Setting Access Permission through Transitive Relationship in Web-based Social Networks

Dan Hong Vincent Y. Shen
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong
{csdhong,shen}@cse.ust.hk

ABSTRACT

The rising popularity of Web 2.0, such as blogs, forums, online calendars/diaries, etc., makes users more interested in keeping their data on the Web. Sharing of such data could make life more enjoyable and convenient. For example, posting new photos about activities or sharing views about an event can let friends know what a user cares about. However, some of these data (such as a person's location during a particular time, opinion about a political event, etc.) are private and should not be accessed by unauthorized users. Although Web 2.0 facilitates sharing, the fear of forwarding sensitive data to a third party without knowledge of the data owners discourages people from using certain applications due to privacy concerns. We take advantage of the existing relationships on social networks and build a "trust network" with transitive relationship to allow data sharing while respecting the privacy of data owners. The trust network linking private data owners, private data requesters, and intermediary users is a directed weighted graph. The permission value for each private data requester is automatically assigned in this network based on the transitive relationship. Experiments were conducted to confirm the feasibility of constructing the trust network from existing social networks, and to assess the appropriateness of permission value assignments in the query process. This privacy scheme can make private data sharing manageable by data owners, who only need to define the access rights of their closest contacts once.

Categories and Subject Descriptors

K.4.1 [**Computers and society**]: Public Policy Issues—*Privacy*; H.3.5 [**Information storage and retrieval**]: Online Information Services—*Data sharing*; H.1.2 [**Models and principles**]: User/Machine Systems—*Human factors*

General Terms

Design, Human Factors

Keywords

Data Privacy, Trust Network, Transitive Relationship, P3P, Social Network

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.
.

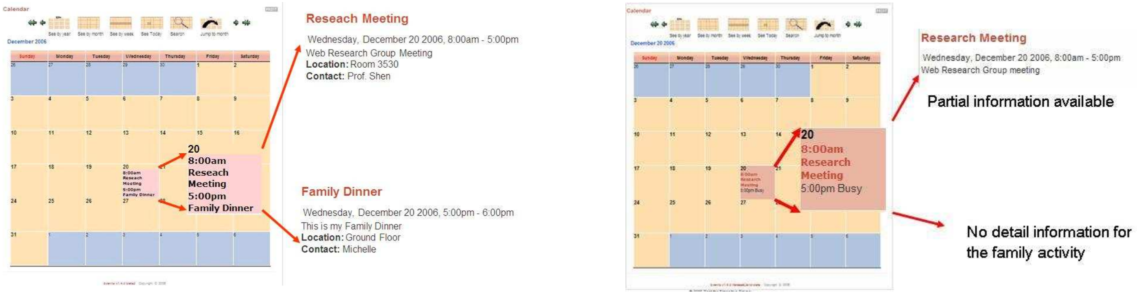
1. INTRODUCTION

With the increasing popularity of Web 2.0 services, more and more Web users post their articles, pictures, comments on the Web through blogs, forums or other Web applications. Based on the "State of the Blogosphere" report [19], 120,000 new weblogs are being created worldwide each day. Many online communities have been established when users create accounts at those website hosts. In these communities users share their beliefs, opinions and interests. Communities on these websites are set up based on the "common interest" page their members marked. Each person can be members of several different communities and private data (e.g. identification, financial record, location, calendar, Web content) are commonly shared along community connections. However, these data sharing activities through Web-based social networking bring serious privacy concerns since users do not have control over who can access their personal data.

Nowadays, many users use a Web-based calendar, such as Google Calendar [9], to arrange their appointment schedules. It is possible to provide a feature to let users define activity categories, such as "family activity", "work activity", "church activity", etc. Figure 1(a) shows such a calendar which has several different categories. When a visitor of the website clicks on an item of the calendar, detailed information (such as location, contact person, etc.) about the event is displayed. It is also possible to provide a feature for the owner to define different groups (user context) who may access different categories of the calendar. For example, assuming Alice is the owner of such a calendar. As a family member, her sister Karen can see "family activity" in detail but not the detailed information of events in other categories. This strict definition of groups is useful, but it does not fully satisfy Alice's needs. To make the calendar more useful, some undefined visitors should also be allowed to see part of her calendar. Consider the following two scenarios:

- Bob, who is one of Alice's colleagues, can check her schedule and see the details of her "work activity". Carl, who is Bob's friend, hopes to make an appointment with Alice for some business discussion.
- Donald, who is Alice's travel agent, can check her schedule for the arrangement of a family vacation. Edward, who works for the car rental company which is a business partner of Donald's agency, needs the information regarding the family's arrival time.

