Modeling Exposure in Online Social Networks

Andrew Cortese
Department of Computer Science
University at Albany – SUNY
Albany, NY, USA
acortese@albany.edu

Amirreza Masoumzadeh
Department of Computer Science
University at Albany – SUNY
Albany, NY, USA
amasoumzadeh@albany.edu

***Abstract*—In online social networks (OSNs), the privacy of users is impacted by exposure of information about those users to other users of the system. Various factors, including design and user behavior, may affect the degree to which information about users is exposed. We propose the notion of knowledge exposure that measures the probability that information about users will be seen by others. We argue that such a measure can give OSN users and designers insight about how privacy is affected based on system design and user behavior. We present exposure as a promising notion that can complement privacy control efforts in an OSN rather than replacing existing measures such as access control. We provide a formal model of exposure in an OSN, and demonstrate through experiments how it can be calculated for various information items.**

***Index Terms*—online social networks, exposure, privacy**

I. Introduction

Online social networks (OSNs) are Web applications where users interact with one another by forming connections (friendships) and sharing information. An OSN like Facebook, LinkedIn, or Google+ contains a large amount of personal information about the users of that system. Much of this information is privacy-sensitive, and users may want to limit who can access their information. Various access control models have been proposed for OSNs to enable users control sharing of their privacy-sensitive information with other users. Such proposals specify and enforce access control policies based on user-to-user relationships [2, 3, 6, 9, 12]. Regardless of the choice of the access control model, access control policies in an OSN determine the authorized users for accessing a piece of information. In other words, the authorization determines who *can access* the information. We note, however, that there are other factors involved that could determine who *would access* the information. Everyone may not have the same chance to access certain information even if authorized. In the same way, a piece of information may not be accessed by every individual authorized for it.

As users browse an OSN, they obtain (access) information about other users in the system. More precisely, each page on OSN presents pieces of knowledge to a user about other users. For example, as Bob browses Alice's profile page on Facebook he learns Alice's birthday. Or as Bob reviews his news feed, he learns about a new photo in which Alice has been tagged. As users obtain pieces of knowledge about other users, those pieces of knowledge become *exposed*. In this work, we propose and formalize the notion of *exposure* for a piece of knowledge as the chance (or probability) that it is accessed by a user. In other words, while authorization determines a possible access, exposure measures chance of materializing that access.

In this paper, we propose a foundational model and associated algorithms for defining and calculating knowledge exposure in OSNs. Knowledge exposure as a measure can be an indicator of privacy risk of user's information. Our intuition is that, given the same impact of privacy violation for two knowledge items, a user will be more concerned about the knowledge item that has more exposure than the item that is less exposed. Such a measure can form the basis of an access control model based on desired exposure levels. Also, it can also inform users about the effect of their behavior/policies on exposure of their data. Finally, it can be employed by OSN developers to test the effects of changes to system design on user privacy, in order to tune the designs and to inform and prepare their users more adequately.

Our contributions in this paper can be summarized as follows.

1) A formal model of an OSN and the knowledge contained therein. This model is abstract and can be used to represent many different kinds of OSNs.
2) A 3-step modular algorithm for computing exposure of OSN pages, knowledge items, and users. Each of the three modules of this algorithm can be exchanged for an alternative approach, without affecting the others.
3) A use case for a simple OSN, in which we demonstrate how the generic model (contribution 1) can be instantiated to represent a real system.
4) An experiment in which we apply the simple OSN (contribution 3) to a real-world dataset from Facebook, and implement the algorithm from contribution 2 to compute exposure on that dataset.

The rest of the paper is organized as follows. We provide an overview of factors that we consider to impact exposure of knowledge items in Section II. In Section III, we establish a generic framework for modeling an OSN and the knowledge contained within it. We outline our methodology for computing exposure of pages, knowledge, and users in Section IV. In Section V, we discuss the application of our model to a

978-1-5386-2487-6/17/$31.00 ©2017 IEEE

specific OSN design based on a simple version of Facebook, and discuss the results of our experiment in Section VI. We finally review the related work in Section VII.

II. Factors Impacting Exposure in OSN

There are several factors that affect information exposure in an OSN. We argue that the two main factors are the design of the OSN, and the behavior of its users. Here, we give an overview of how these factors impact exposure, and argue in the rest of the paper how the model and experimental results support our hypotheses. Figure 1 shows the relationship between various factors. In the diagram, an arrow from one item to another indicates that the first item impacts the second.

User preferences, browsing habits, goals for using the system, and other factors (collectively referred to here as "user psychology") have an impact on the behavior of users in the system. These users take on two distinct roles, the role of a knowledge stakeholder (knowledge about them is contained on pages in the system), and the role of an observer (someone browsing the system and learning knowledge about others). Stakeholder behavior and observer behavior are both impacted by user psychology, but they in turn impact exposure in different ways.

Decisions made in the design of the OSN impact both the layout of each page, as well as the topology of the system's Webpage hyperlink graph. These in turn have an affect on the behavior of users who are observing information contained on pages in the system. For example, the design decision regarding whether or not to show "Alice has been tagged in a photo" on Bob's news feed determines whether or not Bob might navigate to the photo page.

The page layout and hyperlink topology are also affected by the behavior of knowledge stakeholders. For example, Alice's decision to become friends with Bob might cause there to be an additional entry on each of their friend lists. There will also now be hyperlinks from each person's friend list to the other's profile.

Finally, the behavior of observer users (which was influenced by their psychology and by the OSN layout/topoligy) directly determines the exposure of knowledge in the system. For example, Alice's decision (influenced by her browsing habits, the availability of a hyperlink, and the presence of desired knowledge) to navigate to Bob's profile page increases the exposure of knowledge contained on that page.

An additional important factor is access control. Intuitively, denied authorizations results in no exposure, while grant authorizations result in some (even if nominal) exposure. The OSN model we present in this paper does not include access control rules, and we therefore do not discuss this factor in as much detail as the others. However, we foresee access control as having an important relationship to exposure, and therefore plan to address it in future work.

III. The OSN Model

We now present our model of an OSN. The core of the model has two representations: the knowledge model, and the navigation model. The knowledge model represents the system's users, their data, and the relationships between them. The navigation model consists of a Web graph that represents the page-hyperlink structure of the OSN website itself. Pieces of knowledge from the knowledge model are exposed to users on pages in the navigation model.

