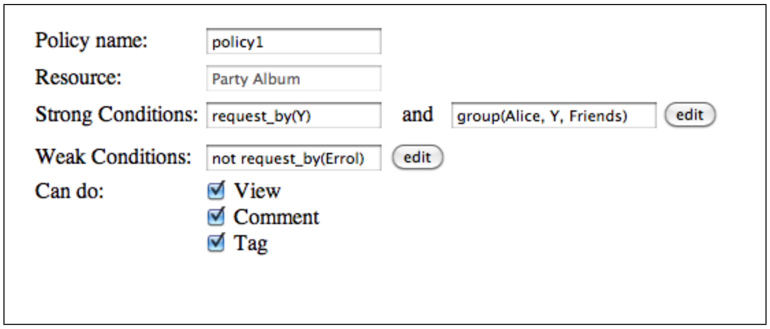
As part of the uploading procedure, the uploader of the content (referred to as the *owner*) can specify an initial privacy policy governing access to the content by other users. This policy controls which users learn about the new content and the conditions under which users are able to view, comment on and tag the content.

A screenshot of the application is shown in Figure 1 depicting the central editing window in which this policy can be written. In our prototype implementation, owners are required to hand-code their policy. We are currently developing a user-friendly GUI implementation to simplify the policy creation and editing process.

As part of our previously outlined collaborative authoring approach, the owner can also nominate friends to act as co-owners of the newly uploaded content. These friends are typically other people the originator believes will be affected by the disclosure of the content. In the case of photographs, co-owners could be considered as the other people appearing in the photograph. These co-owners can edit the policy by adding or removing conditions.

The PRiMMA-Viewer application is implemented as follows. It uses the iFrame external application approach in which Facebook acts as an intermediary between the user and the application server. In this role, Facebook accepts input from the user, forwards it to the application server and displays to the user (possibly sanitized) webpages served up by the application server. These respective steps are shown as 1, 2, 7 and 8 in Figure 2, which depicts the operation and architecture of PRiMMA-Viewer.

Within the PRiMMA-Viewer application server, incoming user requests to view, tag or comment on content are forwarded to the Policy Decision Point (PDP). The PDP evaluates these requests against policies defined for the affected resources (i.e. photographs or albums) and determines whether access should be permitted or denied. The PDP is currently implemented



Policy name: policy1

Resource: Party Album

Strong Conditions: request_by(Y) and group(Alice, Y, Friends)

Weak Conditions: not request_by(Errol)

Can do:
☑ View
☑ Comment
☑ Tag

Fig. 1. Screenshot of the PRiMMA-Viewer application showing the policy authoring screen.

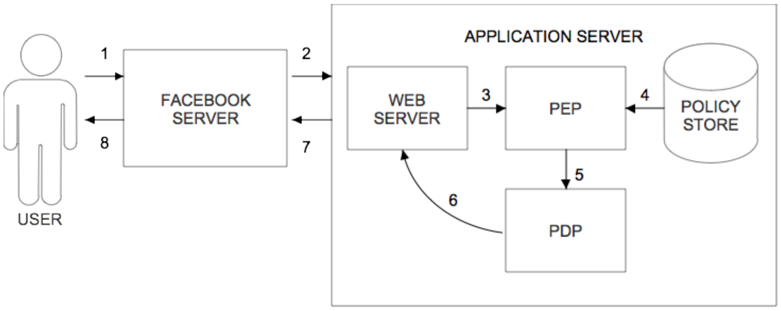


Fig. 2. Architecture of the PRiMMA-Viewer application.

using the Java language IRIS Reasoner[3]. IRIS was chosen as it is open source and supports stratified Datalog reasoning.

The decision of the PDP is then passed to the Policy Enforcement Point (PEP) where the decision is enforced (e.g., access is denied to photo album ”Carol’s Party Album”). The web server component generates the appropriate webpage based on what the PEP permits and returns this to Facebook to display to the user.

VII. Future Work

The collaborative policy authoring approach we present in this paper is dependent on the uploader of the content nominating co-owners with which to author the policy. Currently our approach assumes this occurs without some external incentive mechanism. We are examining the use of an incentivised voting scheme similar to that used by Squicciarini *et al.* [9] to overcome this problem.

Related to this issue is the inability of a user to claim co-ownership of a resource. We do not provide a mechanism for co-owners to assert ownership over a resource without an invite as it is difficult to validate such claims. Supporting this functionality for any type of content is an area of ongoing research. However, it may be possible with photographs to use facial recognition technologies to validate a claim of co-ownership.

An additional area for future work is the development of a user-friendly policy authoring tool. Currently, our PRiMMA-Viewer Facebook application provides limited help with authoring policies, requiring users understand the policy language. We are currently developing a newer version of the tool to make the collaborative policy authoring process more user-friendly and accessible to average users of social networks. This new version will also include the most recent work from our research group on policy conflict analysis techniques.

At present we have not done any user testing of the approach to determine user acceptance. We are planning to incorporate such testing in a larger user trial of our privacy-aware social networking platform to be performed near the completion of our research project.

