

# Usable Privacy Controls for Blogs

Michael Hart  
Stony Brook University  
mhart@cs.sunysb.edu

Claude Castille  
Stony Brook University  
claude.castille@gmail.com

Rob Johnson  
Stony Brook University  
rob@cs.sunysb.edu

Amanda Stent  
Stony Brook University  
stent@cs.sunysb.edu

**Abstract**—Web 2.0 applications, including blogs, wikis and social networking sites, pose challenging privacy issues. Many users are unaware that search engines index personal information from these sites and offer public access to collected data. As a consequence, privacy invasions are rampant. In this paper, we demonstrate that *tag-based* privacy policies are a usable and flexible privacy control method for Web 2.0 applications. Content owners express their privacy policy in terms of tags on content objects; the system then applies the policy to objects based on the tags assigned to them. We implemented tag-based privacy controls as a plugin to the WordPress blogging system, and conducted a user study to measure whether users could efficiently implement real-world privacy concerns with tag-based policies. Despite limited time, training and familiarity with the blogging system, a third of the participants chose the tag-based policy tools. These users were able to perform privacy-related tasks with the same accuracy and increased speed using our tag-based privacy controls versus per-object privacy controls.

**Keywords**—Security; Privacy; Web 2.0; User-Interface Design; Blogs; Tags;

## I. INTRODUCTION

The popularity of Web 2.0 applications (including social networking, content sharing, and blogging sites) has exploded, resulting in more personal information and opinions being available with fewer privacy controls than ever before [7]. Users clearly want control over the private information they publish on Web 2.0 applications [8], [9]. Most sites, however, provide only the most rudimentary of privacy controls: an object can be completely private or completely public. Other sites use a variety of custom, incompatible privacy controls that are hard for users to understand and maintain. Consequently, even users aware of the privacy implications of the Web 2.0 often default to making all their content public, while others try to protect their privacy by maintaining multiple accounts [7], increasing their workload substantially.

The consequences of poor privacy controls are well-documented in the news media. Users of social networking sites and bloggers have lost employment op-

portunities [5] and been subject to disciplinary procedures [1], [13]. Others have become victims of sexual predators [10] or stalkers [12].

Ideally, a privacy control system for the Web 2.0 will address several needs. The privacy policy representation must be flexible and scalable, able to operate over loosely structured and dynamic data rather than rigid data types and categories. It must also be rich enough to capture the many relationships between content owners and viewers, and to express a wide range of privacy preferences. The policy authoring and control tools must adapt as users restructure their social networks, and be efficient and simple enough for users with varying levels of technical knowledge and speaking different languages.

In this paper we present Plog, a *tag-based* privacy control tool. Tags are words or phrases assigned to pieces of content by a content owner. An owner can create tags at different levels of granularity, and create tag rules assigning to each tag or tag combination a different level of privacy. Users of Web 2.0 applications are already familiar with using tags [6], so minimal effort is required to maintain a privacy policy. Tag-based privacy control is more manageable than per-object access control. Also, tag rules correlate well with users' privacy preferences (see Section III-A). Finally, since tags are supported by almost every Web 2.0 application, tag-based privacy policies are easy to transfer across sites.

We have implemented Plog as a plugin to the WordPress blogging system [17] and conducted a user study to compare the usability of tag-based privacy policies and per-object privacy policies. Despite little precedence in the Web 2.0 for rule-based policies, our evaluation results show that participants who wrote tag-based policies completed tasks as accurately and significantly faster than those who set per-post policies. In addition, participants generated tag-based policies of nearly optimal size.

## II. PRIVACY SCHEMES IN BLOGS AND SOCIAL NETWORKS

We surveyed the privacy features provided by seventeen blogging and social-networking sites. Sites in our

survey permitted content owners to assign access rights to viewers in the following ways:

**Private/Public.** Content owners assign each object to be private (viewable only by the owner) or public.

**Friends.** Content owners create a list of *friends* and designate objects as visible only to viewers in this list.

**Other Users.** Content owners restrict posts to be visible only to other registered users of the site.

**Shared Attributes.** Content owners grant access to viewers that share attributes such as belonging to the same school, city, or workplace.