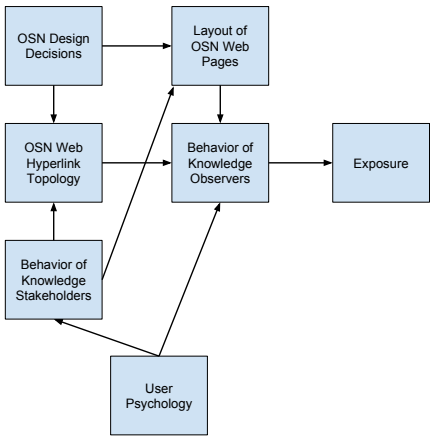


Figure 1: Factors Impacting Exposure

A. Knowledge Model

The knowledge graph is derived from an underlying knowledge ontology, which describes the set of knowledge contained in the system. Each individual piece of knowledge is called a knowledge statement, and corresponds to one edge (and its endpoints) in the knowledge graph. The stakeholder function maps a knowledge statements to the set users whose privacy is impacted by the exposure of that knowledge statement.

1) Knowledge Ontology: A knowledge ontology is a tuple $\langle U, O, \mathcal{R} \rangle$, such that:

- U is the set of users in the system. Each element $u \in U$ represents an individual who has an account in the online social networking system.
- O is the set of objects in the system. These are items such as photos, text posts, location check-ins, pieces of profile information (e.g., name, home address), etc.
- $\mathcal{R}$ is a collection of binary relations representing the relationships in the system. Each element of $\mathcal{R}$ is a set $R_t \subseteq (U \cup O) \times (U \cup O)$. R_t contains pairs of users and/or objects that have the relationship t (i.e. t is the relationship type). For example, if users u_1 and u_2 are friends, then $(u_1, u_2) \in R_{friend}$. If $O \neq \emptyset$, then $\mathcal{R}$ must contain at least the element R_{own} such that $\forall o \in O. \exists u \in U. (u, o) \in R_{own}$. In other words, if there are objects in the model, then each object must have an owner assigned to it.

2) Knowledge Graph: The knowledge graph is the graph-representation of the knowledge ontology. It is a directed graph $G_K(V_K, E_K)$ with typed edges and typed vertices such that:

- $V_K = U \cup O$
- $E_K = \{t(x,y) \mid R_t \in \mathcal{R} \wedge (x,y) \in R_t\}$

3) Knowledge Statements: From the knowledge graph we obtain the set $\mathcal{K}$ of knowledge statements. A **knowledge statement** is a a subgraph of G_K consisting of two nodes x and y, and the labeled edge $(x,y) \in R_t$ for some $R_t \in \mathcal{R}$. It is expressed as $t(x,y)$. For example, if users u_1 and u_2 are friends (i.e. $(u_1,u_2) \in R_{friend}$), then the knowledge statement $friend(u_1,u_2)$ expresses this fact.

Each knowledge statement has a type and a Unix-style timestamp. The type corresponds to the relation to which its corresponding edge belongs. The timestamp records the time that the corresponding edge was created. For a given knowledge statement $k = t(x,y)$, the type is denoted by $type(k) = t$, and the timestamp is denoted by $time(k)$.

4) Knowledge Stakeholders: A stakeholder for a piece of knowledge is a user whose privacy is affected by the exposure of that knowledge. Function $\int : \mathcal{K} \to \mathcal{P}(U)$ assigns stakeholders to knowledge items. Suppose k is the statement $t(x,y)$. Then $\int(k) = \{u \in U \mid x = u \ \vee \ y = u \ \vee \ (u,x) \in R_{own} \ \vee \ (u,y) \in R_{own}\}$. Therefore, a user is a stakeholder for a knowledge statement if that user either appears in the knowledge statement, or owns an object that appears in the knowledge statement.

B. Navigation Model

The navigation model is a tuple $\langle G_N(P,H), contains, priority, weight\rangle$ such that:

- $G_N(P,H)$ is the navigation graph, which represents the Webpage/hyperlink topology of the OSN Web application.
- $contains$ is a function that keeps track of the pages that a particular knowledge statement appears on.
- $priority$ is a function that captures the "prominence" of a knowledge item on a page.
- $weight$ is a function that assigns edge weights to hyperlinks. The weight of a hyperlink is the probability that a user will click on that hyperlink, rather than another one on the same page.

The navigation graph is the Web hyperlink graph representation of the OSN system. P is a set of Web pages, and $H \subseteq P \times P$ is a set of hyperlinks. $G_N(P,H)$ is a directed graph with weighted edges. This represents the system that users navigate. Each page contains knowledge statements that are potentially exposed when a user navigates to that page. For an edge $e = \langle p_1, p_2\rangle$, the weight of e is the probability that a user currently visiting page p_1 will navigate to p_2 by following hyperlink e.

The function $contains$ is defined as follows: Suppose $p \in P$ and $k \in \mathcal{K}$. If $p \in contains(k)$, then knowledge statement k appears on p and can be learned by any user that navigates to page p. In other words $contains(k)$ is the set of all pages that k appears on. The function $contents(p) : P \to \mathcal{P}(\mathcal{K})$ is a pseudo-inverse of the $contains$ function. $contents(p) = \{k \mid p \in contains(k)\}$.

Table I: Example Set of Pages

Page	Contents	Priority
p_1	k_1	0.50
	k_2	$.\overline{3}$
	k_3	$0.1\overline{6}$
p_2	k_1	$0.\overline{6}$
	k_3	$0.\overline{3}$
p_3	k_2	$0.\overline{6}$
	k_3	$0.\overline{3}$
p_4	$k_\perp$	1

The $priority$ function is defined as follows: $priority : \mathcal{K} \times P \to \mathbb{R}$. If $priority(k,p) = c$, then we say that knowledge statement k has priority c on page p. The priority c is a measure of the prominence of k on page p. A higher priority value indicates that the knowledge statement is more prominent on the page. The priority function can be implemented in different ways to capture the fact that users might be more likely to look at certain knowledge statements, as opposed to others on the same page. In section IV-B, we discuss how to compute and interpret priority.

If a page p_1 contains no knowledge (i.e. $/\exists k. \ \ p_1 \in contains(k)$), then p_1 is said to contain a special knowledge statement called the *empty knowledge statement*. We denote the empty knowledge statement as $k_\perp$. Therefore, for an empty page p_1, $contents(p_1) = \{k_\perp\}$ This is important in computing knowledge exposure from page exposure, as we will see in section IV.

