JES
Journal of Engineering Sciences

Vol 10, Issue 9, Sept /2019
ISSN NO: 0377-9254

TRUST-BASED COLLABORATIVE PRIVACY MANAGEMENT IN ONLINE SOCIAL NETWORKS

[1]A.Emmanuel Raju ,[2]k.Samson Paul, [3]Dr.C.Gulzar

[1]Assistant Professor , [2]Assistant Professor ,[3]Associate Professor

DEPARTMENT OF CSE

Dr.K V SUBBA REDDY INSTITUTE OF TECHNOLOGY, KURNOOL

ABSTRACT— In this paper, we propose an approach which is intended for addressing the problem of collaboratively deciding privacy policies for, but not limited to, shared photos. Our proposed algorithm utilizes trust relations in social networks and combines it with the Condorcet preferential voting scheme. An optimization is developed to improve its efficiency. Experimental results show that our trust-augmented voting scheme performs well. An inference technique is introduced to infer a best privacy policy for a user based on his voting history. Online social networks Have now turn out to be the most popular structures for humans to share statistics with others. Along with this, there's a critical threat to individuals' privateness. One privateness chance comes from the sharing of co-owned statistics, i.E., while a person stocks a records object that involves a couple of customers, some customers' privateness can be compromised, on account that specific users commonly have one-of-a-kind reviews on who can get entry to the records. Moreover, the user could make a tradeoff among facts sharing and privateness maintaining through tuning the parameter of the proposed mechanism. We formulate the choosing of the parameter as a multi-armed bandit problem and follow the higher self belief bound policy to solve the problem. Simulation results exhibit that the agree with-based totally mechanism can encourage the consumer to be considerate of others' privacy, and the proposed bandit approach can carry the consumer a excessive payoff.

KEYWORDS: Social trust, voting scheme, multi-armed bandit, collaborative privacy management, online social networks.

I. INTRODUCTION

Social networking is one of the greatest inventions on the Internet during the last ten years. Social network sites provide users platforms to socialize both in the digital world and in the real world, for making friends, information exchange and retrieval, and entertainment. Some of the largest ones, such as Facebook [1] and MySpace [2], provide services to hundreds of millions of registered users. However, partly due to the intention to attract as many users as possible for their commercial success, social networks tend to intentionally or unintentionally expose private information of existing users. Privacy is becoming an important and crucial research topic in social networks. A number of scholars have studied it from different viewpoints, e.g. [3–7]. Moreover, the excessively expanded number of users also bring difficulties into the management of these sites, so that designing effective mechanisms to coordinate users' opinions over their privacy becomes an emerging issue.

As a shared platform, resources in a social network may be co-owned by a number of users. For instance, documents can be co-authored and several users may appear in a same photo. Such co-ownership might cause breach of privacy.

Voting is a natural choice to build a mechanism which takes individual's preferences on their privacy policies into a joint decision reflecting the "general will" of the group of people who are sharing a piece of data. Siquicciarini et al. [8] propose a game theoretical method based on the Clarke-Tax mechanism [9], which can maximize the social utility function by encouraging truthfulness among people in the group. This induces a nice property that the final decision cannot be manipulated by individuals,

as users express their true opinions about the privacy preference. However, their method is not as simple as it is claimed to be, as it requires each user to compute a value for each different preference and the user-input values are essential for their method to derive a joint decision. We argue that this requirement is not realistic and it makes users less interested in participating collaborative privacy control.

II. RELATED WORK

A number of studies have highlighted privacy issues in OSNs. Luders et al.'s [6] study of primarily Norwegian users on Facebook has shown that users' knowledge on how social media functions in regards to use, disclosure and transfer of their personal information is largely inadequate. Majeski et al. [10] found that, in their study, every one of the participants had at least one sharing violation based on their stated sharing intentions. Our recent exploratory study [4] of four OSNs examined has also highlighted the general disconnect between privacy policies and privacy controls. All the above studies highlight the need to both empower users to manage their privacy in OSNs and to provide mechanisms that enable novice users or those new to an application (or even OSN) to navigate the plethora of privacy settings. The CPM framework proposed in this paper responds to both these challenges by offering fine-grained access control over personal data shared by OSNs with TPAs and enabling the OSN users to jointly identify "best fit" privacy configurations for such applications.