**Search Engines.** Content owners indicate objects should not be indexed by search engines.

**Password-protected Posts.** Content owners assign passwords to protect objects.

**Profile Information.** Content owners restrict access to certain portions of their profile, such as “status updates”.

The “friends” model, employed by MySpace and others, is the most prevalent of the existing schemes, but it is too coarse to capture the user’s privacy requirements. For example, few friends-based privacy control systems lets user distinguish real-world friends, who presumably already know the user’s home address and other personal information, from online friends, who may be close but do not need to know the user’s real life details. Friends-based privacy controls also force users to choose between protecting their privacy and being social. Many users equate popularity with having a large number of friends. For example, we found that MySpace users had a median of 115 friends<sup>1</sup>. Few people have 115 close confidants. As the notion of “friend” loses its meaning, friends-based privacy control also becomes meaningless.

These models make an implicit trade-off between simplicity and flexibility. For example, friends-based privacy controls attempt to make it easier for the user to specify *who* can see private information by offering only three options: everyone, no-one, or all the user’s friends. Profile-based controls try to simplify the task of identifying *what* information (e.g. photos, status, contact information, etc.) is private. These rigid policy models may be easy to use, but they cannot meet the needs of most users. Rigid models may choose an inappropriate level of granularity (e.g. friends-based privacy controls) or may group objects or subjects in a way that does not correspond with the user’s privacy preferences (e.g. with profile-based controls, users cannot assign different policies to work photos and school photos).

Although much research has focused on aspects of access control such as policy languages, policy enforce-

ment, policy reasoning engines, and distributed access control systems, little has addressed the policy authoring problem. IBM’s SPARCLE project [3] developed several tools to help experts translate human-readable corporate privacy policies into machine-readable form. Our user interface for specifying privacy preferences is similar to SPARCLE’s structured policy authoring interface.

Systems that, like Plog, use the content and categorization of an object to render access control decisions exist in specialized capacities. Adult content filters are a special instance of a topic-based access control system [11]. Most notable in this category is the “MaX” system of the EUFORBIA project [2], which uses content meta information to enforce an attribute-based access control. Other researchers have investigated techniques for classifying objectionable images based on their content to prohibit accepting their submission [15].

Users need a way to specify the privacy of objects that is both flexible and integrated into normal Web 2.0 activities. Tag-based privacy tools use the multiple levels of granularity of tags to create flexible privacy policies without incurring the management costs of complex schemes, such as password-protected posts. In the next section, we formally define tag-based policies.

### III. THE PLOG POLICY LANGUAGE

Tags are words or phrases that are paired with objects (e.g. blog posts, photos, videos). Content owners author their own tags, so tag usage can be quite varied. Tags often indicate content topic, enumerating topics both broad (e.g. “golf”) and narrow (e.g. “Tiger Woods”). Tags can also describe content properties such as form (e.g. “photo”), type (e.g. “conference paper”), location (e.g. “Sri Lanka”), or origin (e.g. “from my window”). Finally, tags may be used humorously or in other idiosyncratic ways. Because tags often correspond to topics, they are a promising basis for a privacy policy language. However, not all tags are privacy-relevant, and content owners could apply tags inconsistently. Consequently, owners should be able to override tag-based privacy rules for specific objects.

To formalize the Plog policy language, let  $O$  be the set of objects belonging to a content owner. In a blog, these objects would correspond to posts; in a social networking site, these objects could correspond to blog posts, photos, personal details, videos, etc. For each object  $o \in O$ , let  $T_o$  be the set of tags the user has assigned to that object. Let  $\mathcal{V}$  be the set of all potential viewers, and let  $v_0 \notin \mathcal{V}$  be a special “anonymous” viewer. Viewers may be identified by email address, user ID, or some other method – the details are irrelevant to the policy

<sup>1</sup>Based on a random sample of 91660 MySpace users logged-in between September 1st, 2006 and October 23, 2006.

language. Visitors must authenticate to access restricted content. Unauthenticated viewers have the access rights of the anonymous viewer.

The user can specify a set of tag privacy rules  $R \subseteq T \times 2^{\mathcal{V}}$ , where an entry  $(t, V)$  indicates that viewers in set  $V$  can see objects assigned tag  $t$ . Given the tags,  $T_o$ , on an object,  $o$ , the set of applicable tag rules is  $R'_o = \{(t, V) | t \in T_o\}$ . In Plog, the content owner may disable some of these rules on a per-object basis, so access control decisions are actually made based on the owner-specified set  $R_o \subseteq R'_o$ . By default,  $R_o = R'_o$ , so owners only need to explicitly specify  $R_o$  in exceptional situations. The owner may also manually grant or deny access to a particular object. For each object, the user may specify a set  $A_o \subseteq \mathcal{V}$  of allowed viewers, and a set  $D_o \subseteq \mathcal{V}$  of denied viewers. By default,  $A_o = D_o = \emptyset$ .

