Visualizing Privacy Implications of Access Control Policies in Social Network Systems

Mohd Anwar Philip W. L. Fong
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
{manwar, pwlfong}@ucalgary.ca

Xue-Dong Yang Howard Hamilton
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada
{yang, hamilton}@cs.uregina.ca

Abstract

We hypothesize that, in a Facebook-style social network system, proper visualization of one's extended neighborhood could help the user understand the privacy implications of her access control policies. However, an unrestricted view of one's extended neighborhood may compromise the privacy of others. To address this dilemma, we propose a privacy-enhanced visualization tool, which approximates the extended neighborhood of a user in such a way that policy assessment can still be conducted in a meaningful manner, while the privacy of other users is preserved.

1 Introduction

One of the main purposes of privacy preservation is impression management [9, 16]. This is particularly true in the context of social network systems. A profile owner selectively grants a profile viewer access to her profile items in accordance with the impression she wants to convey. For example, say Jill is a friend of Alice, and Bob is a friend of Jill. For proper impression management, Alice may grant Jill, but not Bob, access to her sorority photo album. To check whether her policy allows her to convey the desired impression, Alice may want to look at her profile from the lenses of Bob and Jill, to find out what Bob as well as Jill can see. In our everyday life, we look into a mirror to get a sense of what others see when they look at us. We use the term ***reflective policy assessment*** to refer to this process of assuming the position of a potential accessor for the sake of assessing the privacy implications of access control policies.

Authorization in a social network system is primarily based on the topology of the social graph, which is co-constructed by all the users of the system. It is therefore difficult for a user to mentally keep track of the topology of her constantly changing social network. Furthermore, one's needs for privacy is constantly changing, requiring a user to constantly perform policy assessment. As a result, reflective policy assessment is a nontrivial undertaking. Tool support is definitely desirable.

Unfortunately, a privacy dilemma is inherent in reflective policy assessment. To assess policies reflectively, a user must begin with identifying a potential accessor who is of interest to her. This, however, could lead to breaching the privacy of the potential accessor, as the latter may not want her identity to be disclosed to the user conducting the policy assessment. Suppose the running example is situated in Facebook. If Bob adopts a privacy setting that allows his identity to be revealed only to friends but not friends of friends, then Alice will not be able to conduct reflective policy assessment against Bob without breaching his privacy.

This privacy dilemma is not specific to just Facebook. Fong et al. proposed an access control model to delineate the design space of privacy preservation mechanisms in Facebook-style social network systems [8]. In this model, policies such as "only friends" and "friends of friends" are but examples of more general ***topology-based policies***, whereby accessibility is determined by the present topology of the social graph. For example, Alice may adopt the policy that grants access to her sorority photo album only if the accessor shares three common friends with her. With these policies, it would even be more important to have access to one's extended neighborhood in addition to her immediate friends for the purpose of policy assessment.

This dilemma is rooted in the asymmetric nature of trust. In the process of reflective policy assessment, a resource owner (e.g., Alice) conceptualizes the level of trust she is willing to invest in a potential accessor (e.g., Bob). Yet, this endeavor is possible only if the identity of the potential accessor is known to the resource owner, the feasibility of which may not always be possible because the potential accessor may not trust the resource owner.

This paper is about the design of a privacy enhanced visualization tool for Facebook-style social network systems (FSNSs) to facilitate reflective policy assessment while preserving the privacy of potential accessors. Our

contributions are the following:

1. We introduced the notion of reflective policy assessment, which helps a user assess the privacy implications of her policies by positioning herself as a potential accessor. We also discovered and addressed an inherent privacy dilemma of reflective policy assessment.

2. We transformed the concept of reflective policy assessment into a concrete visualization tool for policy assessment, which is user-centric and intuitive. Since this tool would not require the knowledge of access control policies of all the users of the system, it can be implemented on the client side (e.g., as a third-party Facebook application)

3. At the core of our visualization technique is a visual representation of a user's extended neighborhood. We established graph-theoretic properties common to the social graphs of FSNSs. Based on these properties, we devised an algorithm to generate a surrogate of a user's extended neighborhood. This surrogate can be examined for reflective policy assessment without violating the privacy of other users.