The normal action for Carl is to ask his friend Bob to make the appointment for him. He may also write to Alice directly. This requires some amount of interactions between Carl and Bob, and may also involve Alice directly or indirectly. It will be more convenient if Carl can inherit some access right from Bob, who is Alice's



Reseach Meeting

Wednesday, December 20 2006, 8:00am - 5:00pm
Web Research Group Meeting
Location: Room 3530
Contact: Prof. Shen

Family Dinner

Wednesday, December 20 2006, 5:00pm - 6:00pm
This is my Family Dinner
Location: Ground Floor
Contact: Michelle

Research Meeting

Wednesday, December 20 2006, 8:00am - 5:00pm
Web Research Group meeting

(a) Without Privacy Management

(b) With Privacy Management

Figure 1: Web Calendar Example

colleague, and can check Alice's calendar directly for her "work activity" items when he visits her website. Donald (the travel agent) and Edward (the car rental agent) are in a similar situation; they should have the right to see the "family activity" category, but not the "work activity" category. Moreover, Alice's calendar can be checked by Donald and Edward based on additional context: Alice might only allow Donald to check her calendar after the final arrangement of her trip is settled and before the end of her trip (time context); and Edward is only allowed to check Alice's calendar information related to Edward's city (location context) and during the trip (time context, inherited from Donald).

It is hard for Alice to assign a special group and access right to every potential user for different calendar categories. It is not possible to assign an access right to someone whom Alice does not even know, such as Carl and Edward. But a "trust network" can be used to derive specific access rights when needed. The network is a directed graph which represents the trust relationship among users in it. During the query process, some *private data owners* (PDOs) might be willing to share their private data with *private data requesters* (PDRs) through the network. We note that the trust relationship is transitive; i.e., Alice trusts Bob and Bob trusts Carl implies Alice trusts Carl to a certain extent. It is also directional; i.e., although Alice trusts Carl by implication, Carl may not trust Alice regarding his private data. Since the trusted PDR through transitive trust relationship might have less access right (Edward does not have the same right as Donald has in the above example), the information released to indirectly-trusted PDRs may need to be obfuscated according to the level of trust. The trust network therefore requires:

1. Trust relationship defined by PDOs

2. Obfuscation (Web data annotation) rules defined according to the nature of private data

With the help of obfuscation rules, the access right is no longer binary ("yes" or "no"). The access right for a private data item is considered a PERMISSION VALUE, which represents how much detail the private data item can be given to the PDR based on the level of trust. Figure 1(b) shows the result when Carl looks at Alice's calendar when he visits the website. From the figure we can find out that Carl can only see the "work activity" and for the "family activity" Carl only knows that Alice is busy. The ability to control the sharing of private data makes life easier since Carl does not need to ask Bob, who is Alice's colleague, to help checking Alice's calendar.

In this paper, we are not focusing on how to define Web data in various levels of obfuscations. We solve the problem of assigning data access permission values when there is an existing social network. The contributions of this paper include the construction of a trust network from existing social networks. This network can be used to manage the sharing of private data in the Web environment. This trust network concept may be applied to data sharing in other ubiquitous computing environments.

The rest of the paper is organized as follows. Section 2 describes the related effort in improving privacy management. Section 3 describes how to bootstrap the trust network from an existing social network. Using the Web calendar as a case study, Section 4 demonstrates the process of trust network initialization and data sharing with obfuscation rules. A framework on how the components of the system to manage private data sharing can be implemented is given in Section 5. Section 6 summarizes the experiments we have done using an existing social network (MSN.com) to study the characteristics and significant issues of the trust network. Section 7 discusses possible refinements for the permission assignment techniques. Section 8 contains the conclusions and future work.

2. RELATED WORK

In order to identify Web users and their relationship with others, the Friend of a Friend (FOAF) [2] project creates a set of machine-readable pages describing people, the links between them, and the things they create and do. This could be the basis to construct trust networks by bootstrapping from existing social networks.

Much of the fundamental work in the analysis of social networks and the major advances in the past century have been carried out in the fields of sociology, psychology, and communications [8, 22]. The first step to facilitate social networking is to have a definition of trust that captures the social features for both local and global scopes [24]. Trust management is quite well studied in P2P systems and semantic Web [13, 14, 16, 23, 24]. In [14], a definition that captures the nature of social trust relationships and an algorithm are proposed for computing the trust value in social networks using default logic. Kamvar *et al.* proposed EigenTrust for reputation management for file sharing in P2P systems [13]. Richardon proposed a trust value computation method using probability theory in global belief combination which can provide each user a personalized set of trust values [16]. Trust propagation is another important research topic. Guha *et al.* proposed a method for predicting trust between users [10]. The trust acquisition and propagation model is discussed in [5, 6, 25]. However, the relationship between trust and online private data is not well addressed.

The online data privacy problem has been noticed for quite a long time. The Platform for Privacy Protection (P3P) Project [21] of the World Wide Web Consortium (W3C) is a method for websites to publish their privacy policies. The APPEL language [20] works with P3P and enables users to exchange privacy preferences according to published privacy policies. P3P has not yet received much acceptance from Web users mainly due to its lack of enforcement, since current implementations do not include compliance of user preferences. Kolari *et al.* have pointed out that an enhanced P3P based on the Rei language can provide an improved trust model [15].

