Fine-Grained Access Control of Personal Data

ABSTRACT

The immensity and variety of personal information (e.g., profile, photo, and microblog) on social sites require access control policies tailored to individuals' privacy needs. Today such policies are still mainly specified *manually* by ordinary users, which is usually coarse-grained, tedious, and error-prone. This paper presents the design, implementation, and evaluation of an *automated* access control policy specification tool, xACCESS, that helps non-expert users effectively specify who should have access to which part of their data. A series of key features distinguish xACCESS from prior work: 1) it adopts a role-based access control model (instead of the conventional rule-based paradigm) to capture the implicit privacy/interest preference of social site users; 2) it employs a novel hybrid mining method to extract a set of semantically interpretable, functional "social roles", from static network structures and dynamic historical activities; 3) based on the identified social roles, confidentiality setting of personal data, and (optional and possibly inconsistent) predefined user-permission assignments, it recommends a set of high-quality privacy settings; 4) it allows user feedback in every phase of the process to further improve the quality of the suggested privacy policies. A comprehensive experimental evaluation is conducted over real social network and user study data to validate the efficacy of xACCESS.

1. INTRODUCTION

This is the era of social networking! Online social networks (OSNs) have become a de facto portal for hundreds of millions of Internet users. For example, FACEBOOK, one representative social network provider, claims that it enjoys over 350 million active users [2]. With the help of these social sites, users share information with their friends, participate in online activities, and get to know more new friends. The unprecedented immensity and diversity of personal information over social networks (e.g., it is estimated that over 3.5 billion pieces of content, including web links, news stories, blog posts, notes, and photo albums, are shared by FACEBOOK users each week), however, is far beyond the development of privacy control enforcement tools. Improper privacy control over personal information tends to lead to severe consequences [4, 1].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

So far, users still mainly rely on social network sites to provide privacy control to restrict data sharing with friends, corporate affiliates, or application developers. Nevertheless, the available controls are rather limited. For example, quite recently, FACEBOOK claimed that it launched the first platform for users to personalize their privacy setting, via manually specifying who (classified into three classes {*friend, friend of friend, everyone*}) should have access to which parts of their information. This scheme, however, suffers from several evident drawbacks: the rigid classification of relevant users into three groups is fairly coarse-grained. For example, for two users belonging to the same category, one may desire to assign different permissions. More flexible control could only be achieved through a manual *customized* setting, such as the "circle" concept in GOOGLE PLUS[1], which leads to the next problem; a full manual setting is usually tedious and error-prone. Consider that right now an average user has 130 friends on FACEBOOK, and the number of friends of friends is typically quadratic. With such a large pool of relevant users, it becomes a non-trivial task for ordinary users to effectively specify their privacy policies.

1.1 State of the Art

In both the database and security communities, intensive research efforts have been dedicated to protecting individuals' privacy in social network data publishing, a problem orthogonal to the scope of this work. In [5, 17, 25], it was shown to be possible to re-identify individuals in the published network data even if explicit identification information, e.g., name, affiliation, and address, has been masked. In [26, 7, 13, 15], countermeasures have been proposed against the re-identification attacks while the adversary possesses various background knowledge regarding the original network, e.g., degree, neighborhood structure, or subgraph in general.

Some initiative research works have recently recognized the importance of enforcing user-specified privacy control over personal information on online social networks. For example, xBOOK [22] attempts to enforce control of what third-party applications can do with the information they receive from social network sites, using an information flow model. PERSONA [6] hides user data with attribute-based encryption (ABE) schemes, allowing users to apply fine-grained, customized policies over who may view their data. Nevertheless, all these works focus on how to enforce user-specified control, with the assumption that the privacy policies are completely and clearly specified.

In a recent work [10], Fang and LeFevre proposed PRIVACY WIZARDS, a *semi-automated* privacy setting recommendation tool that extracts a set of permission assignment rules, based on an active learning paradigm, *uncertainty sampling*. This *rule-based* model, however, suffers from two major drawbacks. First, it implicitly re-

[1]https//plus.google.com

lies on a "lazy user" assumption (users are fully capable of, yet not willing to manually specify the policies), and requires accurate user input on a set of highly ambiguous assignments that, however, are typically the most difficult spots for non-expert users. Second, the discovered permission assignment rules may lack semantical interpretation, and are thus difficult to understand by social site users, which severely limits its applicability.

1.2 Challenges and Contributions

This work presents the design, implementation, and evaluation of an access-control policy specification tool, xAccess, for social networking platform. To our best knowledge, this is the first *automated* framework that helps ordinary social network users effectively specify customized privacy polices for their personal data. For a social site user, xAccess suggests a set of high-quality permission assignments for all relevant users. The suggested assignments are semantically meaningful and understandable from the perspective of social activity in that they reflect functional, fine-grained, latent "social roles", e.g., a friend with certain common interest, a co-worker on certain project, etc.

The fundamental assumption of xAccess is the existence of a set of fine-grained, latent social roles that capture the social functions of the users relevant to the target individual. This concept is in spirit similar to the "role" in role-based access control (RBAC) paradigm [11]. We argue that it makes much more sense to reason about user-permission assignment based on social roles instead of social relationships: first, the social relationship could be fairly vague, and deviates from its semantic meaning, e.g., "friend" may actually mean relative or co-worker; second, an individual may carry multiple social roles, and thus should have the union of permissions associated with these social roles, which could not be captured by a single social relationship.

Unlike conventional RBAC frameworks wherein roles are typically captured by auxiliary structures, e.g., enterprise managerial hierarchies, a social network is, however, inherently "flat", in the sense that no hierarchical structures are available to define social roles. To address this challenge, we introduce a novel hybrid mining method that combines graph mining (over social network structure) and event mining (over historical social activities of users). Based on the identified social roles, confidentiality setting of personal information, and predefined user-permission assignments (optional and may contain inconsistency), xAccess matches relevant users to their potential social roles, and social roles to their associated permissions. Moreover, xAccess allows user feedback in every phase of the process to further improve the quality of suggested user-permission assignments. The main framework of xAccess is illustrated in Figure 1.

Our contributions can be summarized as follows. First, we highlight and articulate the problem of helping ordinary social site users understand and specify privacy control policies over their personal data. Second, we propose a novel hybrid mining method that discovers semantically meaningful social roles from both social network structure data and historical activity data. Third, based on the social role mining method, we construct a fully *automated* access control policy specification tool, xAccess, for social networking platforms. Finally, we validate the analytical models and the efficacy of xAccess over real social network and user study data.

The remainder of the paper will be organized as follows. Section 2 introduces the fundamental concepts of xAccess. Section 3 and 4 detail the design and implementation of xAccess, with possible extensions discussed in Section 5. An empirical evaluation of the proposed solution is presented in Section 6. The paper is concluded in Section 7.

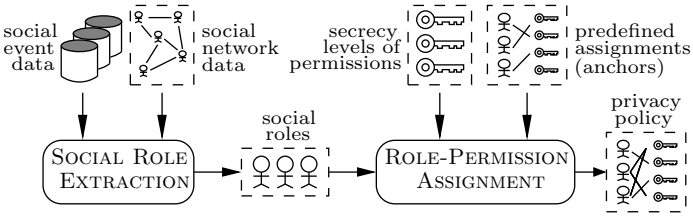

Figure 1: Framework of xAccess.