The set of viewers allowed to see object  $o$  is

$$V_o = \begin{cases} \bigcup_{(t, V) \in R_o} V \cup A_o \setminus D_o & \text{if } R_o \cup A_o \cup D_o \neq \emptyset \\ \mathcal{V} \cup \{v_0\} & \text{otherwise} \end{cases}$$

Because this algorithm takes the union over all tag rules, it grants access to any viewer who is a member of  $V$  for some tag rule  $(t, V) \in R_o$ . An alternative implementation could choose to take intersections in this case, i.e. access would be granted only to viewers who are members of  $(t, V)$  for all tag rules  $(t, V) \in R_o$ . We chose union because, in our system, the user can disable tag rules he does not want applied to a particular object.

When no policies apply to an object ( $R_o = A_o = D_o = \emptyset$ ) then the object is world-readable. For blogs and social networks this seems more appropriate than a default deny policy. This also ensures that if  $v_0$  can view an object, then so can all other viewers. It does not make sense to grant access to anonymous viewers but deny access to an authenticated viewer, because the denied viewer could simply request the object anonymously.

This privacy policy language is very flexible. It also makes policy management simple in the common case. Once the content owner has created a few tag rules, the system will apply the privacy policy to new objects as they are created. Only in a few exceptional cases will the owner wish to override the default policy.

Rules in this language are evaluated at time-of-access. This makes it easier to understand and reason about the policy language and for the content owner to retroactively adjust the policy on existing objects.

In our implementation, content owners can also define groups of viewers and use these groups in tag rule authoring. The name of the group is replaced with the list of its members for the purpose of policy evaluation.

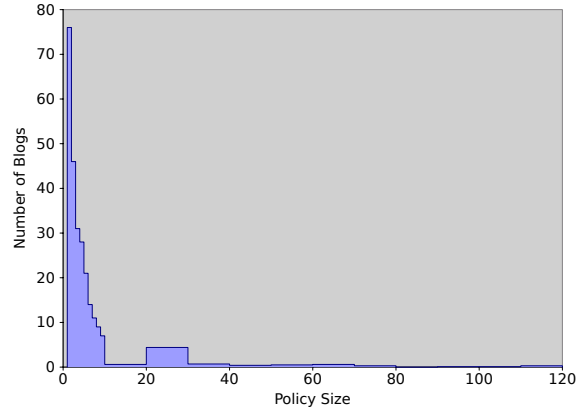


Fig. 1. Histogram of optimal policy sizes for our WordPress blog dataset.

### A. Tag Usage on Existing Private Posts

A good privacy policy language should enable users to express their preferences with a few simple rules. We evaluated the Plog policy language by translating privacy settings on existing blogs into our language and measuring the size of the resulting policies. We used a screen-scraper to download 377 blogs with private posts from the WordPress blogging service. WordPress does not hide the existence of private posts or the tags assigned to them. Thus, we can use this publicly available information to construct a Plog policy that maintains the same public/private dichotomy.

For each blog, our screen-scraper generated a set  $\{(p_i, t_i)\}_{i=1}^n$ , where  $p_i$  is the privacy setting of the  $i$ th post (either “public” or “private”), and  $t_i$  is the set of tags on that post. We implemented a policy solver that constructs from the output a set of tag rules and per-post exceptions that preserves the public/private separation and minimizes the total number of tag rules and exceptions. Constructing such a policy is equivalent to the NP-complete weighted hitting-set problem, so our solver uses a brute-force branch-and-bound algorithm.

Figure 1 shows a histogram of the policy sizes generated by our solver. Over half the bloggers could express their privacy policy in just 4 rules using Plog, and 75% could express their privacy policy using only 9 rules. Our solver found, for example, the following tag-based policies for two different blogs: *private, relationships* and *parenting, tragedy, love, family*. A small percentage of bloggers had huge tag-based privacy policies. These bloggers use tags so haphazardly that tag-based rules offer them no benefit, but they are no worse off, either.

This analysis has several limitations that may cause it to overestimate or underestimate the size of policies in a

Use this dialog to create a tag rule. A tag rule allows only the groups and individuals you specify to read a post tagged with the tags of the tag rule.