A lot of research has also been done on statistical databases to protect privacy through query restriction, data perturbation and output perturbation [1]. Such research focuses on hiding the relationship between the identity of the PDO and relevant private data [11, 18]. An example is that instead of giving the application an exact location, a regional context is used to satisfy the K-anonymity requirement. A list of candidates is returned to obfuscate private data [17]. There has not been much attempt to connect this approach with access control rules.

Since P3P does not provide any mechanism to ensure that these promises are consistent with internal data processing at the website, a purpose-based access control method can be used as an extension of P3P [4]. To address this issue we have proposed to extend the P3P protocol, which is a W3C recommendation for Web applications. We have successfully applied this extension to some context-aware applications [12]. But the extension did not consider transitive trust relationships. PDOs still need to specify every potential PDR's access right based on the categories defined by P3P, which makes management of private data cumbersome.

3. FROM SOCIAL NETWORK TO TRUST NETWORK

In the Web-based social network, the PDOs need to have some control on the management of their private data. However, it is not practical for a PDO to set a particular permission value for each private data category for every potential PDR. The role-based access control (RBAC) has partially solved the problem [7]. In this approach, it is required to define all the potential users' into some groups. For example, in the UNIX file system, the file owner (user) can give each role (user, group, and other users) some specific permissions (read, write, execute). With the role-based access control, a PDO needs to define the permissions based on the roles of PDRs. But it still may be difficult to define the role of every potential PDR. Therefore it will be very nice if a transitive trust relationship exists among the potential PDRs.

It turns out that the transitive relationship does exist in our daily life. For example, if Carl wants to know how much is the toll to travel through the Cross Harbor Tunnel in Hong Kong, he may ask his friend Bob about it. If Bob does not have the answer, he may continue asking his friends by phone calls or by emails. Later from Alice, Bob finds out the toll charge and passes the information to Carl. Formally speaking, this transitive query continues until a satisfactory answer is obtained and returned to the originator along the query path.

When each person who is willing to share data in the community is represented by a vertex, and when how much a PDO trusts a PDR is represented by an edge, the whole community becomes a trust network. When users share private data in a community, the access decision is based on the trust relationship between the PDO and the PDR in the trust network.

DEFINITION 1. *The* TRUST *relationship between a PDO and one of its contacts is a permission value assigned by PDO to a potential PDR:*

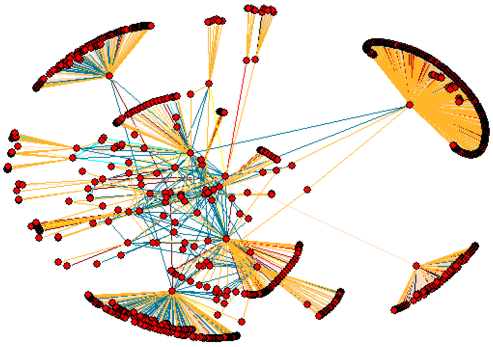


Figure 2: Facebook Network Example (1190 nodes). The data includes friends of ID 655183482 and friends of friends.

$$permission = trust(a, g, c)$$

Where a is the PDO involved, g is a member within a group of contacts that the PDO has defined, and c is the context where the permission value applies. The context in Definition 1 provides the application developer and the PDOs the ability to set the constrains in data sharing. Context may be related to the time, location, nature, etc. of an event. For example, Alice only allows Bob to view his calendar on her "working" activity. The event type "working" in the calendar can be considered as one context. It is extensible based on the needs of the application or the PDOs.

PDOs are requested to define data access permissions for all the direct users using their privacy preferences. The permission value can be a decimal number ranging from [0,1], where 1 represents total trust and 0 represents no trust at all. The 0 permission value is seldom used in online social networks because a PDO joins the network for the purpose of sharing data with friends there. The context in Definition 1 refers to the particular situation a permission value is assigned. The context includes time context, location context, and query context (such as purpose, retention, etc.) When Web data annotation is available in the social network, the annotation can also be part of the query context. For each kind of private data, the PDO can define several permission values to fit different contexts. A GROUP represents a group of PDOs who share the same permission value. A group can either be defined by a third party or by a PDO. One of the most popular Web-based social networks, Facebook, allows users to create private groups or to join the existing regional or alumni networks. Figure 2 shows "my friends" and "friends of my friends" relationship on Facebook for one of the authors. We can see that the relationship has been defined between Facebook users through the profile. When a PDO assigns his friends the permission which can be written in a preference file, the network becomes the trust network. The preference file can be stored as a single document or attached to the private FOAF document [2]. The trust relationship described above only supports the direct relationship. In the Web calendar application, the transitive trust relationship also needs to be considered. Carl, who is not directly connected with Alice, links to Alice through Alice's colleague Bob. In order to achieve this, we define a new operation JOIN.