2. MODELS AND CONCEPTS

A social network is modeled as a graph $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ and $\mathcal{E}$ representing the set of users and their social relationships, respectively. In this paper, we focus on deriving the access control policies for a specific user $v \in \mathcal{V}$, and thus introduce a variant of this definition:

Definition 1 (Viewpoint Network). *We define the h-hop viewpoint network of user v as the subgraph $G_h^v = (\mathcal{V}_h^v, \mathcal{E}_h^v)$ of G which consists of the users within h hops of v (including v) in G and the social relationships among them.*

By limiting h to a small number (typically 2 or 3), we focus on the set of users socially local to v, which is sufficient for most social sites. In the following, when the context is clear, we omit the referred target user v in the notations.

A permission is an access privilege to certain personal information. Its concrete definition depends on social sites and applications (e.g., access one's photo album or comment on one's microblog), and is orthogonal to the scope of this work. We assume a set of predefined permissions. The permissions may be structurally related; later, we will discuss how such structure impacts the setting of privacy policy.

The problem of specifying access control policies for user v is essentially equivalent to identifying a proper permission assignment ϕ_u for each user $u \in \mathcal{V}_h^v \setminus \{v\}$ (with respect to v's personal information). Henceforth, we refer to u and v as the source and target, respectively. The focus of this work is to alleviate the burden of social site users by suggesting *informative*, *personalized* policy settings. Instead of relying solely on static information (e.g., hop distance or relationship type) as currently adopted by most social sites, we construct the recommendation framework atop the notion of *social role*.

Definition 2 (Social Role). *A social role [24] is a set of connected behaviors as conceptualized by individuals with a given social connection to the target individual.*

A social role specifies the expected social functions, thereby implying the expected access rights of individuals with a given social connection to the target user, which makes it an ideal bridge between users and permissions. To capture the social role of the source u (relative to the target v), one needs to consider (i) u's social connection to the target v as reflected in the social network structure, and (ii) u's social behaviors as reflected in the social activities in which u and v participate. Motivated by this observation, we propose a novel hybrid mining method that extracts a set of semantically interpretable roles from social network and social activity data. To our best knowledge, this is the first in its category.

3. SOCIAL ROLE EXTRACTION

Next, we present our hybrid mining method that exploits both social network structure (for social connection) and historical social

activity (for social behavior), with details presented in Section 3.1 and 3.2, respectively.

3.1 Social Network Structure

The social connection between the source and the target may not be solely determined by their hop distance or their relationship type; rather, it involves all relevant users. As an example, consider three friends u_i, u_j, u_k of v, while u_i and u_k are also friends, which may indicate a stronger connection between u_i and v than u_j. We introduce the concept of *social proximity* to capture this notion, which measures the overall strength of a social connection.

Ideally, if two individuals share many common neighbors with close relationships, or they belong to a small and tight community, their social proximity would be high. A variety of measures have been proposed [23] to capture the notion of network proximity, including *Katz measure*, *hitting time*, and *escape probability*. In our implementation, we adopt the measure of *random walk with restart* (RWR), one of the most popular proximity metrics in graphs [18], which is empirically proved to perform the best in our experiments. Specifically, in a RWR, starting from node v, at each step, the walk moves to one of its current neighbors with probability proportional to the corresponding edge weight, or returns to v with a restart probability $(1-c)$. This process can be analogized to the spread of an ink drop on paper. The network proximity between u and v can be defined as the steady-state probability that the walk appears at u. If we stack the proximity scores into a vector $\mathbf{p}$, the definition of RWR is given by:

$$\mathbf{p} = cW\mathbf{p} + (1-c)\mathbf{e} \tag{1}$$

where W represents the column normalized adjacent matrix of the viewpoint network G_h^v (details referred to Appendix A), and $\mathbf{e}$ is the starting vector for v. In the following, we use $P(u)$ to denote the proximity score of user u (relative to v).

For clarity of presentation, we temporarily assume the network structure to be static, which may not hold for real social networking sites. Later, we will lift this assumption and take into account the impact of network dynamics in specifying access control policies.

3.2 Historical Social Activity

While the network structures reflect the relatively static social connection between two users, the social activities capture their dynamic social interactions. Right now, most social sites support myriad online activities (e.g., join online communities, comment on others' microblogs, play online games), which makes it feasible to understand such interactions by extracting semantically meaningful patterns from the activity data.

Without more detailed information, we can model an activity using (i) the set of users who have participated in it, and (ii) its activity type[2] (from a finite set $\mathcal{A}$), which indicates its nature (e.g., photo sharing or game playing). Particularly, since we intend to model the social roles of relevant users with respect to target v, we focus on the set of activities participated by v. We can organize the set of users and activities in a bipartite-like graph, as shown in the left plot of Figure 2.

To extract the set of social roles from such activity data, we introduce a probabilistic user-role-activity generative model. We assume that each user u is associated with a conditional multinomial distribution $P(r|u)$ over a set of roles $r \in \mathcal{R}$ ($\mathcal{R}$ is latent), measuring the degree that u carries each role r; further, each role r is associated with a multinomial distribution $P(a|r)$ over the set of activity types $a \in \mathcal{A}$, indicating the likelihood that an individual

[2]Without ambiguity, in what follows, we use a to denote both an activity and its associated activity type.

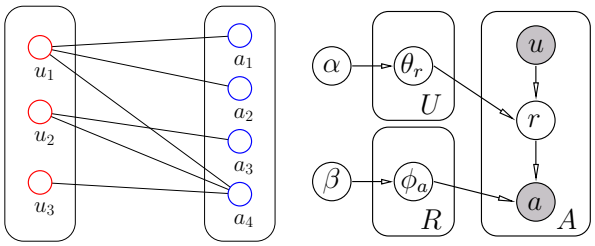

Figure 2: User-activity bipartite graph and user-role-activity generative model.

with role r participates in an activity of type a. Conceptually, the event that user u participates in an activity of type a is generated in two steps: 1) u picks a role r (sample a role r from $P(r|u)$); 2) with role r, u participates in a (sample an activity a from $P(a|r)$). In this model, we have one focused objective, that is identifying the number of latent roles and the association between each user and each role.

The Bayesian network of this general generative model is shown in the right plot of Figure 2: θ_r and ϕ_a denote the Dirichlet priors, parameterized by α and β, while U, R, and A represent the number of users, roles, and activities in the data, respectively. The conditional distribution of parameters for given user and activity is calculated as:

$$P(\theta_r, \phi_a, r|a, u, \alpha, \beta) = \frac{P(\theta_r, \phi_a, r, a, u|\alpha, \beta)}{\sum_a \sum_u P(\theta_r, \phi_a, r, a, u|\alpha, \beta)} \tag{2}$$

where the joint distribution $P(\theta_r, \phi_a, r, a, u|\alpha, \beta)$ can be calculated using

$$P(a|\phi_a, r)P(\phi_a|\beta)P(r|u, \theta_r)P(\theta_r|\alpha)P(u)$$

In our implementation, we apply *Gibbs sampling* [19] to estimate all the parameters, and *perplexity* measure to select the optimal number of roles (details referred to Appendix B). After the parameters are estimated, we can easily identify the association between each user u and each role r, as reflected in the conditional distribution $P(r|u)$ computed as $P(r|u) = P(r|u, \theta_r)P(\theta_r|\alpha)$.

3.3 Computing Role Score

Recall that from the social network structure, we extract a prior distribution $P(u)$ over all the users, while from the dynamic social activities, we obtain a conditional distribution $P(r|u)$. We can therefore use the joint distribution $P(u, r) = P(r|u)P(u)$ (role scores) to measure the probability that user u with role r consumes information from v. In what follows, let $\theta_{u,r}$ denote the score $P(u, r)$. We organize $\{\theta\}$ as a $U \times R$ matrix with the u-th row, r-th column element being $\theta_{u,r}$. For a given role r^*, one can rank all the users according to their role scores θ_{u,r^*}, i.e., the r^*-th column, which serves as the cornerstone for the user-permission assignment, as discussed next.