**Only my friends**  
(separated by commas)  **can read posts tagged**  
(separated by commas)

**Tag(s) apply to these posts**

|                               |                                     |                                        |
|-------------------------------|-------------------------------------|----------------------------------------|
| A Swell(ing) Journey          | <input type="button" value="View"/> | <input type="button" value="Exclude"/> |
| A Swell(ing) Journey Part III | <input type="button" value="View"/> | <input type="button" value="Exclude"/> |

**But don't apply to these posts**

|                              |                                     |                                        |
|------------------------------|-------------------------------------|----------------------------------------|
| A Swell(ing) Journey Part II | <input type="button" value="View"/> | <input type="button" value="Include"/> |
|------------------------------|-------------------------------------|----------------------------------------|

(a)

**Publish Status**  
Unpublished

**Who can see**  
Only the following friends can read: Joe Gilver, Joanna Gilver, Jack Gilver, James Gilver, Justina Gilver, family, Tom, Jane

Draft Saved at 7:07:53 pm.  
Word count: 0

This widget shows you who can read this blog post

Only the following people  Joe Gilver  Joanna Gilver  Jack Gilver  James Gilver  Justina Gilver  family  Tom  Jane can read.

**Tag rules ( a checked box means the rule is applied )**

family Joe Gilver,Joanna Gilver,Jack Gilver,James Gilver,Justina Gilver and groups family can read posts with this tag

friends Tom,Jane can read posts with this tag

(b)

Fig. 2. The Plog interfaces for (a) tag rule authoring, and (b) creating per-post rules.

deployed Plog system. First, we discarded private posts that had no associated tags. Since Wordpress reveals the tags on private posts, users may leave these posts untagged to protect their privacy. We discarded untagged posts to avoid this source of bias. However, leaving out all untagged private posts can cause us to underestimate the Plog policy size, since we don't need to create per-post rules for those posts. Second, we may have underestimated Plog policy sizes because the solver generated policies that only distinguished public and private posts, whereas the actual policies may have granted different viewers access to different posts. On the other hand, we may overestimate Plog policy sizes because bloggers using a Plog-enabled system would be encouraged to tag their content consistently, which would result in smaller Plog policies and a more organized blog.

Despite the limitations of this analysis, the results suggest that tag-based privacy controls would dramatically reduce the effort required to maintain a blog privacy policy. Since users already tag private posts, they could benefit from tag-based privacy policies immediately with many users able to express their privacy preferences in less than 5 tag rules.

#### IV. PLOG WORDPRESS PLUGIN

We have implemented Plog as a plugin to the Wordpress blogging system. The plugin consists of three main components: tag rule authoring, per-post policy editing, and user and group management.

**Tag rule authoring interface:** Content owners author tag rules by filling in templates, as shown in Figure 2(a). While the owner is authoring or editing a tag rule, the interface provides a list of the posts that would be affected by the rule. This helps owners write accurate rules and gives them an opportunity to add exceptions along with the new rule. If the owner does not recall the content of an object, she can view it in a pop-up window. The tag rule authoring interface is similar to the SPARCLE “structured authoring page”, although our implementation uses text fields with auto-completion instead of check-boxes or natural language [3].

**Per-post policy editing interface:** A content owner may create policy exceptions for an object while editing the object by using the policy editing interface as shown in Figure 2(b). The Plog plugin displays the list of viewers who can see the current object next to the “Publish” button. This encourages owners to consider

the appropriate privacy policy for each object before publishing it, and gives the owner a chance to double-check the policy inference engine’s results. If the owner wishes to make an exception to the policy for this object, she can click on the “Edit” link next to the viewer list. This produces the dialog in the center of Figure 2(b), with which the owner can set one of three states for the object. The “Everyone” state gives access to anyone, including anonymous viewers. The “Friends” state gives access only to viewers who have at least one credential proving they are friends with the owner. The owner can exclude specific friends (for example, if the object is an invitation to a surprise birthday party for a friend). Lastly, the “No-one but” state permits the owner to restrict access to a list of authenticated viewers.

We replaced the default access control system (password protection) in Wordpress, where the post is either public or requires a password to read. We implemented a plugin that allows readers to authenticate with their online identities (see the following subsection) that eliminates password management and better facilitates the audit and update of access control decisions.

**User and group management:** Content owners can reference viewers who possess identities outside of Plog. Currently, viewers can identify themselves with their OpenID, GMail, AOL, MSN and Facebook accounts using their respective authentication APIs. These services account for a significant percentage of internet users [4]. When a viewer visits a Plog-enabled blog, the page displays a link so the viewer can identify himself. Since blog authors may identify a viewer by different identities (e.g. his Facebook and GMail accounts), the viewer will have to authenticate for each ID, but Plog will infer that the same viewer owns these accounts. In the future, the viewer only needs to login with one of his IDs.