The organization of this paper is as follows. Section 2 describes an access control model for FSNSs. In Section 3, we present the main idea of assessing policies through visualization. In section 4, we present an algorithm for generating a surrogate of a user's extended neighborhood for policy assessment. Section 5 presents some open questions on how to evaluate the proposed visualization tool. Section 6 surveys related literature, and Section 7 describes conclusion and future work.

2 An Access Control Model for SNSs

In this work, we study reflective policy assessment for a family of FSNSs [8], of which Facebook is a notable member. This section briefly outlines the access control model shared by this family of social network systems so as to anchor the discussion in the sequel. Formal details of this model can be found in [8].

Profile and Profile Items An FSNS allows each user to construct a representation of his- or herself in the form of a ***profile***. A profile displays such ***profile items*** as personal information, multimedia contents, activity logs, or other user-authored contents. Users may grant one another access to their profile items.

Search Listings Access to profile items is authorized in two stages. In ***Stage I***, the accessor must ***reach*** the ***search listing*** of the profile owner. Then in ***Stage II***, the accessor requests access to the profile, and profile items are selectively displayed. The search listing of a user could be seen as a "capability" [6, 15] of the user in the system, through which access is mediated. There are two means by which a profile can be reached in Stage I: ***global name search*** and ***social graph traversal***.

Global Name Search The first means to reach a search listing is to conduct a global name search. A successful search would produce for the accessor the search listing of the target user. A profile owner may specify a ***search policy*** to allow only a subset of users to be able to reach her search listing through a global name search.

Social Graph Traversal A second means to reach a search listing is by traversing the ***social graph***. Users can articulate their relationships with one another through the construction of ***friend lists***. Every user may specify a set of other users as her ***friends***. This induces a simple graph in which users are nodes and relationships are edges. A user may traverse this graph by examining the friend lists of other users. More specifically, the friend list of a user is essentially the set of search listings of her friends. A user may restrict traversal by specifying a ***traversal policy***, which specifies the set of users who are allowed to examine her friend list once her search listing is reached.

Profile Access Once the search listing of a profile owner is reached, the accessor may choose to access the profile, and thereby, initiate Stage II of authorization. Since a profile owner may assign an ***access policy*** to each profile item, not every accessor sees the same profile items when a profile is accessed.

Friendship Articulation Articulating friendship involves a consent protocol, whereby users interact with one another via a fixed set of ***communication primitives*** (e.g., friendship invitation, accepting an invitation, etc). Once a mutual consent is reached, that friendship is recognized by the FSNS. When a sender initiates a communication primitive against a receiver, the the search listing of the latter must be reached before the communication primitive can be initiated. A user can prevent others from initiating a certain communication primitive against her by assigning a ***communication policy*** to that primitive.

Topology-Based Policies User activities are controlled by user-specified policies (i.e., search, traversal, access and communication policies). Each FSNS defines a fixed policy vocabulary for users to choose from when they are to identify sets of privileged users. Since there is no global name space of users, these predefined policies identify user sets indirectly in terms of the topology of the social graph. For example, one may specify that a certain profile item is accessible only by "friends of friends". Sample policies are shown in Fig. 1.

Policy predicate: *When is access allowed*
distance$_k$: distance between owner and accessor is no more than k
clique$_k$: owner and accessor belong to the same k-clique (i.e., they belong to the same close-knit group)
common-friends$_k$: owner and accessor share k common friends (i.e., accessor is a known quantity)

Figure 1: A sample of topology-based policies

3 A Privacy-enhanced Visualization Technique

A Mirror-based Visualization Technique Our visualization technique seeks to provide a mirror-like affordance to users in FSNSs. To create a desired impression, we repeatedly look into the mirror and adjust our getup until we are satisfied. A mirror allows us to see what others see when they look at us. The process of formulating access control policies is similar to what it takes to create a desired look. With our ever changing social network and ever changing desire for privacy, a user needs to repeatedly assess and adjust their policies. We propose a mirror-like tool to help a user visualize what others see when they look at her.