4. USER-PERMISSION ASSIGNMENT

In this section, we present the step of bridging users and permissions via social roles. One key feature of xAccess lies in its consideration of the structure of permissions. For ease of presentation, we assume that the set of permissions $\{\phi\}$ correspond to a single (unknown) social role r^*, and are pair-wise comparable: $\phi_i \prec \phi_j$, if and only if the grant of ϕ_j implies that of ϕ_i, i.e., the confidentiality levels of two permissions are comparable. We will allow the more general case that permissions correspond to multiple social roles, and are not pair-wise comparable (not monotonically sortable) in

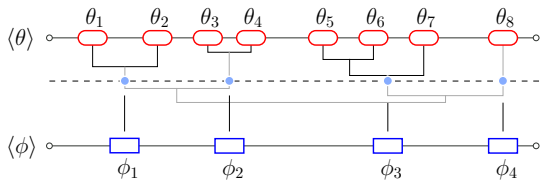

Figure 3: User-permission assignment based on agglomerative clustering.

Section 5. Here, without loss of generality, we assume that $\{\phi\}$ is sorted in a monotonic list $\langle\phi\rangle$.

Intuitively, we are given two measures, role scores $\langle\theta\rangle$ (with respect to a specific role r^*) and permissions $\langle\phi\rangle$. We intend to match each θ_u[3] to certain ϕ. Let $M(\cdot)$ denote the mapping of indices, such that θ_u is mapped to $\phi_{M(u)}$. We dictate that the mapping must be: (1) *Monotonic*. If $\theta_{u_i,r^*} < \theta_{u_j,r^*}$, then $M(u_i) \leq M(u_j)$; that is, for given two users, their assigned permissions and role scores should follow the same order. (2) *Complete*. $\forall u, \exists M(u)$; that is, for every user, there must exist an assigned permission, but may not vice versa. We attempt to match users and permissions while obeying their orders.

We distinguish the case that each permission is associated with a quantitative confidentiality score (e.g., via automated mining [16]) indicating its privacy level, and the more general case that only the ordering information is given. Here, for clarity of presentation, we only consider the latter general case, and the former case is discussed in Appendix C. Next, we start with the case that no predefined user-permission assignments (called "anchors") are given, and later consider the case that the user supplies a set of anchors.

4.1 Matching without Anchors

In the case that no anchors are given, we attempt to provide assistance for the most non-expert users. We therefore rely on the intrinsic structure of the series $\langle\theta\rangle$ to identify the optimal mapping between $\langle\phi\rangle$ and $\langle\theta\rangle$. Without further information, it is impossible to identify the latent role r^* underlying $\langle\phi\rangle$; hence, we evaluate user u based on its marginal role score $\theta_u = \sum_r \theta_{u,r}$.

We assume that both series have been sorted in non-decreasing orders. Intuitively, we consider that two individuals with similar role scores should be assigned similar permissions; the question is thus to find a partition of $\langle\theta\rangle$ into a set of subsets, where the number of subsets is unknown. Clearly, an unsupervised partitioning method is suitable for our purpose. To this end, we apply an agglomerative hierarchical clustering method, which intuitively creates a hierarchy of clusters, called dendrogram, with leaves as the series of role scores, non-leaf nodes as clusters, and root corresponding to the entire collection of users. To construct the dendrogram, one starts at the leaves, and successively merges the closest clusters together, until all the users are included. This process is illustrated in Figure 3. Clearly, one critical measure is the similarity of two consecutive clusters; in out implementation, we adopt the average Manhattan distance as the similarity metric. Cutting the hierarchy at a given height generates a partition at a selected precision. We use a parameter λ to control the precision: the partition continues only if the number of clusters is larger than the number of permissions, or there exist two consecutive clusters with similarity above λ.

After the cluster generation, one may follow a *conservative* (starting from the permission with the lowest confidentiality level, assign a distinct permission to each cluster in an increasing order), *open* (starting from the highest confidentiality level, assign a distinct permission to each cluster in a decreasing order), or *random* (arbitrarily pick the same number of permissions as clusters, and assign them to the clusters following their order) strategy.

[3]For simplicity, we use θ_u as a short version of $\theta_{u,\cdot}$.

4.2 Matching with Anchors

It is possible that the target user may have a set of predefined user-permission assignments, called "anchors", that, in our setting, is equivalent to a set of role score-permission match. More formally,

DEFINITION 3 (ANCHOR). *An anchor is a user predefined role score θ to permission ϕ match $(\theta - \phi)$.*

Anchors provide important implications regarding the target user's expected permission assignment: that is, users with similar role scores to an anchor should be assigned similar permissions. The challenge lies in, however, that the anchors may also introduce inconsistency into the matching process. By inconsistency, intuitively, we mean that for two given anchors, their assigned permissions disobey the order of their associated role scores. Next, we discuss how to incorporate anchors to improve the quality of assignment, and how to detect and resolve potential inconsistency in anchors.

Detecting Inconsistent Anchors

We first introduce the formal definition of *inconsistency*.

DEFINITION 4 (INCONSISTENCY). *An inconsistency is a pair of anchors $(\theta_{u_i} - \phi_{M(u_i)})$ and $(\theta_{u_j} - \phi_{M(u_j)})$, such that $\theta_{u_i,r^*} < \theta_{u_j,r^*}$ and $\phi_{M(u_i)} \succ \phi_{M(u_j)}$.*

To identify the subset of inconsistent anchors, we resort to the principle of *minimal causations* [21]. Intuitively, it states that the best explanation of a given set of data features the minimum set of causes. Based on this principle, we propose the following detection scheme. For each role $r \in \mathcal{R}$ (each dimension of $\langle\theta\rangle$), we check if any inconsistency exists, i.e., if $\exists(\theta_{u_i} - \phi_{M(u_i)})$ and $(\theta_{u_j} - \phi_{M(u_j)})$, $\theta_{u_i,r} < \theta_{u_j,r}$ and $\phi_{M(u_i)} \succ \phi_{M(u_j)}$. If positive, r is added to an initially empty set $\mathcal{R}^I$. If all the dimensions contain inconsistency (i.e., $\mathcal{R} = \mathcal{R}^I$), we regard the dimension (role) r containing the minimum number of inconsistencies as r^*, and proceed to resolving the inconsistency (see below); otherwise, we rank users based on their role scores along the dimensions $\mathcal{R} \setminus \mathcal{R}^I$: $\theta_u = \sum_{r \in \mathcal{R} \setminus \mathcal{R}^I} \theta_{u,r}$. In this case, we can consider the provided anchors as a set of ground-truth role score-permission assignment, $\langle\theta_u, \phi_{M(u)}\rangle$.

This set of anchors slice the two sequences $\langle\theta\rangle$ and $\langle\phi\rangle$ into pieces. Consider two consecutive anchors (in terms of θ values), $\langle\theta_{u_i}, \phi_{M(u_i)}\rangle$ and $\langle\theta_{u_j}, \phi_{M(u_j)}\rangle$. We are left with aligning the two sub-sequences, $\langle\theta_{u_{i+1}}, \ldots, \theta_{u_{j-1}}\rangle$ and $\langle\phi_{M(u_i)}, \ldots, \phi_{M(u_j)}\rangle$, which can be easily solved following the paradigm introduced in Section 4.1.