Example 1: Table I shows an example of part of the Navigation Model for a hypothetical OSN. This example illustrates how a knowledge statement can appear on multiple pages, and how a page can have multiple different knowledge statements contained within it. If a page has no knowledge statements (e.g., p_4), it is assigned the empty knowledge statement (i.e. $k_\perp$). We will refer back to this example in subsequent sections to help illustrate how exposure is computed.

IV. Computing Exposure

We discuss three distinct levels of knowledge exposure: page exposure, knowledge statement exposure, and user exposure. Those are denoted by functions e_p, e_k, and e_u, respectively. These three values are closely linked. In particular, e_u depends on e_k, and e_k depends on e_p.

We define exposure, in general, is a probabilistic measure. If $p \in P$ is a page, then $e_p(p)$ is the probability that at any given moment, a random user will be visiting page p. For a knowledge statement $k \in \mathcal{K}$, $e_k(k)$ is the probability that at any given moment, a random user will be looking at an instance of k. Finally, for a user $u \in U$, $e_u(u)$ is the probability that, at any given moment, a random user will be looking at some knowledge statement for which u is a stakeholder. We derive this notion primarily from the PageRank algorithm [14], which we employ in our calculation of page exposure.

The first step in measuring the exposure of a knowledge item in an OSN is to compute the exposure of the OSN Web pages containing that knowledge. Once the page exposure values are obtained, the knowledge exposure for each piece of

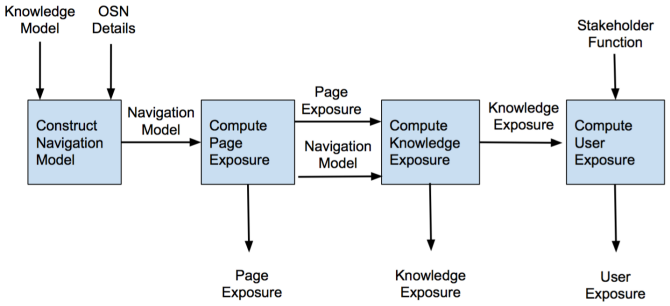


Figure 2: Calculating Exposure

knowledge can be computed by aggregating the page exposure of all the pages containing that piece of knowledge. This calculation must also take into account the layout of the pages with respect to where the knowledge is displayed to users. Finally, the user exposure for a user can be computed by aggregating the exposure scores for all of the knowledge in which that user is a stakeholder.

A summary of this process is as follows:

1) Given the knowledge model, and the details of the OSN being modeled (i.e. the features of the actual, implemented Web application), construct the navigation model of the OSN.
2) Analyze the topology of the navigation graph to compute page exposure.
3) Compute knowledge exposure from the computed page exposure values. This requires use of the $contents$ and $priority$ functions.
4) Compute user exposure from the computed knowledge exposure values. This requires use of the stakeholder function $\int : \mathcal{K} \rightarrow \mathcal{P}(U)$.

This process is designed to be highly *modular*. That is, each of the four steps can be thought of as an abstract module. Each module has specific inputs and outputs, but its implementation details do not depend on the details of the other modules. Figure 2 depicts each module as a "black box" algorithm.

In the subsequent sections, we will explain the details of the step-by step process for computing exposure. In section IV-A, we describe how to calculate page exposure. In section IV-B, we describe how to calculate knowledge exposure from page exposure. In section IV-C, we describe how to compute user exposure from knowledge exposure.

A. Page Exposure

We utilize the PageRank algorithm [14] for calculating page exposure in this paper. We leave exploration of other possibly useful network centrality measures [4] for future work.

PageRank was originally developed as a tool for quantifying the importance of Web page in order to help tailor the results of search engine queries. PageRank approximately simulates the behavior of a user browsing a set of Web pages by clicking a randomly-selected hyperlink on each page. The algorithm also allows for the possibility that the user will decide to jump directly to a page without following hyperlinks; this behavior is called teleportation.

We customize PageRank in several ways to better fit our needs in measuring exposure of pages. First, we specify the probability that, at any given moment, the user will teleport away from the current page. This probability is known as the *damping factor*. Second, we specify, for a particular page, the probability that this page will be the destination for a teleportation. This probability is called the *reset value* of a page. Finally, we assign edge weights to hyperlinks. The weight of a hyperlink is the probability that the user will follow that hyperlink when leaving the page (as opposed to another hyperlink on the same page).

The purpose of the damping factor and reset values are to simulate the user behavior of foregoing hyperlinks and navigating directly to a specific page. In our model we employ these features to capture the scenario in which a user chooses to return to her homepage.

The purpose of the edge weights is to capture the notion that users may be more inclined to follow a particular hyperlink over another. If one hyperlink is deemed to be more likely to be selected (based on the design of the system or the hyperlink's location on a page), it will be assigned a higher weight.

When PageRank runs, it approximately simulates a random walk of the graph. Each vertex will be assigned a rank which depends on two factors: (1) the number of incoming links, and (2) the recursively-calculated rank of the pages that link to it. When weights are used, a higher weight on an incoming link causes effect of that link on the page's rank to be greater.

After executing the PageRank algorithm on the navigation graph, there will be a value $e_p(p)$ for each $p \in P$. As discussed above, each page's e_p value is interpreted as a probability. Therefore, $\sum_{p_i \in P} e_p(p_i) = 1$. This is guaranteed as an invariant of the PageRank algorithm [14]. We rely on this fact in our calculation of e_k in the subsequent section.

B. Knowledge Exposure

Suppose that $k \in \mathcal{K}$ is a knowledge statement. The knowledge exposure, denoted as $e_k(k)$, is the probability that, at a given moment, a random user is visiting some page p such that $p \in contains(k)$ and is currently looking at k (as opposed to some other knowledge statement on the same page).

Since a knowledge statement may appear on more than one page, $e_k(k)$ relies on two factors: the e_p of the pages containing k, and the priority values for k on each of those pages.

We define an *instance* of k to be a member of the set $\mathcal{I}_k = \{\langle k, p\rangle \mid p \in P \wedge p \in contains(k)\}$, which represents an occurrence of k on a particular page. For example, if $contains(k) = \{p_1, p_2\}$ then there are two instances: $\mathcal{I}_k = \{\langle k, p_1\rangle, \langle k, p_2\rangle\}$. The *instance-exposure* of an instance, $e_{k_i}(k, p_i)$ is based on $e_p(p_i)$ and $priority(k, p_i)$. In turn, the e_k of k is an aggregate of the instance-exposures of all instances of k.