DEFINITION 2. TRANSITIVITY *determines whether a trust relationship can be extended outside of the directly-connected PDRs. A propagated trust (Ptrust) relationship based on transitivity can be used to extend the relationship to other users. The* JOIN *opera-*

tion shows that the trust relationship is transitive; that is, if PDO A trusts PDR B, who in turn trusts PDR C, it implies that PDO A Ptrusts PDR C .

$$\forall a : \text{PDO}, i, j : \text{GROUP}, c : \text{CONTEXT}, \exists interim \in i$$
$$trust[a, i, c] = p_1, trust[interim, j, c] = p_2 \Rightarrow$$
$$Ptrust[a, j, c] = trust[a, i, c] \bowtie trust[interim, j, c] = min(p_1, p_2)$$

With the JOIN operation, the permission propagates along the trust network with the maximum possible value. Every potential PDR can be assigned a permission value automatically if he is within the community or from a related community. In a real application a PDO might set more restricted access. Additional operations will be proposed in the future.

4. TRUST NETWORK AND OBFUSCATION

Privacy management is separated into four steps: context pruning, transitive trust network initialization, permission value computation and data obfuscation. The four steps are applied when appropriate. In this section, we use the example when Edward sends a query on Alice's "family activities" in the calendar application to demonstrate these four steps.

4.1 Context Pruning

In a Web-based social network, users are allowed to define a lot of relationships with other users. For example, in Facebook users can define every relationship with all friends, such as "We went to school together" or "We took the course together". Moreover, a user can further specify which school and which course to establish the link between two users. As a result, group definition is quite complicated. Each PDO might need to define permission values for an individual person or a group based on different contexts.

The goal for context pruning is that trust relationship only propagates within the same group of people. For example, Alice would like to share her "work activity" with Bob. But she may not wish to share the information with Bob's family doctor, whom Bob trusts totally. Therefore the trust network should be restricted by context. We zoom in Figure 2 and extract part of the real Facebook network as shown in Figure 3. The church events in the calendar can be exchanged among all members of this network since all these five people are from the same church "CBIBC". But the work event is just shared between Michelle and Cammy since they "worked together" and no other user in the network has a similar context. Here "church" or "work", which might be an attribute of the event, can be considered a context.

Suppose there are two groups of users trusted by a PDO and a PDR is in both of the groups. If the PDR requests information from the PDO then it might be reasonable for the PDO to provide the larger permission value derived from each of the two groups. Another task for context pruning is to find out the maximum permission value for every direct trust relationship on the condition of satisfying the context requirement.

In the previous example, during the trip time the travel agent Donald is trusted by Alice based on RECIPIENT "ours"(see the definition in [21]). For other times, since the query context is not satisfied, Donald is not trusted.

4.2 Transitive Trust Network Initialization

Even if a PDO defines only a small portion of the whole community, data can still be shared based on the PDO's preferences. The users a PDO trusts may also have their own trust relationships (e.g., Donald trusts Edward due to partnership). We need to merge all the relationships together to build the trust network. For the example discussed in 1, after context pruning we know the direct trust relationships form a tree. Figure 4 shows the result after merging all direct trust relationship trees of Alice, Bob, Edward and Donald.

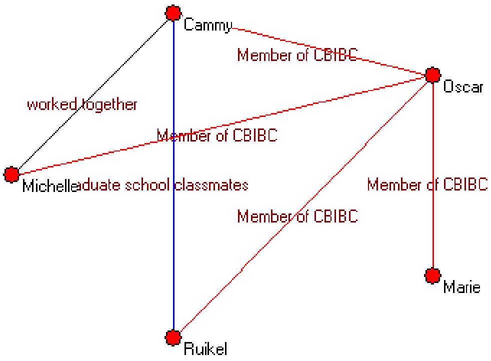


Figure 3: FaceBook Multiple Relationship Example.

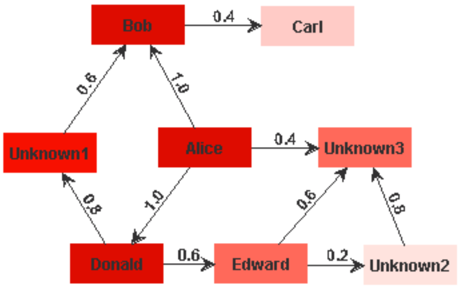


Figure 4: Trust Network- Transitive Relationship

DEFINITION 3. *In a trust network, the hops of a PDR is the number of vertices to traverse along the shortest path from the PDO to this PDR.*

Even if the complexity of privacy preference files has been decreased by using group-based permission assignments, to define the permission values of every potential PDR is still plenty of work. With the transitive relationship, a PDO only needs to define the permission values of those PDRs who have a "close" relationship, or are directly connected in the trust network. Based on the privacy preferences defined for each of these PDRs, the trust relationship can be computed and propagated to the rest of the trust network. Since there are various types of private data on the Web, we need to consider the data categories, sharing contexts during the trust network merging process.

4.3 Permission Value Computation

Note that to apply the transitive relationship, all the trust relationships during the propagation process need to have the same context. Before computation of the permission value for a PDR, context pruning will ensure the network initialized in 4.2 is extendable.