Resolving Inconsistent Anchors

Next we show how to resolve the inconsistency introduced by the anchors, i.e., the permissions of the anchors disobey the order of the associated social role scores. In general, we solve such inconsistency using two mechanisms, *exceptionalization* and *multi-assignment*. Intuitively, exceptionalization allows the cases that some users have exceptional trust (or distrust) by the target user, not reflected in their social relationships and activities; while multi-assignment accommodates the cases that the anchors only reflect certain aspects of the "true" assignments for some users, who essentially should be given certain other permissions. More formally,

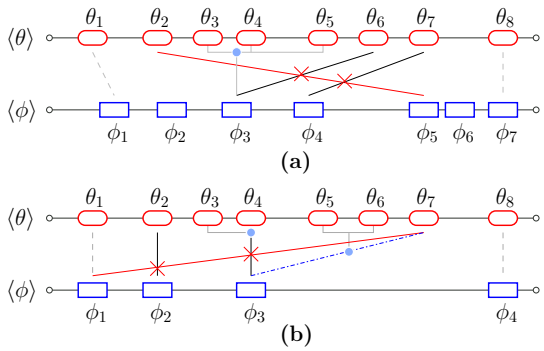

Figure 4: User-permission matching under inconsistent anchors (solid lines): (a) exceptionalization (b) multi-assignment.

DEFINITION 5 (EXCEPTIONALIZATION). *An exception is an anchor* $(\theta_u - \phi_{M(u)})$ *that encodes special trust (or distrust) to user* u *by the target, which should not be used in the assignment process to provide general guidance.*

We apply exceptionalization to identify the set of anchors that would result in inconsistency and should be considered as exceptions, and suspend them from providing guidance for permission assignment for the rest users.

DEFINITION 6 (MULTI-ASSIGNMENT). *For a given anchor* $(\theta_u - \phi_{M(u)})$ *, multi-assignment identifies potentially missing permission assignment for* u *other than* $\phi_{M(u)}$.

We apply multi-assignment to identify the set of anchors that should be accompanied by certain other permissions, and remove the original permission assignments from the matching process. Multi-assignment is especially meaningful when incomparable permissions are taken into consideration, and users tend to feature multiple social roles, e.g., both friend and co-worker, as viewed from different perspectives; in such cases, a single user may be associated with multiple permissions along different dimensions (i.e., multiple roles).

Our approach of matching role score and permission under anchors with possible inconsistency is sketched in Algorithm 1: 1) check if inconsistency exists in the set of anchors; 2) identify the minimum subset of anchors (exceptions), whose absence removes inconsistency; 3) depending on its intersection with other anchors, apply either multi-assignment or exceptionalization to each exception; 4) apply time-warping-based (for permissions associated with confidentiality scores) or clustering-based matching to sub-series of permissions and role scores, as partitioned by the anchors.

While the overall framework is clear, we still need to answer several challenging questions: first, how to identify the minimum set of exceptions? second, whether to apply multi-assignment or exceptionalization, when both are possible? Following, we answer these questions in the case of pair-wise comparable permissions, and the more general permission structures will be discussed in Section 5. For simplicity, we use ψ to denote an anchor, and $\theta(\psi)$ and $\phi(\psi)$ as its associated role score and permission.

Q1: how to find the minimum set of exceptions? We have the following theorem regarding the complexity of finding the minimum set of anchors that result in inconsistency (exceptions).

THEOREM 1. *Identifying the minimum set of anchors responsible for inconsistency is NP-Hard.*

Input: proximity set Θ, permission set Φ, anchor set Ψ
Output: permission assignments for all $\theta \in \Theta$

```
// consistency check
```
$\Psi^e, \Psi^r \leftarrow \varnothing$;
for *each anchor* $\psi \in \Psi$ **do**
  $n_\psi \leftarrow$ number of intersected anchors;
  if $n_\psi > 0$ **then** add ψ to Ψ^e;

```
// inconsistency removal
```
if $\Psi^e \neq \varnothing$ **then**
  sort $\psi \in \Psi^e$ according to n_ψ;
  while *inconsistency exists* **do**
    pop up ψ from Ψ^e to Ψ^r;
    adjust the order of Ψ^e;
  $\Psi \leftarrow \Psi \setminus \Psi^r \cup \Psi^e$;
  for *each* $\psi \in \Psi^r$ **do**
    if $\exists \psi' \in \Psi, \theta(\psi) > \theta(\psi')$ *and* $\phi(\psi) \prec \phi(\psi')$ **then**
      multi-assignment;
    else
      exceptionalization;

```
// proximity-permission matching
```
for *two consecutive anchors* $(\theta_i, \phi_{m_i}), (\theta_j, \phi_{m_j}) \in \Psi$ **do**
  if *permissions associated with confidentiality* **then**
    match $\theta_{i+1} : \theta_{j-1}$ and $\phi_{m_i} : \phi_{m_j}$ using time warping;
  else
    if $j - i > m_j - m_i$ **then**
      match $\theta_{i+1} : \theta_{j-1}$ and $\phi_{m_i} : \phi_{m_j}$ using clustering;
    else
      conservative, aggressive, or arbitrary matching;

Algorithm 1: Role-permission matching under (possibly inconsistent) anchors.

PROOF. The problem can be re-formulated as the following *Set Cover* problem. Let $\mathcal{S}$ be the set of intersection points of anchors, and $\mathcal{A}$ be the set of anchors involved in the intersections. We intend to find the minimum subset of $\mathcal{A}$ that "covers" all the intersection points in $\mathcal{S}$, which is an instantiation of the classical set cover problem, known to be NP-Hard. □

Hence, instead of attempting to find the minimum set, we apply a greedy approach: at each step, we identify the anchor that causes the largest number of inconsistencies in the current anchor set, and remove it as an exception. The intuition behind this scheme is that an anchor in conflict with a larger number of others tends to be an exception with higher possibility. It can be derived that this greedy approach achieves an approximation ratio of $H(s)$, where s is the largest number of intersections on a single anchor, and $H(n)$ the n-th harmonic number.

Q2: exceptionalization or multi-assignment? After identifying the set of exceptions, the anchors are divided into two sets: the exceptions Ψ^e and the rest Ψ^r, the next step is to apply exceptionalization or multi-assignment to handle each $\psi \in \Psi^e$. Conceptually, both mechanisms remove ψ from the matching process; while, in addition, multi-assignment also attempts to find any missing permission assignments potentially neglected by the user. We distinguish the cases that ψ intersects with "up-stream" anchors, i.e., $\exists \psi' \in \Psi^r, \theta(\psi) < \theta(\psi')$ and $\phi(\psi) \succ \phi(\psi')$, as shown in Figure 4$(a)$, or with "down-stream" ones, i.e., $\exists \psi' \in \Psi^r, \theta(\psi) > \theta(\psi')$ and $\phi(\psi) \prec \phi(\psi')$, as shown in Figure 4$(b)$. We claim that in the case of pair-wise comparable permissions, only one case is possible, with the following theorem.

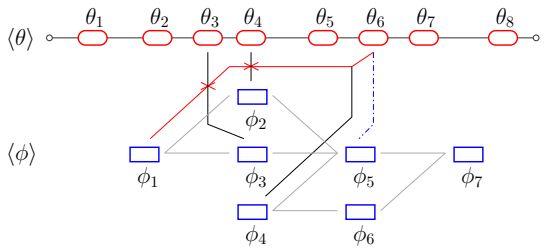

Figure 5: Role-score-permission matching under partially ordered permissions.