The priority $priority(k_1, p_1)$ is the probability that, if a user is visiting p_1, then k_1 is the knowledge statement she

is looking at. The priority of a knowledge statement could depend on many different factors, including but not limited to: the position of the knowledge item on the page, the type of knowledge statement, the timestamp of the knowledge statement, and the endpoints or stakeholders for the knowledge statement (and their relationship to the observer). For now, $priority$ is left as an abstract function that can be customized for different OSN modeling needs. In section V, we describe how exposure is implemented in our specific use case and subsequent experiment.

Once the e_p value for each page, and the $priority$ value for each instance of a knowledge item are computed, we can compute the e_k of a knowledge item as follows:

$$e_k(k) = \sum_{p_i \in contains(k)} e_p(p_i) * priority(k, p_i)$$

This is where the existence of the empty knowledge statement is important, as previously mentioned in Section III-B. If some page p_1 was allowed to exist such that $contents(p_1) = \emptyset$, then the sum $\sum_{k \in \mathcal{K}} e_k(k)$ would not be equal to 1. Thus, we instead set $contents(p_1) = \{k_\perp\}$ This correctly captures the possibility that a user is visiting a page that has no knowledge (i.e., contains only $k_\perp$), which lowers the probability that the user is looking at some non-empty knowledge statement.

Example 2: Consider the example in table I. Suppose that we have computed the page exposure of each page using the process outlined in section IV-A, and obtained the following page exposure values: $e_p(p_1) = 0.25$, $e_p(p_2) = 0.35$, $e_p(p_3) = 0.25$, and $e_p(p_4) = 0.15$.

We can now compute knowledge exposure as follows (we show the details of the calculation for k_1, but only the result for k_2 and k_3. The reader can easily verify the arithmetic):

$$\begin{aligned} e_k(k_1) &= e_p(p_1) * priority(p_1, k_1) + e_p(p_2) * priority(p_2, k_1) \\ &= 0.25 * 0.5 + 0.35 * 0.\overline{6} \\ &= 0.358\overline{3} \\ e_k(k_2) &= 0.25 \\ e_k(k_3) &= 0.241\overline{6} \end{aligned}$$

Notice that, excluding $k_\perp$, the sum of the exposure values for all knowledge statements is $e_k(k_1) + e_k(k_2) + e_k(k_3) = 0.85$ (as opposed to 1). Since p_4 contains no knowledge, some of the exposure value on that page seems to be "lost" when computing knowledge exposure. This is why we have $k_\perp$. We can compute $e_p(k_\perp) = e_p(p_4) * 1 = 0.15$. Now the sum of the exposure for all knowledge statements is $0.85 + 0.15 = 1$.

C. User Exposure

For some user $u \in U$, exposure $e_u(u)$ is computed simply by calculating the sum of the e_k values of the knowledge statements for which that user is a stakeholder. Suppose $\mathcal{K}_u \subseteq \mathcal{K}$ such that $\mathcal{K}_u = \{k \mid u \in \int(k)\}$. Then $e_u(u) = \sum_{k_i \in \mathcal{K}_u} e_k(k_i)$

V. Use Case: Exposure in a Facebook-Like OSN

In this section, we present a use-case implementation of our exposure model for a Facebook-like OSN.

A. OSN Model

We consider a simplified version of Facebook where each user has a profile page and a news feed. A user's profile page includes her "wall," upon which other users may leave messages (wall posts). The profile page also lists that user's friends. The news feed contains a list of recent "stories" about the user's friendship network. Stories in this system include recently-created friendships or wall posts for/by the user's friends. We assume that wall posts and news feed stories are displayed in inverse chronological order. In particular, the following kinds of knowledge is shared in the OSN:

- The existence of a friendship between two users
- The fact that a particular user posted on another user's wall or on her own wall (which is commonly known as "status" message on Facebook)

Each user u has a news feed page, a profile page, a friend list page, and a wall page. The friend list page contains knowledge statements of the form $friend(u, f_i)$ and has a hyperlink to the profile page of each friend f_i. The wall page contains knowledge statements of the form $postedOnWall(a_j, u)$, and has a hyperlink to the profile page of each post author a_j. The news feed contains the most recent knowledge statements for which a friend of u is a stakeholder, and has hyperlinks to pages owned by the users mentioned in those statements. The profile page for each user contains no knowledge, and serves only as a landing page for other users navigating to u's cluster of pages.

For a precise specification of how the knowledge graph and navigation graph are constructed, see Appendix A.

Figure 3a shows a sample knowledge graph. This social network has three users. The types of knowledge statements are $friendship$ and $wallpost$ (e.g. u_2 posted on u_3's wall). Figure 3b shows the corresponding navigation graph, creating using the algorithm outlined above. Each vertex is a page, and each directed edge is a hyperlink. For each user u_i, the vertex n_i represents that user's news feed, p_i represents their profile page, f_i represents their friend list page, and w_i represents their wall page.

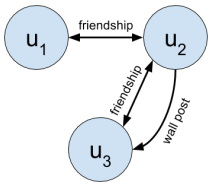
VI. Experimental Results

In this section, we conduct a set of experiments on our use case OSN, described in Section V, using a real-world dataset.

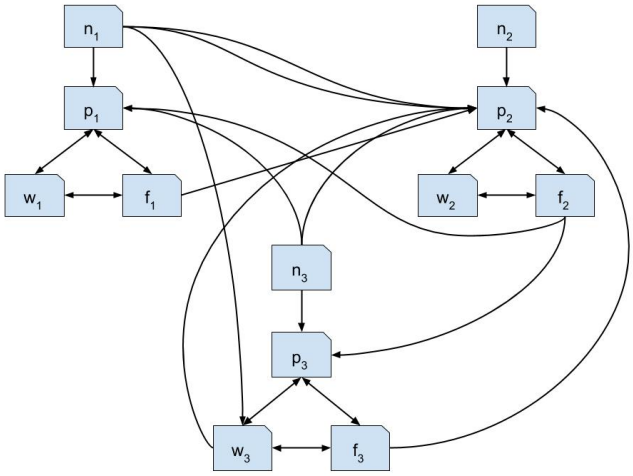
A. Dataset

We utilize a dataset from a 2009 study on user interaction on Facebook [19]. The data set consists of 63,891 users, with 817,091 friendships and 876,994 interactions (wall posts). Each wall post is assigned a timestamp based on the time it was created. Each friendship is assigned a timestamp based on the time that the two users became friends, if this could be determined. Otherwise, it is assigned timestamp 0.