Algorithm 1 can be applied to implement the JOIN operation in order to compute the shortest path from the PDO (source) to a PDR (destination). Given a social network graph $G(V, E)$, where V is the vertices set and E is the trust relationship set. $p(u, v)$ is user v's permission value given by user u. $Extract\_MAX(Q)$ is used to extract the vertex with the maximum permission value which is not in the finished set S. Through Algorithm 1, a user can get the most private data from a PDO based on the permission value assigned. Algorithm 1 is only one simple and possible solution to compute the permission value. The pageRank [3] or Max-Flow might be used to defined and compute the Ptrust.

Algorithm 1 Permission Value Computation

```
Input:A weighted directed graph G(V,E)
edge weight, p(u,v), is the permission from u to v
PDO, PDR
Output:Permission Value
 1: for all vertex v in V do
 2:     permission[v]=0
 3:     previous[v]=undefined
 4: end for
 5: permission[PDO]= 1
 6: S= empty set
 7: Q= V[G]
 8: while Q is not an empty set do
 9:     u= Extract_MAX(Q)
10:     if u equals PDR then
11:         return permission[u]
12:     end if
13:     S= S union u
14:     for all edge (u,v) outgoing from u do
15:         if $min(permission[u], p(u, v)) > permission[v]$
            then
16:             $permission[v] = min(permission[u], p(u, v))$
17:             $previous[v] = u$
18:         end if
19:     end for
20: end while
```

When Algorithm 1 is applied to Figure 4, it first puts Donald and Bob into the waiting queue Q. Then the $Extract\_MAX$ function extracts Donald from the queue and puts Edward and Unknown1 into Q. Then Bob is extracted and Carl is put into Q, too. The $Extract\_MAX$ function processes Unknown1 and Edward in order. When Edward is handled, the algorithm knows Edward's permission value. Therefore the trust is propagated from Alice to Donald and finally to Edward. We compute the permission value of every potential user (all users except Alice herself), and use a gradient color to represent the value as shown in Figure 4. The darker the vertex's color, the higher permission value it holds. We can see the effects of trust propagation by the changing color shades.

4.4 Data Obfuscation

There are lots of data items that can be represented in a hierarchical way. For example, the "current location" is a frequently-used private data in different applications. Room 4208, Floor 4, HKUST, Hong Kong, China is a common address to define a location precisely. To protect privacy, for some PDRs in some applications, a PDO may want different information shown on the PDR's screen. Detailed information (room number, etc.) is given to close friends and general information (Hong Kong) is given to unknown PDRs. Based on the transitive relationship, the permission value can be used to control the degree of obfuscation for a certain private data item based on either the default value or the user's preference.

From Figure 4, we see that when the trust network becomes complex it is quite possible for an unknown PDR to obtain private data after several data passing actions. In order to make sure the private data passing scale is controllable, a PDO needs to set up some control factors:

1. Maximum propagation hops, hop_{max}: how many hops private data can be passed along the network. This is helpful to stop data propagation to PDRs who are too far away.

2. Damping factor, ϖ: How much data is obfuscated through every hop. This method gradually reduces the information available and makes sure that an unknown PDR cannot get too much detailed information through several trustable intermediary users.

Therefore we can replace line 15-18 of Algorithm 1 by:

```
if $min(permission[u], p(u, v)) \times \varpi > permission[v]$
$hop[v] \leq hop_{max} \wedge$ u is not PDO then
    $permission[v] = min(permission[u], p(u, v)) \times \varpi$
    $previous[v] = u$
end if
```

With the help of hop_{max} and the damping factor ϖ, the private data is controlled to spread only within a certain number of hops. Moreover, the farther a PDR is away from a PDO, the less private data he receives. For the previous example, Edward can know Alice is in HKUST without the damping factor. And if the $\varpi = 0.7$, then permission for Edward is 0.42. Edward can only know that Alice is in Hong Kong. The permission values might be hard for PDO to understand. It is helpful to visualize the social network by painting users in the network with colors of different shades based on the permission values assigned as shown in Figure 8. And it is also very helpful to assign the critical person, who has lots of connection the PDO are not familiar with, a sharp ϖ in order to keep the data private.

5. FRAMEWORK OVERVIEW

In the Web calendar example, we use the Privacy Server framework as shown in Figure 5. The PDOs define their private data through the PDO Preference Manager and store their preferences in the PDO Preference Database. When there is a data query initiated by a PDR, the Private Data Query Adapter acts as an interpreter for the query and sets up the trust network based on PDO's preference definition. The data query should include all the context information (e.g., the reason to access the data, how to forward data to third parties and application user name). With the PDO information from Context Database, the Adapter computes the permission value based on PDO's preference and passes the value to the Obfuscation Manager. A fuzzy result is returned based on applicable obfuscation rules and the permission value.

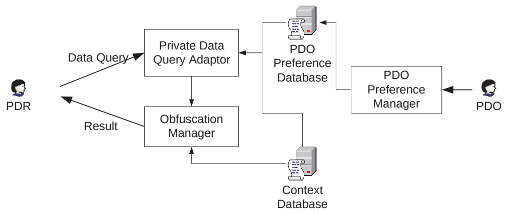


Figure 5: Privacy Server Implementation Framework

A trust network is set up based on the transitive relationship defined by each PDO, which is derived from the online community information. Users in a whole community who are willing to share private data become vertices in the network while the trust relationship between each other becomes edges. The strength of the trust relationship becomes private date permission value which denotes the edge weight in the trust network. Since the trust relationship is asymmetric, the whole trust network is a directed graph. When there is a private data query, the problem becomes the checking of whether there is a path from a vertex (PDO) to another vertex (PDR) in a directed graph.