THEOREM 2. *For pair-wise comparable permissions, each exception can only exclusively intersect with either up-stream or down-stream anchors.*

PROOF. Without loss of generality, consider an exception ψ that intersects with an "up-stream" anchor, i.e., $\exists\psi' \in \Psi^r$, $\theta(\psi) < \theta(\psi')$ and $\phi(\psi) \succ \phi(\psi')$. Assume that it also intersects with certain "down-stream" one, i.e., $\exists\psi'' \in \Psi^r$, $\theta(\psi) > \theta(\psi'')$ and $\phi(\psi) \prec \phi(\psi'')$. We have $\theta(\psi'') < \theta(\psi')$ and $\phi(\psi'') \succ \phi(\psi')$, i.e., an exception, which is a contradiction to that Ψ^r is exception-free. □

Hence, multi-assignment makes sense only when ψ intersects with "down-stream" anchors; that is, an additional permission ψ^* with higher confidentiality level ($\phi(\psi^*) \succ \phi(\psi)$) is assigned. Without further information, we set the additional permission ϕ^* as the *minimum* one that does not cause any inconsistency in Ψ^r: $\psi^* = \max\{\phi(\psi') : \psi' \in \Psi^r \text{ and } \theta(\psi') < \theta(\psi)\}$. An example is shown in Figure 4(b) where the exception $(\theta_7 - \phi_1)$ intersects with two down-stream anchors $(\theta_2 - \phi_2)$ and $(\theta_4 - \phi_3)$, and an additional permission ϕ_3 is assigned to θ_7.

Also note that the multi-assignement policy bears the nature of "suggestion"; it is possible that an exception encodes certain special "distrust" preference by the target user. In such cases, only exceptionalization will be applied.

5. EXTENSION

In the discussion so far, we have made two simplification assumptions: 1) the set of permissions are pair-wise comparable, i.e., a totally ordered set; 2) the target user provides sufficient information (e.g., social network structures, historical social activities, anchors), for xACCESS to perform privacy setting recommendation. Now we lift these simplifications and consider the cases that 1) the set of permissions form a partially ordered set, i.e., not every pair of permissions are comparable, which is fairly common in access control literatures and practice [9], and 2) the information supplied by the target user is insufficient to make informative recommendation; we can, however, gain valuable insights into the reasonable privacy setting by examining the settings of his/her peers. Further, we discuss the problem of semantically labeling a social role, thereby making it interpretable by users.

5.1 Partially Ordered Permissions

Assume that the set of permissions form a partially ordered set (i.e., lattice) according to their confidentiality levels. Now, a pair of permissions can be one of the following relationships, $\succ$, $\prec$, $=$, and *incomparable*; therefore, the techniques in Section 4 is not directly applicable. One can, however, apply *linear extension* [8] over the set of permissions, which generates a *topological ordering* of the permissions, compatible with the original partial ordering. In particular, we are interested in a classed representation of the ordering information, where the permissions in each class are equal or incomparable to each other, and can be considered as having equivalent confidentiality level. For example, in Figure 5, the set of permissions can be grouped into four classes $\{\phi_1\}$, $\{\phi_2, \phi_3, \phi_4\}$, $\{\phi_5, \phi_6\}$, $\{\phi_7\}$. Note that these classes are totally ordered, and it holds that for every two consecutive classes Φ_i and Φ_j, $\forall\phi \in \Phi_i$, $\exists\phi' \in \Phi_j$, $\phi \prec \phi'$. Such classification can be obtained following a breadth-first search paradigm, as sketched in Algorithm 2 (we define two permissions ϕ and ϕ' as predecessor and successor, respectively, if and only if $\phi \prec \phi'$).

Input: set of permissions Φ
Output: classification of Φ
$\mathcal{C} \leftarrow \varnothing$;
$C \leftarrow$ set of permissions with no successors;
$C' \leftarrow \varnothing$;
while *C is non-empty* **do**
  add C to $\mathcal{C}$;
  for *each permission* $\phi \in C$ **do**
    for *each direct predecessor* ϕ' *of* ϕ **do**
      delete relationship $\phi' \prec \phi$;
      if *ϕ' has no successor* **then**
        add ϕ' to C';
  $C \leftarrow C'$;

Algorithm 2: Linear extension of partially ordered set.

Now we can perform role-score permission assignment on the level of permission classes. In the case that no anchors are provided, we apply agglomerative clustering to the set of social roles scores, generate a partition, and match each class of role scores (i.e., users) with a distinct permission class, following their orders. Such class-to-class mapping is then presented to end users for further refinement.

Given the non-comparability of permissions, it is likely that for one social score θ_u, the target user may provide multiple anchors, $\{\psi^u\}$. As an example, in Figure 5, θ_6 is assigned two permissions ϕ_1 and ϕ_4. We assume that each pair of $\phi(\psi_i^u)$ and $\phi(\psi_j^u)$ are incomparable; otherwise, one can remove the one with lower confidentiality level, without affecting the overall capacity.

We consider that such multiple anchors correspond to multiple social roles; that is, each $\psi_i^u \in \{\psi^u\}$ is associated with a role. Conceptually, two anchors ψ_i^u and $\psi_j^{u'}$ associated with the same role should be consistent; hence, we intend to find a permission-role mapping such that the number of inconsistencies could be minimized. After that, one can perform user-permission assignment along each dimension (role) independently, solve the possible inconsistency, and finally collect and merge the exception-free anchors. Exception handling is similar to that in Section 4.2, except that the selected additional permission ϕ^* should be a successor of the permissions associated with conflicting anchors, i.e., their least common ancestor (LCA). One then merge the set of exception-free anchors and the identified missing permission assignment to form the anchor set for proximity-permission matching. For example, in Figure 5, $(\theta_6 - \phi_1)$ conflicts with anchors $(\theta_3 - \phi_3)$ and $(\theta_4 - \phi_2)$; the LCA of θ_2 and θ_3, θ_5, is identified as the missing permission, which is then merged with another assignment ϕ_4, with ϕ_5 as the final assignment for ϕ_6.

5.2 Collaborative Privacy Setting

The privacy settings by his/her peers provide valuable information for determining the best access control policy for a specific

user, especially when the information associated with the user (e.g., social activities, anchors, etc.), is insufficient for xAccess to perform informative recommendation. Here, we discuss how to leverage such information in suggesting reasonable privacy setting.

The most straightforward solution is based on the principle of *mutual equivalence*: a pair of individuals tend to demonstrate similar trust/distrust inclination in information sharing with each other; hence, one can "mirror" the setting of a peer: given two individuals u and v, let $\phi(u \rightsquigarrow v)$ denote the permission assigned by u to v; v can simply copy this setting as $\phi(v \rightsquigarrow u) = \phi(u \rightsquigarrow v)$. This solution, though simple, considers only the information of the specific peer when determining his/her access level. A more comprehensive solution is based on the paradigm of *collaborative filtering*. Given two individuals u and v, for the sets of users relevant to u and v, $\mathcal{V}_u$ and $\mathcal{V}_v$, one creates a mapping (let $M_{uv}(w)$ be the counterpart of w of $\mathcal{V}_u$ in $\mathcal{V}_v$), based on the social role scores of $\mathcal{V}_u$ and $\mathcal{V}_v$ with respect to u and v, respectively. The setting of $w \in \mathcal{V}_u$ can be calculated as: $\phi(u \rightsquigarrow w) = \arg\min_\phi \prod_v f(\phi(v \rightsquigarrow M_{uv}(w)), \phi)$, where $\phi(v \rightsquigarrow M_{uv}(w))$ is the actual assignment to $M_{uv}(w)$ by v, and $f(\phi(v \rightsquigarrow M_{uv}(w)), \phi)$ is the *cost* function of assigning ϕ to $M_{uv}(w)$ by v. Various instantiations are possible, L^1 norm for example, $f(\phi(v \rightsquigarrow M_{uv}(w)), \phi) = |\phi(v \rightsquigarrow M_{uv}(w)) - \phi|$.