Figure 3: An example of a simple knowledge graph, and the corresponding navigation graph for our use case.



(a) Example Knowledge Graph



(b) Example Navigation Graph

B. Setup

We implemented a program to import our dataset and construct the knowledge and navigation graphs (details in Appendix A). We then used the personalized PageRank algorithm (see section IV-A), provided by the Python iGraph library [5], to compute page exposure for vertices in the navigation graph. We customized the algorithm with the following parameters:

1) The damping factor is set to 0.15. This means that the probability of 'teleporting' away from a page (instead of clicking one of its hyperlinks) at any given time is 0.85. (The teleportation probability is always $1 - d$, where d is the damping factor).
2) The reset value for a node representing a news feed page is set to 1, while the reset value for each other node is set to 0. The PageRank implementation automatically redistributes this evenly among all feed-page nodes so that the sum of all reset values is 1. This ensures that if a user teleports away from the page she is visiting, then the probability that she will arrive at a feed page is 1, and each feed page has an equal chance of being the destination. This simulates the user behavior of returning back to their news feed during a browsing session.

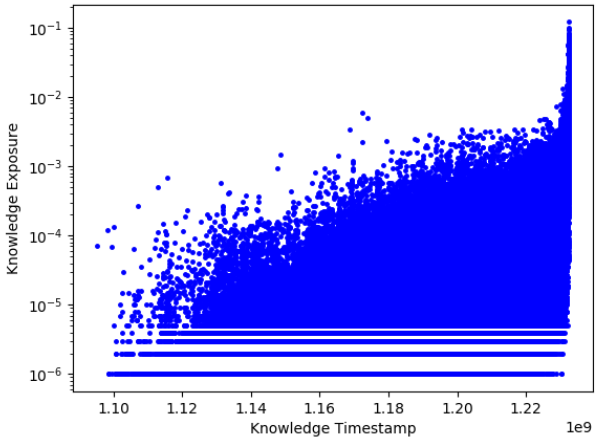


Figure 4: Exposure of knowledge statements vs. their timestamps

3) The edge weights are set based on the contents of the pages and proportional to the priority of the corresponding knowledge items (details given in Appendix A). The PageRank implementation will automatically normalize these values so that the sum of the weights of all outgoing edges from a page is 1. When a user navigates away from a page by clicking a hyperlink, then $weight(e)$ denotes the probability that e is the hyperlink she will choose.

Once the PageRank calculation gives us our e_p value for each page, we compute e_k for each knowledge statement and e_u for each user as discussed in section IV.

It is important to note that all exposure values are multiplied by 100 to preserve precision. Therefore, the sum of all page exposure values is 100, rather than 1. Similarly, the sum of all knowledge exposure values is 100, rather than 1. This is merely a convenience for the reporting of values. The exposure values can be interpreted in nearly the same way as before.

C. Knowledge Characteristics and Exposure

We investigated the relationship between exposure of knowledge statements and their characteristics, more specifically, their timestamp and appearance on pages. Figure 4 shows the relationship between the timestamp for a knowledge statement, and that knowledge statement's exposure. The exposure shows a strong correlation with time (Spearman correlation = 0.71). We utilize Spearman correlation because it captures the fact that, as time increases, exposure increases as well. This is not expressed well by a Pearson correlation, since Pearson only tracks linearity between two variables.

Figure 5 shows the relationship between the number of pages that a knowledge statement appears on, and that knowledge statement's exposure. The Pearson correlation for this relationship is 0.90. This suggests that the number of pages that a knowledge statement appears on has a strong positive effect on its exposure.

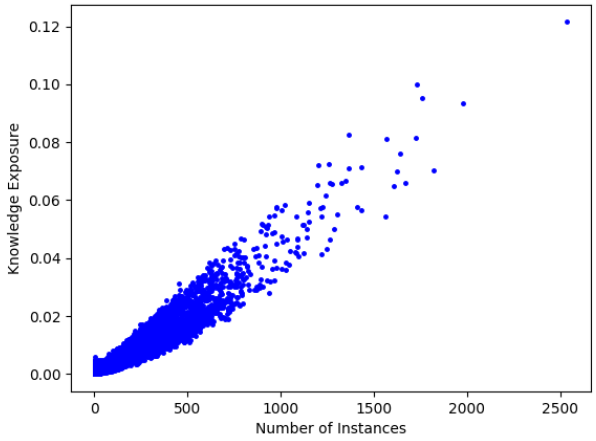


Figure 5: Exposure of knowledge statements vs. the number of pages they appear on

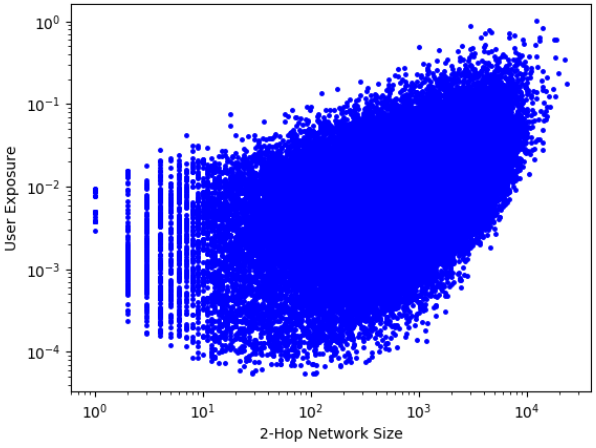


Figure 6: Exposure of users vs. their two-hop neighborhood size

D. User Characteristics and Exposure

We also investigated the exposure of users in relation to their characteristics. In figure 6, we show the relationship between a user's two-hop neighborhood size and their user exposure. The Pearson correlation is 0.53, showing a moderate effect of neighborhood size on user exposure.

E. Observer Behavior and Exposure

As discussed in section II, one of the primary factors that impacts exposure is the behavior of observer users. As users browse Web pages in the system, they may arbitrarily decide to return to their news feed, instead of following a hyperlink to another page.