PoX [7] is a client-side browser plug-in that acts as a proxy between TPAs and Facebook. In order for users to benefit from PoX, a TPA must use the PoX library to channel all requests to Facebook. Similarly, xBook [12] introduces a secure hosting platform; developers deploy and host their TPAs on this platform which intercepts not only interactions between the application and the OSN but also with other web sites. Both these works contrast our implementation which simply intercepts all the calls to the Facebook Graph API. Most significantly, they do not support sharing and reuse of privacy configurations for TPAs. MockDroid [5] is a modified version of the Android Operating System, which enables the user to create access templates for each application or just accept/deny access to a resource at run-time. This is similar to the fine-grained privacy control offered by the CPM framework. However, unlike CPM, sharing and collaborative management of privacy configurations is not supported.

III. SYSTEM MODEL

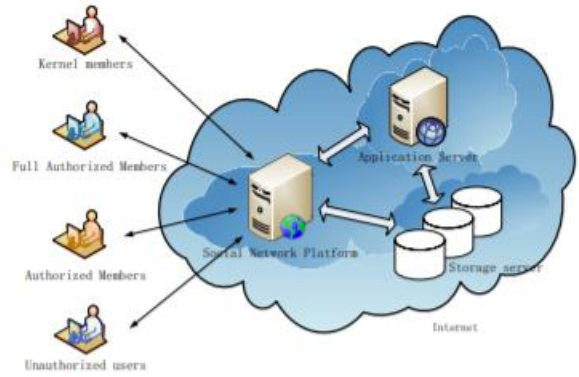


Figure:1.The system architecture for a private OSN

We introduce a generic model to implement above-mentioned collaborative OSN applications. Figure 1 shows a system architecture for our model. In this architecture, a third party is required to be responsible for the web-based applications, as well as the storage of published data. Meanwhile, it also provides some services for users, such as Web browser service. But we do not demand that this third party is credible for a private OSN.

Existing social network sites, such as Facebook, Flickr, and Myspace, even cloud computing platforms are appropriate environments applying our model.

In order to define the range of access control in a private OSN, we classify the users in social networks into four categories:

· Kernel members (KM): can create and manage a special community by collaboration and have rights to publish, delete, access or update resources released by other members of the community;

· Full authorized members (FAM): have full rights to publish and access resources in the community, but

do not have permissions to delete or update resources;

· Authorized members (AM): can access the resources by using her own access permission, but cannot publish these resources;

· Unauthorized users (UU): may not have permissions to access resources published by community members.

PROPOSED SYSTEM

- ❖ In the proposed trust-based privacy management mechanism, we introduce a threshold based on which the user makes the final decision on data posting. Simply speaking, a high threshold indicates that the user has a relatively low tendency to share the data with others, and only when the majority of the involved users or users that are highly trusted agree to post the data; the data can finally be posted. By tuning the threshold, the user can make a trade-off between data sharing and privacy preserving.
- ❖ Considering that a user continually posts data items in an OSN, we model the threshold selecting problem as a sequential decision-making problem. More specifically, the system formulates the problem as a multi-armed bandit problem [9] and apply the upper confidence bound (UCB) policy to solve the problem. Simulation results show that dynamically adjusting the threshold according to the UCB policy can lead to a higher payoff than using a fixed threshold.

COLLABORATIVE PRIVACY MANAGEMENT (CPM) FRAMEWORK

the framework provides an interceptor mechanism that acts as a membrane between the TPAs and an OSN. All information requests pass through this membrane and are intercepted. At the time of installation, the framework makes explicit all the personal data items that an application will access. The user now has the choice to not allow certain permissions or where the permissions are mandatory choose to return dummy data. An example of the latter is the scenario where an application requires access to the user's name to personalise certain elements of the application but also requests access to the user's hobbies or friends list without a clear need for it. The user can now choose to return empty sets or even dummy data so as to safeguard his/her privacy1 . The user can then also choose to share this new privacy configuration with others in the OSN. The user can also change the privacy configuration for the application in due course (e.g., to share more or less data as desired).

Alternatively, when a user installs a TPA, instead of manually configuring the privacy settings, s/he can choose to search for existing privacy configurations for the application shared by other users in the OSN and how they have been ranked by others utilising them. The user can then load a chosen configuration (either by popularity or based on the kind of information s/he might be willing to allow access) and either use it as is or make modifications before deploying it for that particular application. The user can then also share the new configuration with others in the OSN.