## V. EVALUATION

We performed a user study to (1) test whether users could leverage tag rules to enforce privacy constraints, and (2) test whether users naturally preferred tag rules over per-post policies. The experiment and results are discussed in the following sections.

**Participants:** Twenty-eight undergraduate and graduate students were recruited for this study (Male= 18, Female= 10, Mean Age= 21.17). Participants were at least 18 years old, fluent English speakers, and registered students of our university. None were Wordpress users.

**Experimental Design:** Each participant completed five training tasks and thirteen experimental tasks in which they had to role play as Ted, a blogger. The experimental phase included both privacy tasks and “distractor” tasks

that covered normal blogging activities. The distractor tasks provided a baseline for comparing performance between participants.

In order to lend realism to our tasks, we used a real blog released under the Creative Commons license, which we anonymized by replacing person and place names. This blog contained posts written by a married couple with compelling life stories. Topics discussed included struggles with kidney disease, the death of a father, and photos of family, which motivate genuine privacy concerns. So that our participants could complete the experiment in a reasonable amount of time, we shortened each blog post to a maximum length of 400 words and reduced the total number of posts to thirty. We did not modify the tags on any posts. We also created 15 individuals that constituted Ted’s social network. The individuals belonged to four groups: family, church friends, work and college buddies.

We used two deceptions to reduce participant bias. First, the participants were told that they were evaluating the Wordpress open-source blogging system for learnability and usability. Since the participants were led to believe that we had not designed any part of the Wordpress system, they had less incentive to withhold criticism, try to “get the right answer”, or praise the system. Second, the distractor tasks obscured the privacy focus of our study. This reduced bias between participant performance on privacy versus distractor tasks.

**Training:** Participants in our study were not Wordpress users, so each participant first completed five training tasks as an introduction to the Wordpress system. Each participant read the entire blog (thirty posts), wrote a blog post, managed posts, used the per-post privacy control tool, and used the tag rule privacy control tool. The system randomized the order of the two privacy training tasks for each participant. This allowed us to detect whether participants were biased towards the privacy control presented to them first.

To be confident in participants’ understanding of the software, we evaluated participant performance in the training phase. All participants successfully completed most training tasks. Nine participants did not successfully complete the training task that involved setting a per-post privacy policy. Five of these participants failed to specify any privacy policy for this task, but were able to successfully use the per-post privacy policy tool in the first experimental privacy task. The other four were excluded from our experimental analyses.

**Tasks:** Each participant completed thirteen experimental tasks. In the distractor tasks, the participant performed activities unrelated to privacy control, such as modifying

| Task | Privacy Concern            | Posts | Total tags | Minimal Tag-based Policy          | Optimal Policy   |
|------|----------------------------|-------|------------|-----------------------------------|------------------|
| 10   | Church community posts     | 2     | 47         | One tag rule and one exception    | Tag-based policy |
| 11   | Memorial to father         | 1     | 18         | One tag rule                      | Per-post policy  |
| 13   | Work related posts         | 2     | 28         | One tag rule                      | Tag-based policy |
| 14   | Daughter sees author naked | 1     | 17         | One tag rule                      | Per-post policy  |
| 15   | Health problems            | 6     | 21         | One tag rule                      | Tag-based policy |
| 18   | Humor and opinions         | 4     | 48         | Two tag rules                     | Tag-based policy |
| 21   | Family photos              | 5     | 35         | Five tag rules and two exceptions | Per-post policy  |

Fig. 3. Privacy tasks and their optimal solutions

the blog design, commenting on a post, managing blog posts, and reading other blogs on the site. In the privacy tasks, the participant was asked to restrict readership of topically related posts to a group of viewers (e.g. work friends, college friends). Each privacy task expressed privacy preferences as generally and plainly as possible to model realistic privacy concerns, and did not suggest a solution strategy. Participants were allowed to solve the tasks using either tag rules, per-post specifications or a combination of both.

For privacy tasks, the level of difficulty in authoring tag rules varied greatly due to the usage of tags on the relevant posts. Tag rules were not always the optimal strategy because of instances where the tag rules plus exceptions exceeded the number of relevant posts. (see Figure 3).