5.3 Labeling Social Roles

To make the extract social roles interpretable, it is imperative to attach "semantical tags" to them. Back to our discussion in Section 3.2, we assume that each activity type a is associated with a set of descriptive terms, from a finite set $\mathcal{W}$. We can extend the generative model in Section 3.2 by including another observable w, i.e., the terms of an activity type a, associated with multinomial distribution ψ_w parameterized by γ. Now, the joint distribution is given by $P(w, a, r, u|\alpha, \beta, \gamma)$, which can be estimated following that sketched in Section B. From the joint distribution, one can derive the conditional distribution $P(w|r)$.

We can extract a set of candidate labels using frequent pattern mining. For each candidate label l, we evaluate its semantic relevance to a role r, $S(l, r)$. More formally, let $l = w_0^l w_1^l \dots w_m^l$, we can estimate its semantical relevance to a role r using multiple metrics, the simplest case, for example:

$$S(l, r) = \log \frac{P(l|r)}{P(l)} = \sum_{i=0}^{m} \log \frac{P(w_i^l|r)}{P(w_i^l)} \quad (3)$$

alternatively, the negative KL divergence of $\{P(w|r)\}$ and $\{P(w|l)\}$ over $w \in \mathcal{W}$ could also be used.

6. EMPIRICAL EVALUATION

In this section, we present an empirical evaluation of the efficacy of xAccess over two real-life social network and user study datasets. The experiments are specifically designed centering around the following metrics: 1) the efficacy in capturing individuals' implicit privacy preference for relevant users, 2) the effectiveness in incorporating users' predefined preference to improve the quality of privacy setting, 3) the scalability with respect to the scale of underlying social network and the volume of historical activity data.

6.1 Datasets and Experimental Design

In the first set of experiments, we apply xAccess to analyzing a publicly available speed dating dataset [12] from a study conducted by Fishman et al. [12]. It involves 530 participants and consists of data regarding 4,150 dynamic "dates" arranged between pairs of participants. For each participant, the demographic information (e.g., age, race, zipcode, etc.) and the information of hobby activities (e.g., entertainment, museum, hiking, etc.) the participants usually take part in is also collected. After the date, the satisfaction of each participant regarding his/her partner is recorded with a score on a scale from 1 to 10, which we regard as the implicit privacy preference indicated by the participant. For each individual, we apply xAccess to extracting the roles of his/her partners, based on the structure of dating arrangement (as the static network structures) and the description of their hobbies (as the dynamic social activities), and match it against the set of permissions (the set of integers over $[1, 10]$). We compare the predicated results with that given by the participants in the dataset.

In the second set of experiments, we analyze the social network of a subset of IBM employees who participated in the Small Blue project [14] and the archive of bookmarks tagged by these social users (as the dynamic activity data), to predict individuals' information sharing behavior. The dataset corresponds to the social network as of January 2009, which involved 41,702 IBM employees. The personal information regarding each individual includes his/her (i) work location, (ii) managerial position, and (iii) social connections with other employees. The associated bookmark archive consists of the webpages tagged by the individuals appearing in the first dataset, collected by Dogear [3], a personal bookmark management application that as well supports sharing the community's bookmarks. The archive contains 20,870 bookmark records, relevant to 7,819 urls. Attributes of interest to us are listed in Table 1; in particular, *email* and url uniquely identifies a user and a webpage, respectively, and *tags* encode the semantics of the object. We regard the volume of email sent from an individual to his/her relevant user (note this communication is directional) as a quantitative indication of his/her intension of information sharing, and evaluate the result predicted by xAccess against it.

Attribute	Description
email	email address of user s (identifier of subject)
url	url o bookmarked by s (identifier of object)
tags	bookmark tags made by s regarding o
time	time-stamp that s accesses o

Table 1: Attributes and descriptions of Dogear dataset.

In the last set of experiments, we implement and deploy xAccess on the platform of Facebook and conduct a concrete user study on helping everyday users specify their privacy policies. For a given Facebook user u, we consider the following types of social activities of u's friends with respect to u's Facebook page: *comment* (post), *like* (page, post, status), and *tag* (photo).

Structure	Permission order
total order	$(4) > (5) > (7) > (8) > (2) > (6) > (3) > (9) > (10) > (11) > (12) > (1)$
partial order	$\{(4)\} > \{(5), (7), (8)\} > \{(2), (6), (3), (9)\} > \{(10), (11), (12)\} > \{(1)\}$

Table 2: Alternative structures of permission order.

Additionally we consider the following set of private data items (permissions): *(1) About me, (2) Personal Info, (3) Birthday, (4) Religious and Political Views, (5) Family and Relationship, (6) Education and Work, (7) Photos and Videos of Me, (8) Photo Albums, (9) Posts by Me, (10) Allow to post on my Wall, (11) Posts by Friends, (12) Comments on Posts*. We consider alternative access structures of these permissions (e.g., totally ordered set, partially ordered set) as listed in Table 2. We then collect the privacy settings regarding these permissions by 23 volunteers, and compare the privacy settings suggested by xAccess with that manually labeled by the participating users.

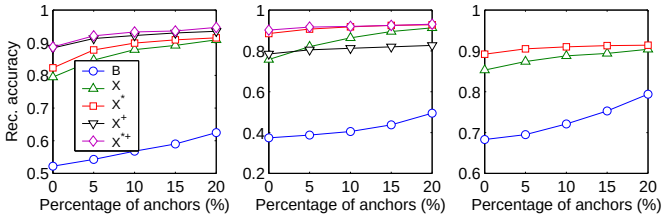

Figure 6: Recommendation accuracy of xAccess and baseline approach with respect to the number of correct and false anchors. From left to right, the three columns correspond to the speed dating, small blue, and facebook datasets, respectively.

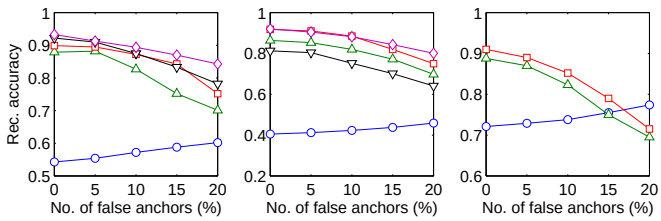

Figure 7: Robustness of xAccess and baseline approach against false anchors.

We use all three datasets in the experiments, aiming at capturing the influence of factors such as activity types and user characteristics. The algorithms are implemented using Python, and all the experiments are conducted on a workstation with 1.6GHz Pentium IV and 2GB memory, running Windows XP.