Our proposed visualization tool offers the following functionalities to a profile owner.

1. The tool provides a visual representation of an extended neighborhood of a profile owner in the social graph. The profile owner may specify the size of her extended neighborhood.

2. This tool allows the profile owner to point to any user in the extended neighborhood as a potential accessor of her profile. This action signals to the tool that the profile owner intends to position herself as the selected user and examine her profile from the vintage point of that user.

3. The tool displays a succinct representation of the profile, as seen from the eyes of the potential accessor.

This tool contributes to policy assessment in the following ways:

What-if Analysis: It allows a profile owner to perform "what-if" analysis on her access policies. More specifically, it allows her to assess the adequacy of her access policies in concrete access scenarios, and to evaluate the effect of adopting these policies when her extended neighborhood possess a certain topological structure.

Targeted Effort: As the tool displays how other users are topologically related to a profile owner, it helps her identify topologically interesting nodes in the extended neighborhood, thereby allowing her to properly target her policy assessment effort. For example, in Figure 3, the node *FOF* a topologically interesting node when the profile owner *Me* attempts to assess a "friends of friends" policy.

Visualizing without Breaching Privacy The visual representation of the extended neighborhood must be generated in such a way that the privacy of a potential accessor is preserved. To see this, recall in Section 2 that not every potential accessor is reachable from the profile owner, even if there is a path between them. This scenario may arise if at least one of the intermediate nodes along the path has a traversal policy that prevents the profile owner from examining the friend list of that intermediate node. Consequently, depicting the extended neighborhood in full accuracy compromises privacy. Fortunately, an accurate rendering of the extended neighborhood is not necessary for reflective policy assessment. Rather, an approximate rendering that exhibits the topology typical of social networks should suffice. Therefore our approach is to approximate the unreachable region of the extended neighborhood by generating synthetic nodes and edges in a way that preserves such properties of social networks as power law vertex degree distribution [7] and small-world characteristic [14]. Details of the graph generation algorithm can be found in Section 4.

Mockup In Figure 2, we show a mockup of our visualization tool. Here, the black node is the profile owner (*Me*). White nodes (e.g., *Jay*) and solid edges (e.g., *Jay-Doe*) depict the interior of the profile owner's reachable region in the social graph. Grey nodes (e.g., *Doe*) mark the boundary (inclusive) of the reachable region. The dotted nodes and dotted edges are generated to approximate the unreachable region of the policy owner. As the profile owner selects a potential accessor by pointing her cursor over the latter, an information box pops up. The information box displays what profile items of the profile owner that the selected user can see as a result of the profile owner's current policies. Specifically, the information box displays three categories of information: (i) the profile items of the profile owner that the selected user can access, (ii) a list of the profile owner's friends that the selected user can reach through the profile owner, and (iii) a list of communication primitives that the selected user can initiate against the profile owner.

Section (i) of the information box is a "reflection" of the profile under assessment. This section supports the assessment of access policies. Section (ii) of the information box supports the assessment of traversal policies. A user's traversal policy has privacy implications not only on the user, but also on her friends. Specifically, an overly relaxed traversal policy will expose one's friends to un-

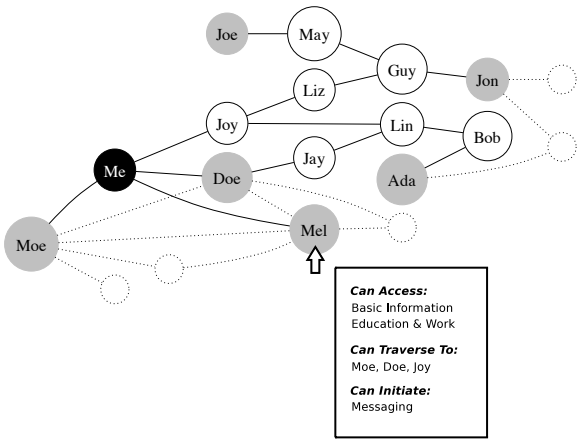

Figure 2: Visualization for reflective policy assessment.

wanted accessors. In a similar vein, section (iii) of the information box supports the assessment of one's communication policies.