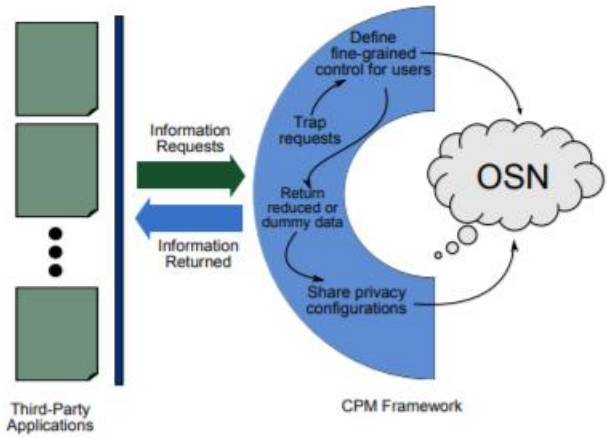


Figure 2: Overview of the Collaborative Privacy Management (CPM) Approach

Trust-Augmented Voting Scheme

In some situations not all voters are equal. Typical examples include decisionmakings in a shareholder's meeting where the weight of each voter corresponds to his volume of share. Likewise in social networks,

users' opinions on deciding a privacy policy do not necessarily carry the same weight. For example, Alice has a picture in which there are Bob, Clare, Danny and Elisabeth, and she wants to publish that picture in her album. She is willing to give right to the co-owners of the picture, i.e. Bob, Clare, Danny and Elisabeth, on deciding whether the privacy level of that picture is P1 or P2 (see their definitions in Sect. 2.2). Here we suppose that Bob is a friend of Alice and Clare is a friend of Bob but not a direct friend of Alice, i.e., Clare is a friend of a friend of Alice. Similarly, Danny is a friend of Alice and Elisabeth is a friend of a friend of Alice. In this case it seems more reasonable to give Bob's and Danny's opinion more weight than Clare's and Elisabeth's. In this section we propose an extension of Condorcet's voting system for weighted votes. The weight of each vote reflects the trust level of the owner of the shared resource having on the co-owners in their votes for setting a privacy policy for publishing the resource.

In the original Condorcet voting system all votes carry the same weight, and the preferential orders can be compared by only looking at the (integer) exponents of p regarding to their likelihoods. In this paper we measure the likelihoods of these orders by allowing discounted votes to reflect the degree of trust of a user by the owner of a resource, so that each vote carries a real valued weight in [0, 1] instead of always being an integer 1.

Trust-augmented Condorcet voting algorithm.

The trust-augmented Condorcet voting algorithm is detailed as in Alg. 2. The whole procedure of calculation is exactly the same as that of Alg. 1, except that now the sequence likelihood (sql) and maximal likelihood (ml) are of real valued type instead of integer type, as shown above. A trust-based voting profile, which includes the preference lists and trust level for each participant, is taken as input by the algorithm (e.g., left part of Fig. 3), while the output, a set of winders, remains unchanged. In Fig. 3, we assume that A, as the owner of the picture, fully trusts himself, i.e., his trust value is 1.0. The trust of A in other participants (0.8 for B and 0.6 for C) is also shown in the table. From the trust-based voting profile, the revised Condorcet weighted matrix is obtained (e.g., middle part of Fig. 3). The weighted matrix is slightly different from that in Example 1 in the way that simply counted (integer) votes are replaced by accumulated trust values throughout the table. From the Condorcet directed graph in the right part of Fig. 3, we can find a unique winner P1, after computing the likelihood of all sequences of nodes in the top level SCC (containing P1, P2 and P3). This can be easily verified since the likelihood of the sequence P1P2P3P4, as calculated as follows, is greater than the likelihood of every other sequence. Note that here we replace the number of votes as integer exponents over "p" and "1 − p" by their corresponding sums of trust values

Algorithm 1 A trust-augmented Condorcet voting algorithm.

```
input : trustvotingprofile : VotingProfile;
output : winners : set ⟨string⟩;
var cwm : double[ ][ ]   init null
    cdg : int[ ][ ]      init null
    tlv : int[ ]         init null
    sql : double         init 0.0
    ml  : double         init 0.0
    ms  : set ⟨string⟩   init ∅

begin
cwm := getCondorcetWeightedMatrix(trustvotingprofile);
cdg := getCondorcetDirectedGraph(cwm);
winners := getWinners(cdg);
(* the rest is the same as Alg. 1 *)
end
```