**Procedure:** The user study took place in computer teaching labs over the course of a week. Participants were asked to give two hours of their time and were compensated with \$20. Participants were allowed to ask questions. All participant interface actions were logged. Following the user study, each participant completed a fifteen-item questionnaire rating the effectiveness, learnability and ease of use of the major functions of the blogging system on a seven point Likert scale.

**Tag rule usage:** Eight of the twenty four participants (the *tag-rule* group) used tag rules to solve five of the seven experimental privacy tasks. The remaining sixteen participants (the *per-post* group) used the per-post tool to complete all of the experimental privacy tasks. Four participants in the tag-rule group and four participants in the per-post group completed the tag rule training task before the per-post training task. The other sixteen participants saw these tasks in the opposite order. We conclude that the participants' strategy was not dependent on the order of training tasks for the system's privacy features ( $p = 0.36$ , Fisher's Exact Test).

**Accuracy on privacy tasks:** Each post is associated to a single privacy task with a set  $V_o$  of intended viewers. We scored accuracy on privacy tasks as  $1 - \frac{|U|}{|V|}$  where  $U$  includes viewers to whom the participant granted

access but are not in  $V_o$  and viewers in  $V_o$  to whom the participant did not grant access. If a participant left a post public, her accuracy for that post is zero. For each participant, we divided posts into a tag-rule set and a per-post set, depending on how the participant attempted to solve the task. We then computed overall accuracy for each method by averaging, across all participants and tasks, the accuracy on posts in each set. These overall accuracy scores were compared using an independent two-tailed Welch's *t*-test. For all posts related to a privacy task, the accuracy for tag-rule posts ( $M = 0.82$ ,  $SD = 0.12$ ) was nearly identical to the accuracy for per-post posts ( $M = 0.82$ ,  $SD = 0.13$ ;  $t(350.10) = 1.97$ ,  $p > 0.97$ )<sup>2</sup>.

We also computed per-task accuracy for each method in the same way, but only considering the posts relevant to each task. Accuracy scores were compared using two-tailed Welch's *t*-tests. Results are shown in Table I. We could not perform this comparison for tasks 11 and 14 because too few participants used tag rules. For tasks 10, 13, 15, 18, the accuracy for tag-rule posts was not significantly different from that for per-post posts. However, for task 21, accuracy on per-post posts was higher, because many of those who wrote tag rules for this task assumed the tags they choose ("photos" and "photography") applied to all the photo posts.

**Accuracy on distractor tasks:** We also computed accuracy for the six experimental distractor tasks. For each task, the participant received a score of 1 if the task was completed successfully, or 0 if it was not. There was no statistically significant difference between average accuracy for the tag rule group ( $M = 0.88$ ;  $SD = 0.11$ ) and average accuracy for the per-post group ( $M = 0.90$ ;  $SD = 0.09$ ;  $t(159) = 1.974$ ;  $p > 0.99$ ).

**Time to completion of privacy tasks:** We computed time to completion (in seconds) for all experimental privacy tasks. We compared these times using two-tailed Welch's *t*-tests. Tasks solved using tag rules were completed significantly faster than tasks solved using

<sup>2</sup>We use Holm-Bonferroni adjusted  $p$  values when testing multiple hypotheses on the same data set.

| Task | Time (Seconds) |                        |          |                       | Significance |
|------|----------------|------------------------|----------|-----------------------|--------------|
|      | <i>N</i>       | Tag rules<br>Mean (SD) | <i>N</i> | Per-post<br>Mean (SD) |              |
| 10   | 7              | 153.14 (37.37)         | 17       | 205.71 (58.78)        | n.s.         |
| 13   | 8              | 87.88 (36.93)          | 16       | 157.00 (77.65)        | $p < 0.05$   |
| 15   | 8              | 189.00 (78.71)         | 16       | 303.44 (92.23)        | $p < 0.05$   |
| 18   | 8              | 129.75 (47.30)         | 16       | 249.5 (105.11)        | $p < 0.01$   |
| 21   | 6              | 173.67 (54.11)         | 18       | 348.39 (144.90)       | $p < 0.005$  |

TABLE I  
ACCURACY AND TIME COMPLETION FOR EXPERIMENTAL PRIVACY TASKS

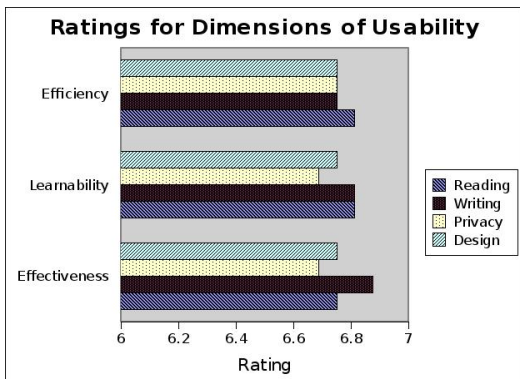


Fig. 4. Average participant ratings of the major Plog and WordPress features.

per-post policies (see Table I).