As an example, in Figure 2, when the profile owner *Me* points to *Mel*, the tool displays the following: (i) *Mel* can access two profile items of the profile owner: "*Basic Information*" and "*Education and Work*"; (ii) *Mel* can reach *Moe*, *Doe* and *Joe* through *Me*; (iii) *Mel* can send a message to *Me*, but cannot invite *Me* to be a friend.

Assessing Topology-based Policies A critical reader may question why it is necessary to consider unreachable nodes in the process of reflective policy assessment. We illustrate the utility of this practice by giving some examples. Consider the extended neighborhood of user *Me* in Figure 3. We show how various topology-based policies need to be evaluated from the vintage point of unreachable nodes.

distance$_k$: Suppose user *Me* adopts distance$_5$ as the access policy for her wedding video, thereby granting access to anyone within a distance of five. Let us suppose further that *Jon* is at distance four, whose traversal policy does not allow *Me* to traverse to *Jon*'s friends, including, for example, *D5*. However, user *Me* may precisely want to examine her profile from the perspective of *D5*, which is at distance five from *Me*, in order to evaluate her distance$_5$ policy.

common-friends$_k$: Suppose the profile owner *Me* specifies common-friends$_3$ as the access policy of her "Contact Information", so that the latter is accessible to those users sharing three common friends with *Me*. According to Figure 3, users *Me* and *CF2* have only two common friends (*Moe* and *Mel*). Even though it is to the interest of user *Me* to assess her policies reflectively from node *CF2*, the prohibitive traversal policies of *Moe* and *Mel* may render this as a breach of privacy.

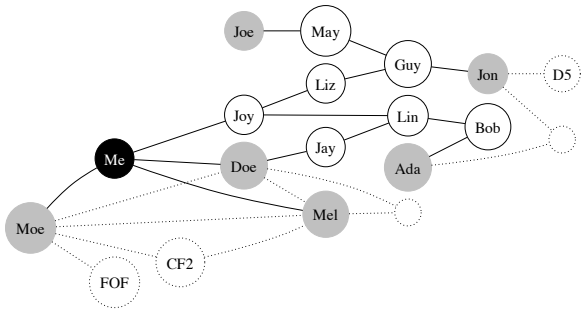

Figure 3: The extended neighborhood of a profile owner.

clique$_k$: Suppose user *Me* specifies an access policy, clique$_4$, for her "Status". That is, access is granted to her friends who belong to the same 4-clique as she does. In Figure 3, users *Me*, *Moe*, *Doe* and *Mel* belong to the same 4-clique. Even though user *Me* needs to confirm that *Moe* and *Doe*, *Doe* and *Mel*, and *Mel* and *Moe* are friends in order to assess her clique$_4$ policy, the traversal policies of *Doe*, *Moe* and *Mel* do not allow the *Me* to discover these relationships.

4 Constructing a Social Graph for Policy Assessment

This section describes an algorithm for generating a visual representation of the social graph for policy assessment. We set the stage by describing some graph-theoretic properties of FSNS social graphs (Section 4.1), and then apply the properties to devise the algorithm and establish its correctness (Section 4.2).

4.1 Properties of Social Graphs

A node v is u-traversable if the traversal policy of v allows u to examine the friend list of v. If there is a uv-path $uv_1 \dots v_n v$ in the social graph such that every v_i is u-traversable, then we say user v is ***u*-reachable**. Otherwise, v is ***u*-unreachable**. A u-reachable node is an ***u*-interior node** if it is u-traversable, and a ***u*-fringe node** otherwise. An edge is ***u*-visible** if one of its ends is a u-interior node, otherwise it is ***u*-hidden**. The node u in the above definitions is called the ***origin***. We drop the "u-" prefix when the origin is clear from the context.