After the permission value is obtained, the Obfuscation Manager blurs the private data according to the value and still returns some information to the PDR (unless the permission value is zero, indicating that the PDR is forbidden from accessing the data). Context data can often be represented in many ways and forms. For example, the location context can be represented at a particular point geographically, or in regions of various sizes which contain that point. Alice's location, in the previous example, could be represented as <Alice, at, Cross Harbor Tunnel, Hong Kong, China>, showing that Alice's location information at a certain time is one of Cross Harbor Tunnel, Hong Kong and China depending on the permission value. The Obfuscation Manager returns different results for different queries based on the relationship between the PDO and the PDR.

The transitive relationship and obfuscation rules break the current binary private data access characteristic and make context sharing easier. We modify the Web calendar component, JEvent, and build the Privacy Management Framework for it as shown in Figure 1. Figure 1(a) is the original JEvent service. Users are allowed to check all the detailed calendar information by clicking on the event. With the privacy management as shown in Figure 1(b) only the registered users can check the calendar and the "Family activities" is not available based on the data category the calendar owner (PDO) has defined. The successful hacking of the code for JEvent shows that the transitive trust relationship does work in a real application.

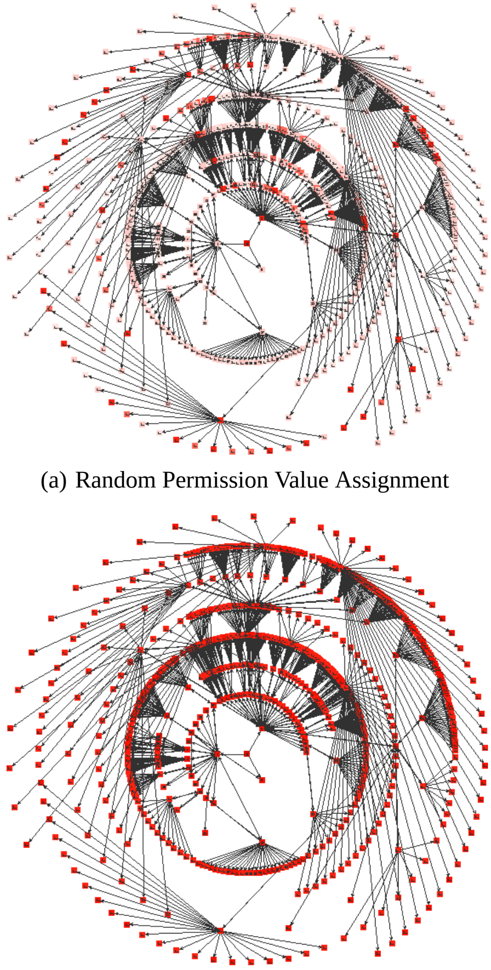
This framework is not specially defined for the Web calendar; other applications can also connect to the Privacy Server through an HTTP connection for the current CGI version.

6. EXPERIMENT

6.1 General Characteristics of the Trust Network

Our study is focused on the "trust network" where edge (u, v) means u trusts v with a labeled permission value. There are lots of online communities available currently, such as MSN, Facebook, Blogger, etc. We picked MSN due to its popularity to test the implementation of permission value assignment scheme. Starting from one of the authors' friends who posts her friends list on the Web[1], we used a crawler to trace the friends lists. We visited 187 users who are connected with the friend within four hops and obtained other 1,181 related users. None is more than four hops away from the friend. Since there was no permission value currently supported by MSN, we randomly assigned different permission values for every relationship.

Figure 6 contains the trust propagation results after we randomly assigned permission values using Math.random (range [0,1)). The permission values became very small after four hops as shown in Figure 6(a), since most peripheral nodes are in light color. If these peripheral nodes wish to see the central node's information, their requests will not be successful. Since friend lists on MSN are defined by the users manually, the trust relationships should be higher than random assignments in the range of [0,1). By changing the range to [0.6,1), the results are shown in Figure 6(b). Compare Figure 6(a) and Figure 6(b), we see that the colors in Figure 6(b) are darker, which means that the permission values are higher after trust propagation when higher permission values are assigned initially. Therefore the permission values defined by PDOs are indeed affecting the private data propagation process.