VIII. Conclusion

In this paper we presented a novel approach to collaborative policy authoring demonstrated within the context of social networking. Our approach permits the originators of content on the social network to specify policies for the content they upload. The conditions under which the policy applies can then be edited by nominated users of the social networking service. These parties represent people interested in the dissemination of the content. Limitations are enforced on the policy editing so that the scope of the policy can only be decreased by the nominated parties.

The approach was implemented in a prototype Facebook application to demonstrate its efficacy. The application permitted the uploading of content (limited to photographs in the prototype), specification of policy on that content and then the shared editing of the policy. Once the policy was finalized, the application enforced the policy.

[3] http://iris-reasoner.org/

Acknowledgment

This work was funded by EPSRC Grant EP/F024037/1 as part of the PRiMMA: Privacy Rights Management for Mobile Applications project as well as by EU FP7 research grant 213339 (ALLOW).

References

[1] R. Stross. (2007, December) How to lose your job on your own time. online. The New York Times. [Online]. Available: http://www.nytimes.com/2007/12/30/business/30digi.html?ex=1356670800&en=55ef6410d3cac28e&ei=5088&partner=rssnyt&emc=rss

[2] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. (2003, November) Enterprise privacy authorization language (epal 1.2). online. IBM. [Online]. Available: http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/

[3] OASIS. (2005, February) eXtensible Access Control Markup Language (XACML) version 2.0. [Online]. Available: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#XACML20

[4] L. Cranor, M. Langheinrich, and M. Marchiori. (2002, Jan) The platform for privacy preferences 1.0 (P3P1. 0) specification. [Online]. Available: http://www.w3.org/P3P/

[5] ——. (2002, April) A P3P Preference Exchange Language 1.0 (APPEL 1.0). online. W3C. [Online]. Available: http://www.w3.org/TR/2002/WD-P3P-preferences-20020415/

[6] D. D. Walker, E. G. Mercer, and K. E. Seamons, "Or Best Offer: A Privacy Policy Negotiation Protocol," in *Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks (POLICY 2008)*, 2008, pp. 173–180.

[7] B. Carminati, E. Ferrari, and A. Perego, "Rule-Based Access Control for Social Networks," in *Proceedings of the OTM workshops*, ser. Lecture Notes in Computer Science, vol. 4278. Springer, Jan 2006, pp. 1734–1744.

[8] A. Tootoonchian, K. Gollu, S. Saroiu, Y. Ganjali, and A. Wolman, "Lockr: Social Access Control for Web 2.0," *Proceedings of the First ACM SIGCOMM Workshop on Online Social Networks (WOSN 2008)*, pp. 43–48, 2008.

[9] A. Squicciarini, M. Shehab, and F. Paci, "Collective Privacy Management in Social Networks," in *Proceedings of the 18th International Conference on World Wide Web (WWW2009)*, 2009, pp. 521–530.

[10] S. Spiekermann, J. Grossklags, and B. Berendt, "E-privacy in 2nd generation E-commerce: privacy preferences versus actual behavior," in *Proceedings of the 3rd ACM conference on Electronic Commerce (EC-2001)*. ACM, 2001, pp. 38–47.

[11] J. D. Ullman, *Principles of Database and Knowledge-Base Systems, Volume 2*. Computer Science Press, 1989.

[12] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino, "A Unified Framework for Enforcing Multiple Access Control Policies," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '97)*, 1997, pp. 474–485.

[13] A. Chaudhuri, P. Naldurg, S. K. Rajamani, G. Ramalingam, and L. Velaga, "EON: Modeling and Analyzing Dynamic Access Control Systems with Logic Programs," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS'08)*, 2008, pp. 381–390.

[14] J. DeTreville, "Binder, a Logic-Based Security Language," in *Proceedings of the IEEE Symposium on Security and Privacy (SP'02)*. IEEE Computer Society, 2002, p. 105.

[15] N. Li, J. Mitchell, and W. Winsborough, "Design of a Role-Based Trust-Management Framework," in *Proceedings of the IEEE Symposium on Security and Privacy (SP'02)*, 2002, pp. 114–130.

[16] S. Ceri, G. Gottlob, and L. Tanca, "What you always wanted to know about Datalog (and never dared to ask)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, no. 1, pp. 146–166, March 1989.

[17] K. R. Apt, H. A. Blair, and A. Walker, "Towards a theory of declarative knowledge," in *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, 1988, pp. 89–148.

[18] R. Craven, J. Lobo, E. Lupu, J. Ma, A. Russo, A. Bandara, S. Calo, and M. Sloman, "Expressive policy analysis with laws of system change," in *Proceedings of the Annual Conference of ITA (ACITA2009)*, September 2009.

[19] A. Bandara, E. Lupu, and A. Russo, "Using event calculus to formalise policy specification and analysis," in *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, 2003.

[20] C. Arthur. (2010, April) Facebook privacy hole 'lets you see where strangers plan to go'. [Online]. Available: http://www.guardian.co.uk/technology/2010/apr/26/facebook-privacy-hole