Property 1. *Given an origin, every neighbor of an interior node is reachable, and thus, no hidden edge can have an interior node as an end.*

Property 2. *Suppose an origin is given. By definition, at least one end of each visible edge is an interior node. Therefore, no visible edge can join two fringe nodes.*

4.2 A Graph Generation Algorithm

We present an algorithm for generating a graph to approximate an extended neighborhood of a user u in the social graph. The generated graph is composed of two regions. The first region is made up of the reachable nodes and the visible edges. The second region is randomly generated to approximate the unreachable nodes and the hidden edges of the social graph. To ensure that the randomly generated region reflects the topological structure of a typical social graph, we employ the R-MAT [5] algorithm, which randomly generates graphs exhibiting statistical properties of a real-world social network. (Other appropriate graph generation algorithms can also be used.)

algorithm $A(u)$

1. Using u as the origin, construct a graph consisting of all reachable nodes and visible edges.
2. Temporarily remove all interior nodes and visible edges, leaving only the fringe nodes.
3. Add a desirable number of "synthetic nodes".
4. Use R-MAT to randomly generate a desirable number of "synthetic edges".
5. Add back the interior nodes and visible edges removed in step 2, and return the resulting graph.

The correctness of algorithm A can be justified as follows. By **Property 1**, no hidden edge can have an interior node as an end, and thus interior nodes can be removed from consideration in Step 2. By **Property 2**, no visible edge can join two fringe nodes. Therefore, Step 4 starts with an empty graph, and thus the statistical properties of R-MAT (or other graph generation algorithms) is preserved.

Step 1 can be achieved by an elementary third-party Facebook application[1] that performs a breadth-first search. This means the algorithm can be executed on the client side. Algorithm A also has two parameters: the number of synthetic nodes and edges to be added into the graph.

5 Open Questions

Our proposal motivates a number of open questions.

To what extent does our visualization technique facilitate the assessment of access control policies in FSNSs? If a tool is effective in supporting policy assessment, we should observe that privacy-aware users tend to formulate a different set of policies after adopting the tool. An empirical user study will help us test if this is indeed the case for our visualization technique. Such a user study shall compare the policies formulated by the user in at least three configurations: (i) no visualization is available, (ii) mirror-based visualization with the rendering of reachable nodes only, (ii) mirror-based visualization with the rendering of both reachable and unreachable nodes.

How do we build a testbed to run the proposed user study? A deployed FSNS, such as Facebook, would have been a convenient environment to conduct the proposed user study. There are, however, two problems with this approach. First, not all topology-based policies are supported in Facebook. As a result, the effectiveness of reflective policy assessment against advanced topology-based policies cannot be gauged. Second, such a study will harvest information of users located in the reachable region of a participant. This setup thus requires consent from a population much larger than the participating group. Even if this aggressive experimental design is approved by the institutional research ethics committee, successfully obtaining consent from such a large population is not likely. We anticipate that the resolution of this problem will involve a clever design of a simulated environment that addresses these privacy challenges.

To what extent are the randomly generated graphs (Section 4.2) useful approximations of the unreachable region of one's extended neighborhood? We hypothesize that the graphs generated by algorithm A cover topologically interesting scenarios needed by the profile owner for conducting reflective policy assessment against unreachable nodes. Intuitively, repeated policy assessment on multiple generated graphs should increase the coverage of topologically interesting scenarios. A natural research question is thus the following: "*how many graphs does one need to generate in order to gain enough confidence on the policies under assessment?*"

6 Related Works

Assessing the security implications of access control policies traditionally lies in the domain of safety analysis [10, 13], or, more recently, security analysis [12, 11]. When the projection of security implications becomes a challenging computational problem, safety or security analyses are indispensable. While appreciating the scope and analytical rigor of such approaches, this paper seeks to address the *cognitive challenges* of users in the projection of the *privacy implications* of their access control policies. A visualization tool can reduce the cognitive load of users in policy assessment. It is also a better fit with the requirements of impression management.

Our proposed visualization technique supports impression management for a family of FSNSs. This family was defined by Fong et al. [8], who formally specify an access control model that delineates the design space of social network systems employing the same access control paradigm as Facebook. A distinctive feature of FSNSs is that no global name space is available for identifying

[1] For example, the third-party Facebook application TouchGraph performs a similar search.

users, and thus access control policies are specified in terms of the present topology of the social graph. This element of distributed access control causes policy assessment to be a nontrivial undertaking, thereby necessitating our visualization technique. Furthermore, Fong et al. formulated some policies that are purely based on topological information: e.g., Degree of Separation, Known Quantity, Clique, Trusted Referrers, etc.