6.2 Experimental Results

Capture of Privacy Preference

This set of experiments are designed to evaluate the efficacy of xAccess (denoted by X) in capturing social users' implicit preference of information sharing with relevant users. In particular, we intend to examine the contributions by different features (i.e., static social network structure, dynamic historical activities) in capturing such implicit reference. Let $\phi^*(\cdot)$ and $\phi^r(\cdot)$ be the access control level set manually by the user, and suggested by a recommendation method, respectively. We measure the quality of recommendation using the metric of *recommendation accuracy*,

$$1 - \frac{\sum_{i \in \mathcal{I}} |\phi^*(i) - \phi^r(i)|}{|\mathcal{I}| \cdot |\Phi|}$$

where $\mathcal{I}$ and Φ are the set of individuals, distinct access levels, respectively. Further, we construct a baseline bayesian approach (denoted by B) that makes recommendation solely based on hop distance, i.e., *friend*, *friend-of-friend*, and minimizes the recommendation error, i.e., a unique setting ϕ^r_h for all users with hop distance h to the target individual that satisfies

$$\phi^r_h = \arg\min_{\phi_h} \sum_{i \in \mathcal{I}_h} |\phi^*(i) - \phi^r(i)|$$

where $\mathcal{I}_h$ is the set of users with hop distance h. We also consider the possibility of leveraging the possible quantitative confidentiality levels associated with permissions (detailed discussion in Section C). Overall, we implemented four versions of xAccess, X, X*, X+, and X*+, where the symbols $*$ and $+$ indicates that the version considers dynamic social activities and confidentiality scores, respectively.

	B	X	X*	X+	X*+
speed dating	0.522	0.795	0.823	0.884	**0.887**
small blue	0.374	0.758	0.885	0.785	**0.902**
facebook	0.683	0.853	**0.892**	N/A	N/A

Table 3: Accuracy of privacy settings suggested by xAccess and baseline approach.

Table 3 shows the accuracy of the four versions of xAccess and the baseline approach with respect to the three datasets. It is observed that, for all three datasets, over the baseline approach, xAccess achieves approximately 1.3 ~ 2.4 times higher recommendation accuracy. It is noted that the incorporation of dynamic social behavior significantly boots the accuracy, especially for the small blue dataset (17.4% increase). This can be explained by that bookmarks well capture users' interests and preferences, and the behavior of recommending bookmarks is a good indicator of user's will of information sharing. Also, the incorporation of quantitative confidentiality information of permissions further improves the quality of recommendation, which is especially evident for the speed dating dataset. This is explained by that the social network structure in this dataset is much simpler (mainly composed by 1-hop neighbors), while the social activity data includes 11 attributes, and contains much semantically richer information.

Incorporation of User Input

In this set of experiments, we take into account user predefined permission assignment (anchor). Specifically, we measure the recommendation accuracy of xAccess and the baseline approach with respect to varying percentage of anchors (over the total number of assignments), where the anchors are randomly selected.

The result is shown in Figure 6. First notice that, as the number of provided anchors grows, the accuracy of all the models increase; intuitively, the anchors provide valuable clues regarding users' implicit preference. Also notice that xAccess (all four versions) demonstrates higher effectiveness in leveraging such hints to improve the quality of assignment; for example, for the small blue dataset, even the basic version X achieves accuracy approximately 0.87 when 10% of the assignments are provided as anchors, compared with approximately 0.42 of B. This is explained by the fact that xAccess leverages the anchors as "structural clues" for aligning social role measures and permissions, which improves the overall quality of the alignment, in contrast to the point-wise improvement by the baseline approach.

To evaluate the impact of the inconsistency possibly existing in the anchors, we randomly generate a set of "false" anchors, in addition to the anchors provided by users. With the percentage of "correct" anchors fixed as 5%, we measure the accuracy of xAccess with respect to the varying percentage of false anchors (note that the baseline approach treats anchors as point-wise information, therefore is not affected by the false anchor). The result is shown in Figure 7: on all three datasets, the accuracy of xAccess is fairly stable under the influence of false anchors, mainly attributed to the exceptionalization mechanism.

Efficiency and Scalability

Now, we proceed to evaluating the operation efficiency of xAccess. In particular, we intend to capture the influence of two factors: the scale of the underlying social network, and the volume of historical activity data. We use the small-blue dataset in this set of experiments, given its large scale.

First, we measure the wall execution time of xAccess as a function of the maximum hop h of the viewpoint networks. The result

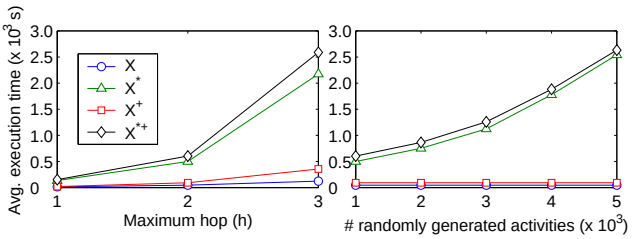

Figure 8: Average execution time (per user) of xAccess versus the maximum hop and the volume of social activity data.

is illustrated in the left plot of Figure 8. Overall, it is noticed that X and X^{+} are fairly efficient, even though the number of relevant users grows approximately quadratically. This is attributed to the fact that extracting the social proximity measure from the social network only involves solving a linear equation system, typically featuring polynomial complexity for sparse matrices. While the extraction of social roles in X^{*} and X^{*+} is costly; their overall execution efficiency, however, is fairly reasonable, considering the scale of the small blue social network (over 40K individuals).

Further, in addition to the activities (bookmarks) in the dataset, we randomly injected in a set of user-activity pairs to evaluate the scalability of xAccess against the size of activity data. The right plot of Figure 8 demonstrates how the volume of activity data affects the efficiency of xAccess (with h fixed as 2), which exhibits even less significant impact over the performance of xAccess, compared with the scale of social network (note that X and X^{+} are not affected). This can be attributed to that 1) Gibbs sampling and the optimization of entropy filtering significantly reduces the overall complexity of social role mining; and 2) the number of social activities usually grows quadratically with the scale of the underlying social network.

7. CONCLUSION

This work presents a systematic study on the problem of specifying access control policies over personal data on social sites. We proposed xAccess, a novel automated policy specification tool that can help ordinary social site users understand, specify, and diagnose their privacy settings. Compared with prior work, xAccess highlights itself with three distinct features: 1) it adopts a role-based access control model, instead of the conventional rule-based one, which leads to privacy policies semantically interpretable by users; 2) it exploits both static social network structures and dynamic social activities in extracting the underlying social roles; 3) it considers potential inconsistency in user input permission assignments, and proposes effective countermeasure against such inconsistency. Extensive experiments over real social network data have been conducted to validate the efficacy of xAccess.

8. REFERENCES

[1] British spy chief's cover blown on Facebook: http://www.reuters.com/article/idustre56403820090705.

[2] facebook - Press Room: http://www.facebook.com/press.

[3] Lotus Connections - Dogear: http://www.ibm.com/dogear.

[4] Teacher fired over Facebook sues district: http://www.cbsatlanta.com/news/21573759/detail.html.

[5] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.

[6] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. In *SIGCOMM*, 2009.

[7] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anonymization for social network data. *Proc. VLDB Endow.*, 2(1):766–777, 2009.

[8] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[9] D. E. Denning. A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243, 1976.

[10] L. Fang and K. Lefevre. Privacy wizards for social networking sites. In *WWW*, 2010.

[11] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

[12] R. Fisman, S. S. Iyengar, E. Kamenica, and I. Simonson. Gender differences in mate selection: Evidence from a speed dating experiment. *The Quarterly Journal of Economics*, 121(2):673–697, 2006.

[13] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, 2008.

[14] C.-Y. Lin, N. Cao, S. X. Liu, S. Papadimitriou, J. Sun, and X. Yan. Smallblue: Social network analysis for expertise search and collective intelligence. In *ICDE*, 2009.

[15] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD*, 2008.

[16] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *ICDM*, 2009.

[17] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *SP*, 2009.

[18] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD*, 2004.

[19] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., 2005.

[20] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[21] H. Shaklee and B. Fischhoff. Discounting in Multicausal Attribution: The Principle of Minimal Causation. *SSRN eLibrary*, 1905.

[22] K. Singh, S. Bhola, and W. Lee. xbook: Redesigning privacy control in social networking platforms. In *SECURITY*, 2009.