In our experiment, we capture this behavior using the damping factor and reset values in the PageRank algorithm. The damping factor controls the probability of "teleporting", or navigating directly to a page, instead of following a hyperlink. A damping factor of d, such that $0 \leq d \leq 1$, indicates that the teleportation probability is $1 - d$. Meanwhile, the reset value for a page controls the probability that that page will be the destination of a teleportation.

Damping Factor	Total News Feed Exp.	Total Profile Exp.
0.15	84.976	14.999
0.50	50.027	49.997
0.85	15.015	84.997

Table II: The effect of the PageRank damping factor on page exposure by page type in OSN-1

We assign a reset value of 1 to each news feed page, and a reset value of 0 to each other page. These values are automatically redistributed by the PageRank implementation, so that the destination of a teleportation will always be a news feed page. In the experiments discussed thus far, we have used a damping factor of .15, indicating that the probability of teleporting away from any given page is 0.85.

In this experiment, we observe the effect of different damping factor values on the page-exposure of different kinds of pages. We run PageRank with damping factors of 0.15, 0.5, and 0.85. Suppose $F \subseteq P$ is the set of all friend list pages, $W \subseteq P$ is the set of all wall pages, $Q \subseteq P$ is the set of all profile pages, and $N \subseteq P$ is the set of all news feed pages. For each damping factor value, we compute the followings: $\sum_{f \in F} e_p(f)$, $\sum_{w \in W} e_p(w)$, $\sum_{n \in N} e_p(n)$, and $\sum_{q \in Q} e_p(q)$.

Table II shows the results. As the damping factor increases (i.e. the teleportation probability decreases), the exposure of news feed pages decreases, and as a result, the exposure of other kinds of pages increases. In fact, the sum of the exposure values news feeds has nearly a perfect correlation with the teleportation probability.

F. OSN Design and Exposure

As discussed in section II, one of the primary factors that impacts exposure is the design of the implemented OSN system. In our main OSN model (discussed in section V), the design of the hypothetical OSN dictates that each user has a feed page, a profile page, a friend list page, and a wall page. We will refer to this design as OSN-1. To demonstrate how a change in the design impacts exposure, we introduce a second model, OSN-2. In OSN-2, each user has only a profile page and a news feed page. The news feed is the same as in OSN-1. Meanwhile, the profile page now contains all of the knowledge and outgoing hyperlinks that used to be stored on the friend list and wall pages. This knowledge is displayed in two columns: one for friendships, and one for wall posts. The first friendship is assigned a $priority$ value that is equal to the first wall post, and this trend continues down the page. To demonstrate the impact of this change, we conducted the same damping-factor experiment discussed in Section VI-E on OSN-2.

Table III shows the result of repeating the damping factor experiemnt on OSN-2. As in OSN-1, the teleportation proba-

Damping Factor	Total Feed Exposure	Total Profile Exposure
0.15	84.976	14.999
0.50	50.027	49.997
0.85	15.015	84.997

Table III: The effect of the PageRank damping factor on page exposure by page type in OSN-2

bility ($1 - d$, where d is the damping factor) is a very strong predictor of the total exposure of all news feed pages. The difference is that, without an empty profile page "stealing" some of the total exposure, all of the remaining exposure is going to a page that contains knowledge. Therefore, knowledge appearing on non-news feed pages will have more exposure than it did in OSN-1.

VII. Related Work

Various access control models have been proposed for OSNs in order to allow users specify policies which protect the privacy of their information. In particular, relationship-based access control policies specify authorizations based on relationship between owner and accessor in the social network. Metrics such as type and distance of friendship [3, 9], various topology-based constraints [6], and more expressive ontologies [2, 12] have been discussed in the literature. While exposure is influenced by access control policies, it is also significantly driven by other design aspects of an OSN as well as user behavior. We argue that exposure can be a metric that can further enhance user privacy in OSN and complement existing access control policy models.

Mondal et al. introduce the notion of exposure control as an alternative to access control [13]. They define exposure for a piece of information as the *set of principals* who we expect to eventually learn about it. They then suggest that OSNs can employ item popularity algorithms [17] in order to inform their users about the exposure of their items and allow them to fine-tune their sharing policies. Despite similarity in name, our notion of exposure has concrete differences with Mondal et al.'s work. Looking at it from the perspective of the audience for a piece of information, Mondal et al.'s exposure is basically a subset of authorized set of users who will know about the item. In other words, it's a binary notion: you are either exposed or are not exposed to an item. However, we define exposure as the *chance* of a user getting exposed to the item. Therefore, our notion of exposure is probabilistic look at the authorized individuals for a piece of item, a more fine-grained perspective in a way. Moreover, while for assessing Mondal et al's exposure you would need access to actual OSN log's on user-item accesses, our notion of exposure can be calculated based on a theoretic model of an OSN. The latter has clear advantage for change-impact analysis tasks and tend to be more accurate if the theoretical model is captured precisely. The authors of [11] propose a similar approach to exposure as taken in this paper, in which they compare exposures in a system with and without news feed. In comparison, we propose a formal model of exposure in OSNs which provides a more efficient approach to calculating exposure. The formal model is easily extensible to incorporate other page importance algorithms beyond PageRank. Furthermore, we model exposure of a user and conduct experiment on a Facebook dataset that can demonstrate employing of exposure in OSNs better.

Liu and Terzi [10] have proposed a privacy score metric for users of OSNs which is based on sensitivity and visibility of users' information. However, they only consider the structure of the friendship network and visibility according to the (relationship-based) privacy settings. The key distinction in our approach is to consider structure of OSN's user interface and how it exposes users to various information, as opposed to measurement solely based on existing authorization policies.

Researchers in HCI privacy community have looked into how users' sharing behavior is impacted if they receive feedback on that. It should be noted exposure in this context simply refers to who is authorized to access rather than a standalone metric dependent on system design as in our work. Tsai et. al. [18] investigated user response to being given feedback about when their location information was shared. They found that when users were provided feedback, they felt overall more comfortable with their level of privacy and had fewer privacy concerns. Schlegel et. al. [16] devised a system to provide real-time feedback to users, displaying a set of eyes which grow larger as more location requests are granted. Hoyle et. al. [7] design a similar mobile app, presenting an "avatar" of the user. As more people are granted access to the user's data, the clothing of the avatar changes to reflect that they are now more exposed. Patil et. al. [15] conducted a user experience study in which they found that when users are provided immediate feedback about disclosure of their information, they are more likely to feel as though they have "over-shared" that information. The authors propose methods to make disclosures more actionable, or to delay feedback to avoid a "knee-jerk" response from the user.