**Time to completion of distractor tasks:** We summed the task completion times (in seconds) for all distractor tasks for each participant. We compared the times of each group using a two-tailed  $t$ -test. There was no statistically significant difference between the tag-rule group ( $M = 159.68$ ;  $SD = 177.82$ ) and the per-post group ( $M = 207.81$ ;  $SD = 238.75$ ) in terms of distractor tasks completion time ( $t(142.12)=1.98$ ;  $p > 0.56$ ).

**Subjective evaluation results:** Sixteen of the participants rated fifteen blog features on a seven point Likert scale for three aspects of usability: effectiveness, efficiency and learnability [14]. We performed ANOVAs on each dimension of usability by comparing the averages of the Likert ratings for the four major blog features (reading, writing, privacy and design) and found no statistical differences (effectiveness:  $F(3,13) = 0.53$ ,  $p = .66$ ; learnability:  $F(3,13) = 0.30$ ,  $p = 0.82$ ; efficiency:  $F(3,13) = 0.08$ ,  $p = 0.97$ ). We also compared the scores of the tag rule group and the per-post group for each usability dimension using two-tailed  $t$ -tests. We found no significant differences in usability ratings.

**Size of tag rule policies:** In this analysis, we only consider participant data for tasks that the participant completed with perfect accuracy using tag rules. On av-

erage, participants' tag rule policies came within one tag of the optimal policy. Half of the perfectly accurate tag-based policies created by our participants were optimal. The mean difference in size between the optimal and participant policies was 0.92 ( $SD = 1.38$ ,  $Q_3 = 1$ ).

We also looked at all tag-based policies regardless of accuracy. Participants in general did not write redundant tag rules (those that if removed from the policy, would not reduce the number of posts covered). On average, the tag-based policies created by our participants contained 0.79 redundant tags ( $SD = 1.28$ ).

## VI. DISCUSSION

Our evaluation results show that:

- The tag-rule group created policies that were just as accurate as the per-post group.
- The tag-rule group applied policies significantly faster than the per-post group.
- Participants wrote near-optimal tag-based privacy policies in terms of size.

We conclude that content owners in Web 2.0 applications can benefit from tag-based privacy control.

In our evaluation, we made a deliberate choice not to force study participants to use either tag-based or per-post privacy policy authoring. We did not advertise the benefits of tag rules. Nor did we provide any guidance or training in the mental tasks of authoring tag rules, which requires the ability to abstract away from post content and reason from precondition to effect. Given that our participants were blogging novices, we find it promising that a third of them were able to use and prefer the abstraction of tag-based policy authoring in a short period of time.

We are also encouraged to see that there were no statistically significant differences in accuracy on privacy tasks between the per-post and tag-rule sets. In our experiment, per-post participants only had to think about 30 short posts, and no task involved setting a policy on more than 5 posts. On the other hand, tag-rule participants had to deal with over 100 unique tags, some of which were spurious, occurred infrequently, or were not topically related to the post. Even so, they were able to write policies that were as accurate as those authored by per-post participants. As the size of a blog grows, the number of posts usually quickly exceeds the number of tags. So a real Plog user would probably find it increasingly efficient and accurate to use tag-rule policy authoring rather than per-post policy authoring.

Based on our evaluation results, we are confident that Plog addresses the problematic properties of security interface design raised by Whitten et al [16]. Plog is

unlikely to suffer from the *unmotivated user property* because bloggers already tag their posts and a third of our study participants wrote tag-rule policies without prompting (so we know that tag-rule authoring is not too abstract or difficult for most bloggers). In several ways, the Plog system also addresses the *lack of feedback property*. For example, the Plog plugin prominently displays, in plain English, the privacy policy applicable to each post. It also uses a tag-rule authoring interface previously demonstrated to be effective for policy authoring [3]. Another problem that arises in security systems is the *barn-door property*, in which something is secured only after it has been published unprotected. We partially mitigate this problem by associating privacy with tagging, a task that a most bloggers already do. By making privacy policy authoring so much a part of the blogger's normal workflow, we also strengthen the *weakest link* (the content owner).