A Heuristic Algorithm

The algorithm presented in this section provides another way to resolve general ties in the top-level SCC, as well as to reduce the computational time. A comparison between all the algorithms is conducted in Sect. 4. Taking a Condorcet directed graph, first we have the following arguments. Suppose w(Pi , Pj) is close to 0, it is very likely to be the case that the voters are relatively indifferent with respect to the two policies Pi and Pj . Therefore, regarding to Condorcet's assumption, Pi does not have a significant chance to precede Pj in the underlying invisible order. This justifies our choice in the following algorithm to weaken such difference by adding another (reversing) edge in the graph from Pj to Pi . By doing this, we equalize the votes between policies Pi and Pj . Technically, we only apply this operation within the top-level SCC, gradually by starting from the pairs (Pi , Pj) with least w(Pi , Pj), then the pairs with second least weight, and so on. Each time we add new edges it is required to check

whether a set of Condorcet winners have been generated in the new graph. The running time of the algorithm has an upper bound of O(|V | 2), where V is the set of vertices of the Condorcet directed graph, which is much faster than Alg. 2. Nevertheless, the experimental results in Sect. 4 reveal strong similarity with respect to the results of Alg. 3 and Alg. 2, which provides a concrete support to the applicability of Alg. 3.

Algorithm 2 Optimized trust-augmented Condorcet voting algorithm.

```
input : trustvotingprofile: VotingProfile;
output : winners : set ⟨string⟩;
var cwm      : int[ ][ ]      init  null
    cdg      : int[ ][ ]      init  null
    tls      : int[ ][ ]      init  null
    lwes     : set ⟨string⟩ init  ∅
    nodes    : set ⟨string⟩ init  ∅

begin
cwm := getCondorcetWeightedMmatrix(trustvotingprofile);
cdg := getCondorcetDirectedGraph(cwm);
winners := getWinners(cdg);
if winners = ∅ then
   tls := findTopLevelSCC(cdg);
   while true do
      lwes := findLowestWeightEdges(tls);
      addReverseEdges(lwes, cdg);
      nodes := findNodeN-1OutDegree(cdg);
      if nodes ! = ∅ then
         winners := nodes;
         return
      end if
   end while
end if
end
```

Optimized trust-augmented Condorcet voting algorithm. Similar to Alg. 1, Alg. 2 takes a voting profile as input and produces a set of winning privacy policies. The first part of the algorithm is exactly the same as in the above two algorithms. However, if Alg. 2 cannot find Condorcet winners, it will extract the whole toplevel SCC into a subgraph tls. Then starting from the lowest weighted edges, it adds reverse edges into the original Condorcet directed graph cdg, and searches for Condorcet winners in the modified graph. This will repeat until Condorcet winners are found in cdg. The algorithm is guaranteed to terminate before every pair of vertices in the top-level SCC has two connecting edges pointing to each other. Therefore it is bounded by O(|V | 2) where V is the set of vertices in cdg. An application of Alg. 2

IV. IMPLEMENTATION

- **OSN Admin**

In this module, the Admin has to login by using valid user name and password. After login successful he can perform some operations such as View all authorized users, view friend and request and response, view all users and give link to view post access controls to their friends, View All users with weighted by trust value(vote) and view non trusted users(vote is 0) , View All trusted and non trusted user's post (vote is 0), View All shared and not shared Posts and video posts with Votes and Reviews, View All posts with Vote in chart, View All Video posts with Vote in chart, View All users with weighted by trust value(vote) in chart

Friend Request & Response

In this module, the admin can view all the friend requests and responses. Here all the requests and responses will be displayed with their tags such as Id, requested user photo, requested user name, user name request to, status and time & date. If the user accepts the request then the status will be changed to accepted or else the status will remains as waiting.