(a) Random Permission Value Assignment

(b) Assign High Permission value for Relationship

Figure 6: Transitive Network Efficiency

6.2 Control Factors

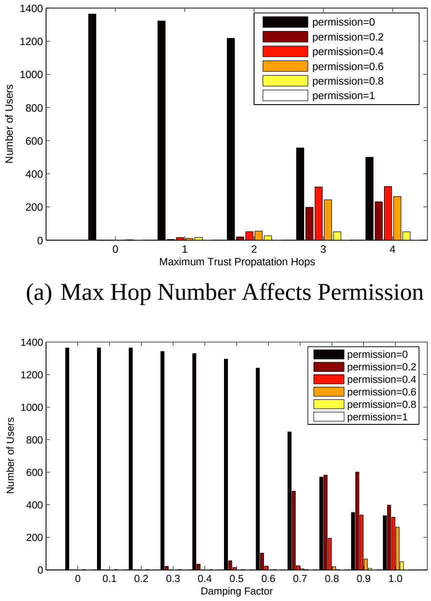
Figure 6 demonstrates that it is possible to construct a trust network from an existing social network for managed data sharing, if the social network supports the setting of permission levels. We then explore how a PDO can control the transitive relationship with partial trust.

The maximum number of hops hop_{max} can be set by a PDO in order to control how far the private data can be forwarded. We again use the MSN social network as a test base. We randomly assigned permission values to every trust relationship and then kept this directed graph unchanged in the following experiment.

When no transitive relationship was allowed ($hop_{max} = 0$), 1,350 queries got no permission during data sharing in Figure 7(a). When the transitive relationship was allowed, the non-empty query number was dramatically increased when $hop_{max} = 3$. This is

[1] MSN URL:http://rp20040619.spaces.live/friends

because there are few users on the first one or two hops of the trust network. The bigger hop_{max} was, the more detailed result could be obtained. Moreover, we noted that blanket permission was not granted since only a small number of queries could get access to the private data. We can also see that even if the friend has only defined three close friends, if she allows three hops of data sharing, then around 700 users can see her obfuscated data.



(a) Max Hop Number Affects Permission

(b) Damping Factor Affects Permission

Figure 7: Control Factors

Figure 7(b) demonstrates how the damping factor discussed in section 4.4 affected the permission value. If the damping factor ϖ is zero, it meant that there was no transitive relationship. If ϖ is very small (e.g., 0.1 or 0.2), it strongly restricted the access permission of private data. Even when ϖ became 0.6, most users got permission value less than 0.2. When ϖ became bigger, the influence of ϖ significantly affected the permission value to access private data.

We understand that the number of users who get permission might be different due to different social network topologies. For example, if the PDO defines a lot of close friends, there will be a number of users who get permission to access private data even when $hop_{max} = 0$,. The selection of ϖ and hop_{max} will indeed affect the topology of the trust network. But the trend of trust propagation will not change too much. In practice the damping factor should be used with maximum hop number together in order to achieve the desired access control. Moreover, the PDO can set up different damping factors to different groups or specific users if he wishes.

7. DISCUSSIONS

7.1 Trust Priority

In a trust network, it is possible that a PDR may obtain more private data through transitive relationships. For example Figure 8 is the result after running Algorithm 1. The gray line represents the trust propagation path when Unknown3 queries Alice's information. Through a full transitive relationship, Unknown3 can get 0.6 permission value through the path: Alice $\rightharpoonup$ Donald $\rightharpoonup$ Edward $\rightharpoonup$ Unknown3. However, Unknown3 is directly defined in Alice's trust tree with permission value 0.4 (green line). There is now a conflict between the direct trust and trust derived from transitive relationships. Since trust based on multiple recommendations from a single source should not be higher than that from independent sources, if the PDR is one of the directly-connected vertices with the PDO then the permission for this PDR cannot be higher than the permission value originally assigned by the PDO.

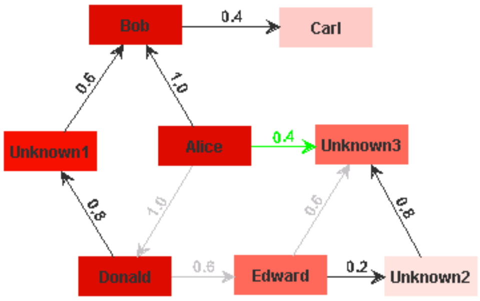
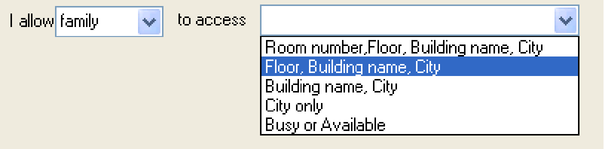


Figure 8: Trust Priority: Directed vs. Transitive

7.2 Standardized Private Data Levels

It is often hard for users to assign accurate decimal permission values to others. Therefore, we should provide a visualization of the data. The user can directly select the data level they would like to share and the program can easily convert the level into a decimal number.


I allow family to access

Room number,Floor, Building name, City
Floor, Building name, City
Building name, City
City only
Busy or Available

The levels of a private data item are either defined by a PDO or by a public ontology. Then different PDOs might have different data levels in real applications. For example, Alice defines her location in 5 levels, such as "Room 4208, Floor 4, HKUST, Hong Kong, China". Her secretary only uses "HKUST, Hong Kong, China". When the secretary grants Bob with permission value 0.67, Bob can only know "Hong Kong". If Alice gives a 0.8 permission value to her secretary, then with the transitive relationship Bob gets 0.67 (the maximum of 0.8 and 0.67) permission value and consequently a more specific area name (HKUST) of Alice's location. This could be a big privacy hole.