A number of recent proposals attempt to advance beyond the access control mechanisms found in commercial social network systems. A notable example is that of Carminati et al., in which a decentralized social network system with relationship types, trust metrics and degree-of-separation policies is developed [2, 3, 1, 4]. An interesting research issue is to design tools that support reflective policy assessment in these next-generation social network systems.

7 Conclusion & Future Work

We anticipate that our visualization technique can reduce users' cognitive load in understanding the privacy implications of their access control policies in a FSNS. Specifically, this visualization technique helps a profile owner assess her policies by displaying how potential accessors are topologically related to her in an extended neighborhood, and allowing her to visually assess her policies via a mirror-like facility from the perspective of a potential accessor of her choice. This technique supports the reflective assessment of access, traversal and communication policies in FSNSs. We plan to conduct an empirical study to gauge the effectiveness of this visualization technique.

References

[1] Barbara Carminati and Elena Ferrari. Privacy-aware collaborative access control in web-based social networks. In *Proceedings of the 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DAS'08)*, volume 5094 of *LNCS*, pages 81–96, London, UK, July 2008. Springer.

[2] Barbara Carminati, Elena Ferrari, and Andrea Perego. Rule-based access control for social networks. In *Proceedings of the OTM 2006 Workshops*, volume 4278 of *LNCS*, pages 1734–1744, October 2006.

[3] Barbara Carminati, Elena Ferrari, and Andrea Perego. Private relationships in social networks. In *Proceedings of Workshops in Conjunction with the International Conference on Data Engineering – ICDE'07*, pages 163–171, Istanbul, Turkey, April 2007. Springer.

[4] Barbara Carminati, Elena Ferrari, and Andrea Perego. Enforcing access control in web-based social networks. *ACM Transactions on Information and System Security*, 2009. To appear.

[5] Deepayan Chakrabarti, Christos Faloutsos, and Yiping Zhan. Visualization of large networks with min-cut plots, A-plots and R-MAT. *International Journal of Human-Computer Studies*, 65:434–445, 2007.

[6] Jack B. Dennis and Earl C. van Horn. Programming semantics for multiprogrammed computations. *Communications of the ACM*, 9(3):143–155, March 1966.

[7] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251– 262, 1999.

[8] Philip W. L. Fong, Mohd Anwar, and Zhen Zhao. A privacy preservation model for Facebook-style social network systems. Technical Report 2009-926-05, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, April 2009. Submitted for review.

[9] E. Goffman. *The Presentation of Self in Everyday Life*. Anchor-Doubleday, New York, NY, 1961.

[10] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman. Protection in operating systems. *Communications of the ACM*, 19:461–471, 1976.

[11] N. Li and M. V. Tripunitara. Security analysis in role-based access control. In *the Ninth ACM Symposium on Access Control Models and Technologies (SACMAT 2004)*, pages 126–135, 2004.

[12] N. Li, W. H. Winsborough, and J. C. Mitchell. Beyond proof-of-compliance: Safety and availability analysis in trust management. In *IEEE Symposium on Security and Privacy*, pages 123–139, 2003.

[13] R. J. Lipton and L. Snyder. A linear time algorithm for deciding subject security. *Journal of the ACM*, 24:455–464, 1977.

[14] S. Milgram. The small world problem. *Psychology Today*, 1:60–67, 1967.

[15] Mark S. Miller, Ka-Ping Yee, and Jonathan Shapiro. Capability myths demolished. Technical Report SRL2003-02, System Research Lab, Department of Computer Science, The John Hopkins University, Baltimore, Maryland, USA, 2003.

[16] S. Patil and A. Kobsa. Privacy as impression management. Technical Report UCI-ISR-03-13, Institute for Software Research, University of California - Irvine, Irvine, CA, USA, December 2003.