- **Users**

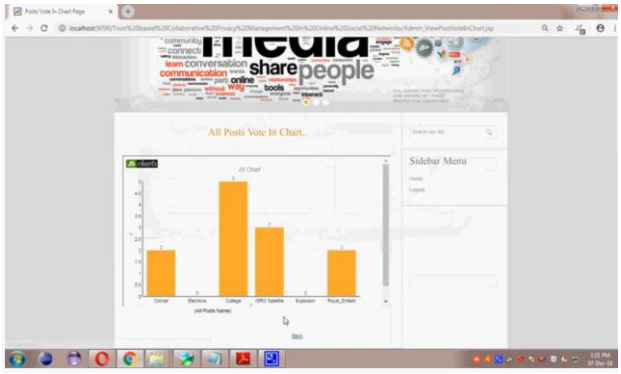
In this module, there are n numbers of users are present. User should register before performing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Verify finger print and Login Once Login is successful user can perform some operations like View your Profile, Search Friends, View Friend Request and Response, View My Friends, Create Post, Create Vidoe Clip Post data, View all your friends and set Access Control, View all your Posts with votes and reviews and share to friends(view all post and give link to share option to all your friends), View all your Video Posts with votes and reviews and share to friends, View all your friends post and give reviews and Vote Option, View all your friends Video post and give reviews and Vote Option,View all your friends(stakeholder) and give vote option to trust

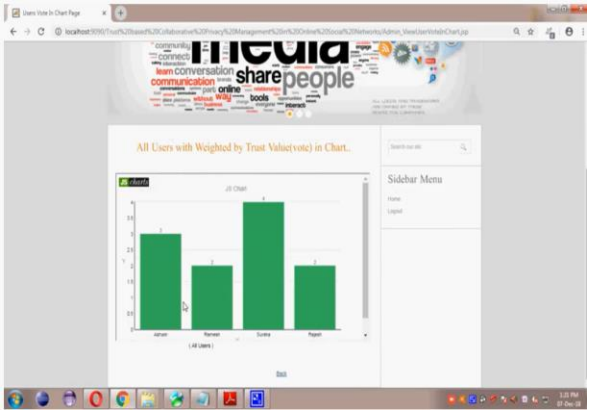
Searching Users to make friends

In this module, the user searches for users in Same Network and in the Networks and sends friend requests to them. The user can search for users in

other Networks to make friends only if they have permission

RESULTS.





V. CONCLUSIONS AND FUTURE WORK

Privacy management in OSNs is an area that is of increasing importance owing to their large user base and the amount of data stored in such OSNs. It is not just sufficient to provide effective privacy control mechanisms for OSNs but, equally or perhaps more importantly, engender trust in such mechanisms. Our proposed CPM framework addresses both these needs by not only enabling users to have fine grained control over their data shared with TPAs but also utilising the social construct of "friends" to identify "best of" configurations that can be trusted by other users. Our work in the future will focus on further evaluating the effectiveness of our framework through larger user studies both on Facebook and (through additional prototypes) on other large OSNs. We also aim to develop techniques/mechanisms to expose the "privacy context" to users as they undertake certain actions on OSNs or TPAs to further improve their awareness about the visibility of their personal data as a result of specific actions on their part

REFERENCES

[1]http://techcrunch.com/2012/02/01/facebooks-s-1-845- million-users.

[2]http://www.symantec.com/connect/blogs/facebook applications-accidentally-leaking-access-third-parties.

[3] Gross, R., Acquisti, A., John Heinz III, H.: Information revelation and privacy in online social

networks. In: Proc. 2005 ACM Workshop on Privacy in the Electronic Society, ACM (2005) 71–80

[4] Ellison, N.B., Steinfield, C., Lampe, C.: Benefits of Facebook "Friends": Social capital and college students' use of online social network sites. Journal of Computer Mediated Communication-Electronic 12(4) (2007)

[5] Rosenblum, D.: What anyone can know: The privacy risks of social networking sites. IEEE Security and Privacy 5(3) (2007) 40–49

[6] Carminati, B., Ferrari, E.: Privacy-aware collaborative access control in web-based social networks. In: Proc. 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security. Volume 5094 of LNCS., Springer (2008) 81–96

[7] M. Egele, A. Moser, C. Kruegel, and E. Kirda. Pox: Protecting users from malicious facebook applications. In Pervasive Computing and Communications Workshops, 2011 IEEE International Conference on, pages 288 –294, march 2011.

[8] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. Commun. ACM, 35:61–70, December 1992.

[9] S. Gurses, R. Rizk, and O. Gunther. Privacy design in online social networks: Learning from privacy breaches and community feedback. In ICIS 2008 Proceedings, New York, USA, 2008. ACM.
[10] M. Majeski, M. Johnson, and S. M. Bellovin. The failure of online social network privacy settings. Technical Report CUCS-010-11, Feb. 2011.