A possible solution is that for each category a standardized private context data level is set up and shared by all PDOs through a separate central information directory which provides all kinds of information level descriptions. The PDO Preference Manager connects to that directory and automatically helps users to search what other preferences the PDO has defined. Initially, there are only a few default levels for the data. When a PDO wants to have more specific context levels, he can insert a level himself and record the new level in the central information directory. For example, if Alice wants to identify the current building as a new context level, she finds that this information is between the floor information and the area name. Alice can then insert the building name between them and set the permission value for this new context level to be ($\frac{0.6+0.8}{2} = 0.7$). When other PDOs define their location information, this new level can also be used by them. Since the permission value is a decimal number between 0 and 1, an infinite number of context levels can be supported. Another advantage of using standard levels is that a PDO can see and directly choose the information level he wishes to share with other users instead of assigning a permission value which may not be meaningful to the PDO.

7.3 Other Applications

With the development of ubiquitous computing, more and more private data is available to the public either on the Web or through other applications. For example, a mobile service provider has already started friend location service. Users can dial a special number to trace friends' location. Users might lose privacy control in that situation because he may not know what information about him is shared, compared with the social network situation that the user is the publisher of his own data on the Web. It is possible that through such a service, a thief can find out a user's regular schedule, such as the time to go home, by tracking the user's location for a period of time before breaking into his home when he is not there. The convenience of ubiquitous computing applications will not be enjoyed unless users can control what private data to share with whom at what time. Our privacy server framework can be helpful in these applications.

8. CONCLUSIONS AND FUTURE WORK

In this paper we propose a transitive trust network for private data sharing in social networks. Private information can be shared through the trust network. We use a Web calendar application to show the process of using trust network algorithms to share data. We finally demonstrate the feasibility of constructing the trust network from an existing social network. The characteristics of such a trust network are analyzed which may be applied to data sharing in ubiquitous computing environments. We plan to launch the Web calendar service with trust network and collect data for further development. We shall also develop plug-ins and propose to owners of social networks that users be allowed to use them to assign permission values to their contacts.

Acknowledgement

This research is supported in part by Hong Kong Research Grant Council Project 619507.

9. REFERENCES

[1] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Survey*, 21(4):515–556, 1989.

[2] D. Brickley and L. Miller. Foaf project. http://www.foaf-project.org/, 2007.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[4] J.-W. Byun, E. Bertino, and N. Li. Purpose based access control of complex data for privacy protection. In *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110, New York, NY, USA, 2005. ACM Press.

[5] M. Conrad, T. French, W. Huang, and C. Maple. A lightweight model of trust propagation in a multi-client network environment: To what extent does experience matter? In *ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*, pages 482–487. IEEE Computer Society, 2006.

[6] B. Esfandiari and S. Chandrasekharan. On how agents make friends: mechanisms for trust acquisition. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies 2001*, pages 27–34, 2001.

[7] D. F. Ferraiolo, J. F. Barkley, and D. R. Kuhn. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security*, 2(1):34–64, 1999.

[8] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3(1), June 1997.

[9] Google. Google calendar. www.google.com/calendar.

[10] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM Press.

[11] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *VLDB*, pages 9–16, 2006.

[12] D. Hong, M. Yuan, and V. Y. Shen. Dynamic privacy management: a plug-in service for the middleware in pervasive computing. In *MobileHCI 2005*, pages 1–8, Salzburg, Austria, September 2005. ACM.

[13] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[14] Y. Katz and J. Golbeck. Using social network-based trust for default reasoning on the web. Submitted to Journal of Web Semantics, 2007.

[15] P. Kolari, L. Ding, S. G. A. Joshi, T. Finin, and L. Kagal. Enhancing web privacy protection through declarative policies. In *POLICY '05: Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, pages 57–66, Washington, DC, USA, 2005. IEEE Computer Society.

[16] R. Matthew, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, 2003.

[17] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, pages 763–774, 2006.

[18] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[19] Technorati. State of the blogosphere / state of the live web. http://www.sifry.com/stateoftheliveweb/, 2007.

[20] W3C. A p3p preference exchange language 1.0 (appel1.0). http://www.w3.org/TR/P3P-preferences/.

[21] W3C. Platform for privacy preferences (p3p) project. http://www.w3.org/TR/P3P/, April 2002.

[22] S. Wasserman and K. Faust. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1994.

[23] C.-N. Ziegler and G. Lausen. Analyzing correlation between trust and user similarity in online communities. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proceedings of the 2nd International Conference on Trust Management*, volume 2995 of *LNCS*, pages 251–265, Oxford, UK, March 2004. Springer-Verlag.

[24] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, pages 83–97, Washington, DC, USA, 2004. IEEE Computer Society.

[25] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, pages 83–97, Washington, DC, USA, 2004. IEEE Computer Society.