Researchers have studied how users allocate their attention and interact in OSNs [1, 19, 20]. Backstrom et al. propose a new *attention measure* for analyzing social network of OSN users based on how they allocate their attention to different users [1]. The authors consider communication and viewing actions as attention modalities. Using a complete activity log of Facebook users they analyze how users' attention is allocated differently between communication and observation, and how they are affected by other factors such as age and gender. Wilson et al. propose to use actual user interaction within an OSN (e.g., posting on somebody's wall) as an indicator of social connection between users instead of relying solely on friendship graphs [20]. They show that an overlay interaction network is more fruitful than relationship network to be used in algorithms that rely on social network structure, such as for socially-attested messaging or for detecting Sybil (fake) identities in OSNs. Both notions of attention and interaction as discussed rely on a posteriori analysis of users' behavior in an OSN. Our focus in this paper is on how the design of an OSN could shape users' exposure to information, i.e., to measure the extent of exposure in an a priori fashion without

having access to an OSN's activity log.

We have adopted the PageRank algorithm [14] in this work in order to calculate page exposures in an OSN. PageRank and other similarly well-known methods such as HITS [8] are primarily used for ranking web search results by calculating importance of web pages.

VIII. Conclusions

In this paper, we defined the concept of exposure in OSNs at three distinct levels: page exposure, knowledge exposure, and user exposure. Each kind of exposure specifies the probability that the item in question will be accessed. We proposed a methodology for measuring these exposure values in a general sense that can be applied to various different real-world OSN systems. Finally, we experimented on a real-world Facebook dataset of friendships and wall posts, and showed that our calculations can capture the impact of various factors including knowledge and user characteristics, observer behavior and OSN design. As future work, we plan to investigate computing knowledge exposure in presence of fine-grained access control (we simply assumed global access in this work), modeling exposure of complex knowledge statements, and quantifying/managing privacy according to exposure.

References

[1] Lars Backstrom, Eytan Bakshy, Jon M. Kleinberg, Thomas M. Lento, and Itamar Rosenn. "Center of Attention: How Facebook Users Allocate Attention across Friends". In: *ICWSM*. Vol. 11. 2011, p. 23.

[2] Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. "A Semantic Web Based Framework for Social Network Access Control". In: *Proc. 14th ACM Symposium on Access Control Models and Technologies*. ACM, 2009, pp. 177–186.

[3] Barbara Carminati, Elena Ferrari, and Andrea Perego. "Rule-Based Access Control for Social Networks". In: *Proc. OTM 2006 Workshops (On the Move to Meaningful Internet Systems)*. Ed. by Robert Meersman, Zahir Tari, and Pilar Herrero. Vol. 4278. LNCS. Springer, Oct. 2006, pp. 1734–1744.

[4] L. Da F. Costa, F. a. Rodrigues, G. Travieso, and P. R. Villas Boas. "Characterization of Complex Networks: A Survey of Measurements". In: *Advances in Physics* 56.1 (Jan. 2007), pp. 167–242. ISSN: 0001-8732.

[5] Gabor Csardi and Tamas Nepusz. "The Igraph Software Package for Complex Network Research". In: *InterJournal* Complex Systems (2006), p. 1695.

[6] Philip W.L. Fong and Ida Siahaan. "Relationship-Based Access Control Policies and Their Policy Languages". In: *Proc. 16th ACM Symposium on Access Control Models and Technologies*. SACMAT '11. Innsbruck, Austria: ACM, 2011, pp. 51–60.

[7] Roberto Hoyle, Sameer Patil, Dejanae White, Jerald Dawson, Paul Whalen, and Apu Kapadia. "Attire: Conveying Information Exposure Through Avatar Apparel". In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work Companion*. CSCW '13. New York, NY, USA: ACM, 2013, pp. 19–22.

[8] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. "The Web as a Graph: Measurements, Models, and Methods". In: *Proceedings of the International Computing and Combinatorics Conference (COCOON'99)*. Ed. by Takano Asano, Hideki Imai, D. T. Lee, Shin-ichi Nakano, and Takeshi Tokuyama. Vol. 1627. Lecture Notes in Computer Science. Tokyo, Japan, Springer Berlin Heidelberg, June 1999, pp. 1–17.

[9] S. R. Kruk. "FOAF-Realm: Control Your Friends Access to the Resource". In: Workshop on Friend of a Friend, Social Networking and the Semantic Web. 2004.

[10] Kun Liu and Evimaria Terzi. "A Framework for Computing the Privacy Scores of Users in Online Social Networks". In: *ACM Trans. Knowl. Discov. Data* 5.1 (Dec. 2010), 6:1–6:30. ISSN: 1556-4681.

[11] Amirreza Masoumzadeh and Andrew Cortese. "Towards Measuring Knowledge Exposure in Online Social Networks". In: Workshop on Privacy in Collaborative & Social Computing (PiCSoC 2016). 2016, pp. 522–529.

[12] Amirreza Masoumzadeh and James Joshi. "OSNAC: An Ontology-Based Access Control Model for Social Networking Systems". In: *Proc. 2nd IEEE Int'l Conference on Information Privacy, Security, Risk and Trust (PASSAT 2010)*. Minneapolis, MN, USA, Aug. 2010, pp. 751–759.

[13] Mainack Mondal, Peter Druschel, Krishna P. Gummadi, and Alan Mislove. "Beyond Access Control: Managing Online Privacy via Exposure". In: *USEC 2014*. Internet Society, 2014.

[14] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. 1999.

[15] Sameer Patil, Roman Schlegel, Apu Kapadia, and Adam J. Lee. "Reflection or Action?: How Feedback and Control Affect Location Sharing Decisions". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. New York, NY, USA: ACM, 2014, pp. 101–110.

[16] Roman Schlegel, Apu Kapadia, and Adam J. Lee. "Eyeing Your Exposure: Quantifying and Controlling Information Sharing for Improved Privacy". In: *Proceedings of the Seventh Symposium on Usable Privacy and Security*. SOUPS '11. New York, NY, USA: ACM, 2011, 14:1–14:14.