In our study, access control policies were implemented retroactively. Bloggers will most likely set access control policies at the time of content creation. If users do not express their privacy preferences with tag rules, our system will recognize patterns in previous policies that identify a set of subjects with the same access rights on similar content. As seen in section III-A, tag usage on private posts is consistent and tag rules could easily be suggested from existing tags. Otherwise, the system could facilitate tag rule creation by associating new tags with posts and subjects.

## VII. CONCLUSION

The Web 2.0 has given users new opportunities to create content and share with others. Unfortunately, the lack of privacy in most Web 2.0 applications has had significant impact on some users' lives. Developers of these applications have a responsibility to give users the tools to manage their own privacy, but it is hard to create general tools that a wide variety of users can use effectively.

Our solution, Plog, is a tag-based privacy policy control that allows users to mediate their competing desires for privacy and publicity by specifying their privacy preferences in a language that is natural to them. Tag-based policy authoring produces policies that are short, precise, and easy to create and maintain. Our implementation of Plog includes several user-interface features designed to make privacy management as simple and non-intrusive as possible and could be easily integrated into current Web 2.0 applications. Since tags are a ubiquitous feature of Web 2.0 services, tag-based privacy policies can be transferred easily between sites. And by choosing tags as

the basis for a policy language, we can suggest tags while the bloggers type their posts. This not only provides an attractive feature that reduces work and promotes consistent tag usage, but also provides policy inference that could help prevent unintended privacy disclosure.

## REFERENCES

- [1] S. B. Barnes. Privacy paradox: Social networking in the United States. *First Monday*, 11(9), September 2006.
- [2] E. Bertino, E. Ferrari, and E. Perego. Content-based filtering of web documents: The MaX system and the EUFORBIA project. *International Journal of Information Security*, 2(1):45–58, November 2003.
- [3] C. Brodie, C. Karat, and J. Karat. Intelligible access control: An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In *Proceedings of SOUPS*, July 2006.
- [4] M. Brownlow. Email and webmail statistics. Available from <http://www.email-marketing-reports.com/metrics/email-statistics.htm>, January 2009.
- [5] A. Doyle. How blogging and social networking can impact your job search. <http://jobsearch.about.com/od/jobsearchblogs/a/jobsearchblog.htm>.
- [6] Marti A. Hearst and Daniela Rosner. Tag clouds: Data analysis tool or social signaller? In *HICSS '08: Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 160, Washington, DC, USA, 2008. IEEE Computer Society.
- [7] A. Lenhart and S. Fox. Bloggers: A portrait of the internet's new storytellers. Available from [http://www.pewinternet.org/pdfs/PIP\\_Bloggers\\_Report\\_July\\_19\\_2006.pdf](http://www.pewinternet.org/pdfs/PIP_Bloggers_Report_July_19_2006.pdf), July 2006.
- [8] moveon.org. Petition: Facebook, stop invading my privacy! Available from <http://www.facebook.com/group.php?gid=5930262681>, November 2007.
- [9] R. Pegoraro. Facebook backs into a 'bill of rights'. Available from <http://www.washingtonpost.com/wp-dyn/content/article/2009/02/18/AR2009021803121.html>.
- [10] K. Poulsen. Myspace predator caught by code. Available from <http://www.wired.com/news/technology/0,71948-0.html>, October 2006.
- [11] H. A. Rowley, Y. Jing, and S. Baluja. Large scale image-based adult content filtering. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, February 2006.
- [12] D. Rowse. Blog stalkers - personal safety for bloggers. Available from <http://www.problogger.net/archives/2006/02/07/blog-stalkers-personal-safety-for-bloggers/>, February 2006.
- [13] E. Simonetti. I was fired for blogging. Available from [http://news.cnet.com/I+was+fired+for+blogging/2010-1030\\_3-5490836.html](http://news.cnet.com/I+was+fired+for+blogging/2010-1030_3-5490836.html), December 2004.
- [14] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha. *User Interface Design and Evaluation (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann, March 2005.
- [15] J. Ze Wang, J. Li, G. Wiederhold, and O. Firschein. System for screening objectionable images. *Computer Communications*, 21(15):1355–1360, 1998.
- [16] Alma Whitten and J. D. Tygar. Why johnny can't encrypt: a usability evaluation of pgp 5.0. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 14–14, Berkeley, CA, USA, 1999. USENIX Association.
- [17] Wordpress. <http://wordpress.org>.