[23] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu. Scalable proximity estimation and link prediction in online social networks. In *IMC*, 2009.

[24] R. Stark. *Sociology*. Cengage Learning, 2006.

[25] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *SP*, 2010.

[26] L. Zou, L. Chen, and M. T. Özsu. k-automorphism: a general framework for privacy preserving network publication. *Proc. VLDB Endow.*, 2(1):946–957, 2009.

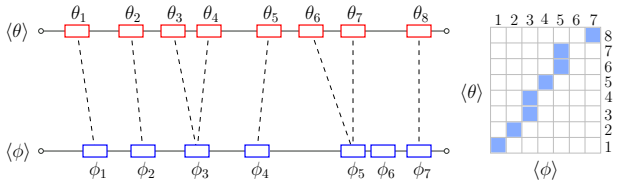

Figure 9: User-permission matching (with quantitative confidentiality scores) using dynamic time warping.

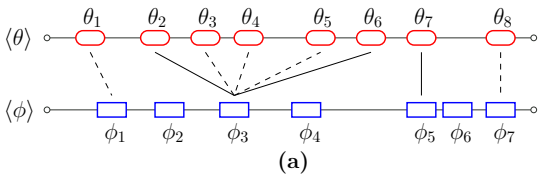

Figure 10: User-permission matching under consistent anchors (solid lines): (a) permissions with confidentiality levels.

APPENDIX

A. RANDOM WALK WITH RESTART

We assume that each relationship type is associated with a weight, indicating its strength. We use w_{ij} to denote the weight of the relationship $\overline{ij}$ (between two direct friends i and j). Specifically, in RWR, at each step, the walk moves from a user j to one of its friends k with probability proportional to the weight w_{jk}, and returns to j (restart) with probability $(1-c)$ (c is a parameter). More concretely, let $\mathcal{N}_j$ be the set of friends of j. The transition probability from j to $k \in \mathcal{N}_j$, p_{jk}, is given as:

$$p_{jk} = \frac{w_{jk}}{\sum_{k' \in \mathcal{N}_j} w_{jk'}} \tag{4}$$

where the parameter c controls the probability of returning to the original node. Stacking p_{ij} into a matrix, column-wise, which produces the column, normalized adjacent matrix W.

B. PARAMETER ESTIMATION

To obtain parameter estimates for the generative model, we employ Gibbs Sampling, a Markov chain Monte Carlo (MCMC) algorithm, as it provides a simple method of performing parameter estimation for Dirichlet priors and allows combinations of estimates from several local maxima of the posterior distribution.

Instead of estimating the model parameters directly, we first evaluate the posterior distribution on role r, then use the results to infer θ_r and ϕ_a. For each activity, the role of users who participate in it (role assignment) is sampled from the following term:

$$P(r_i = j | a_i = m, u_i = k, \mathbf{r}_{-i}) \propto \frac{C^{AR}_{mj} + \beta}{\sum_{m'} C^{AR}_{m'j} + B\beta} \frac{C^{UR}_{kj} + \alpha}{\sum_{j'} C^{UR}_{kj'} + R\alpha}$$

where $r_i = j$ represents the assignment of the i-th activity, and $a_i = m$ and $u_i = k$ represent that the observation that the user k participates in the i-th event of activity type m; A, B, R, U are the number of activity types, activities, roles, and users, respectively; C^{AR}_{mj} is the number of times that an activity of type m is associated with a social role j, similar for C^{UR}_{kj}; $\mathbf{r}_{-i}$ represents the all the role assignment except the i-th activity. From these count matrices, one can easily estimate the parameters θ_r and ϕ_a as:

$$\phi_{mj} = \frac{C^{AR}_{mj} + \beta}{\sum_{m'} C^{AR}_{m'j}} \qquad \theta_{kj} = \frac{C^{UR}_{kj} + \alpha}{\sum_{j'} C^{UR}_{kj'} + R\alpha}$$

Further, in this process, we use *entropy filtering* to filter non-informative trash activities to improve efficiency. Specifically, after N (a user-specified parameter) iterations of sampling, we start to ignore the set of non-informative activities (trash activities). In our implementation, we measure the informativeness of activities using the entropy of the variable C^{AR}. Particularly, we ignore the i-th activity a_i if the i-th row of C^{AR} has entropy above a threshold ω.

The remaining question is how to select the optimal number of latent roles. We employ the perplexity measure, a standard measure of estimating the performance of a probabilistic model. We run the Gibbs sampling using perplexity score as the termination condition; the number of roles is determined by using the minimum number of roles that leads to the near maximum perplexity. More details are referred to [20].

C. PERMISSIONS WITH CONFIDENTIALITY SCORES

Here we consider the case that each permission is associated with a quantitative confidentiality level[4].

Intuitively, we intend to match the shapes of the entire series $\langle\theta\rangle$ and $\langle\phi\rangle$ to the maximum extent; that is, if the difference between θ_i and $\theta_{i'}$ is (non)significant, so should be the case for ϕ_{m_i} and $\phi_{m_{i'}}$. We can formalize this notion as follows:

$$\min_{m_{(\cdot)}} \sum_i \Delta(\theta_i, \phi_{m_i}) \tag{5}$$

where $\Delta(\theta_i, \phi_{m_i})$ is the distance between θ_i and ϕ_{m_i}; its concrete definition depending on the definitions of θ and ϕ.

We assume that both series $\langle\theta\rangle$ and $\langle\phi\rangle$ have been properly normalized to the interval of $[0, 1]$ (e.g., via linear interpolation), and $\Delta(\theta, \phi)$ may simply be the absolute value of their difference. Essentially, the optimization problem of Eq. 5 can be re-formulated as computing the minimum *time warping distance* between $\langle\theta\rangle$ and $\langle\phi\rangle$, $\Delta(\langle\theta\rangle, \langle\phi\rangle)$, with definition given as:

$$\min \begin{cases} \Delta(head(\langle\theta\rangle), head(\langle\phi\rangle)) + \Delta(rest(\langle\theta\rangle), rest(\langle\phi\rangle)) \\ \Delta(\langle\theta\rangle, rest(\langle\phi\rangle)) \end{cases}$$

where $head(\cdot)$ is the first element of a series, and $rest(\cdot)$ is the sub-series without the first element. Specifically, we have

$$\Delta(\langle\rangle, \langle\rangle) = 0 \qquad \Delta(\langle\theta\rangle, \langle\rangle) = \infty \qquad \Delta(\langle\rangle, \langle\phi\rangle) = 0$$

This time warping distance defines a path in the matrix composed of the elements of (θ_i, ϕ_j), corresponding to the alignment of θ_i and ϕ_j, i.e., $m_i = j$, as shown in the right plot of Figure 9. This path represents an optimal mapping between $\langle\theta\rangle$ and $\langle\phi\rangle$. Given the mapping $m_{(\cdot)}$, users with social proximity score θ_i are assigned permission ϕ_{m_i}. The computation of minimum time warping distance can be approached using dynamic programming.

In the case of consistent anchors, we perform piece-wise time-warping distance matching for each piece-pair $\{\theta_{i^*+1}, \ldots, \theta_{j^*-1}\}$ and $\{\phi_{m_{i^*}}, \ldots, \phi_{m_{j^*}}\}$. An example is shown in Figure 10, where the solid lines represent anchors, and the dashed ones derived matches. Note the difference of the match for θ_5 from that in Figure 9. In the case of permission without confidentiality levels,

[4]Here we abuse the notation a little bit, and use ϕ to denote both the permission and its associated confidentiality level (if available).