[17] Gabor Szabo and Bernardo A. Huberman. "Predicting the Popularity of Online Content". In: *Commun. ACM* 53.8 (Aug. 2010), pp. 80–88. ISSN: 0001-0782.

[18] Janice Y. Tsai, Patrick Kelley, Paul Drielsma, Lorrie Faith Cranor, Jason Hong, and Norman Sadeh. "Who's

Viewed You?: The Impact of Feedback in a Mobile Location-Sharing Application". In: CHI '09. Boston, MA, USA: ACM, 2009, pp. 2003–2012.

[19] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. "On the Evolution of User Interaction in Facebook". In: *Proceedings of the 2nd ACM Workshop on Online Social Networks*. WOSN '09. New York, NY, USA: ACM, 2009, pp. 37–42.

[20] Christo Wilson, Alessandra Sala, Krishna P. N. Puttaswamy, and Ben Y. Zhao. "Beyond Social Graphs: User Interactions in Online Social Networks and Their Implications". In: *ACM Trans. Web* 6.4 (Nov. 2012), 17:1–17:31. ISSN: 1559-1131.

APPENDIX A
DETAILS OF IMPLEMENTED OSN MODEL

A. Knowledge Graph

Our implemented knowledge graph consists of user nodes and two kinds of edges: friendships, and wall posts. The friendship relation $R_{friend} \subseteq U \times U$ contains the two pairs $\langle u_i, u_j \rangle$ and $\langle u_j, u_i \rangle$ whenever u_1 and u_2 are friends. The wall post relation R_{post} has a pair $\langle u_i, u_j \rangle$ for each time that u_i has posted on u_j's wall (note that a user may post on their own wall). Therefore, the knowledge graph is defined as $G_K(U, R_{friend} \cup R_{post})$. Each edge $e \in (R_{friend} \cup R_{post})$ is assigned a timestamp, based on the time that the friendship or wall post was created. We denote this as $time(e)$.

B. Navigation Graph

We base the structure of our navigation graph on a simplified version of Facebook (see section V-A). Each user has 4 pages associated with them: a news feed, a profile, a friend list, and a wall. There are hyperlinks from the news feed to each of the profile, friend list, and wall. There are also a hyperlink for each 2-permutation of those three pages. Suppose that for each user u_x, the vertices n_x, p_x, f_x, and w_x are that user's news feed, profile, friend list, and wall, respectively. Then our navigation graph is constructed as follows:

1) $P = \bigcup_{u_x \in U} \{n_x, p_x, f_x, w_x\}$
2) $H = \bigcup_{u_x \in U} \{\langle n_x, p_x \rangle, \langle p_x, f_x \rangle, \langle f_x, p_x \rangle, \langle p_x, w_x \rangle, \langle w_x, p_x \rangle, \langle f_x, w_x \rangle, \langle w_x, f_x \rangle\}$
3) For each $u_x \in U$:
   a) $contents(f_x) = \{friend(u_x, v) \mid \langle u_x, v \rangle \in R_{friend}\}$
   b) $contents(w_x) = \{post(v, u_x) \mid \langle v, u_x \rangle \in R_{post}\}$
   c) $contents(n_x) = \{k \in \mathcal{K} \mid \exists v \in \int(k). \langle u_x, v \rangle \in R_{friend} \wedge u_x \notin \int(k)\}$
4) For each knowledge statement $friend(u_x, u_y)$ on a user u_x's friend list page f_x, add a hyperlink $\langle f_x, p_y \rangle$ to H, where p_y is the profile page of user u_y.
5) For each knowledge statement $post(u_y, u_x)$ on a user u_x's wall page w_x, add a hyperlink $\langle w_x, p_y \rangle$ to H, where p_y is the profile page for user u_y
6) For each knowledge statement k on a user u_x's news feed page n_x:
   a) If $k = friend(u_y, u_z)$, then add hyperlinks $\langle n_x, f_y \rangle$ and $\langle n_x, f_z \rangle$ to H, where f_y and f_z are the friend list pages for u_y and u_z, respectively.
   b) If $k = post(u_y, u_z)$, then add hyperlinks $\langle n_x, p_y \rangle$ and $\langle n_x, w_z \rangle$ to H, where p_y is the profile for user u_y and w_z is the wall for user u_z.
7) Assign a position to each knowledge statement on each page as follows: Suppose $p \in P$ and $contents(p) = \{k_1, k_2, ...k_n\}$. We sort the statements in non-increasing order by timestamp. The statement with greatest timestamp is assigned position 0, and we then increment the position by 1 for each subsequent timestamp. The result is that each statement has a position $0 \leq position(k_i, p) \leq n - 1$. The item with position 0 is the "most prominent" item on the page (i.e. the one that is most likely to be seen).
8) Assign the $priority$ value to each knowledge statement based on position: $priority(k_i, p) = \frac{n - position(k_i, p)}{n*(n+1)/2}$. Note that $\sum_{k_i \in contents(p)} priority(k_i, p) = 1$
9) Assign a weight to each edge as follows:
   a) If e is an edge added in item (2), then $weight(e) = 0.1$
   b) Otherwise, e is an edge $\langle p_1, p_2 \rangle$ added in item (6), which means e was added due to the presence of some knowledge statement $k_1 \in contents(p_1)$. Therefore, $weight(e) = priority(k_1, p_1)$

An intuitive explanation of the process is as follows. Each user has a profile page, a news feed page, a friend list page, and a wall page. There is a link from the news feed (which acts as a homepage) to the profile, and hyperlinks between each pair of the other three pages. The friend list page for a particular user contains a knowledge statement for each bidirectional friendship edge incident to that user in the knowledge graph. The wall page for a particular user contains a knowledge statement for each incoming wall post edge incident to that user in the knowledge graph. The news feed page for a user contains a knowledge statement for each edge incident in the knowledge graph to a friend of that user, excluding edges incident to the user themselves. Knowledge on each page is sorted in non-increasing order of timestamp. Each friend list page has an outgoing hyperlink for each friendship contained on that page, which leads to that friend's profile page. The wall post page has an outgoing hyperlink for each post on the page, which leads to the profile of the author of that post. For each knowledge statement on a news feed page, there are two outgoing hyperlinks: If the knowledge statement is a friendship, then the hyperlinks lead to the profiles of the two users involved in that friendship. If the knowledge statement is a wall post, then there is one hyperlink leading to the author's profile page, and another leading to the receiver's wall page.