PriMa: An Effective Privacy Protection Mechanism for Social Networks

Anna Squicciarini
College of Information Sciences and Technology
The Pennsylvania State University, USA
acs20@psu.edu

Federica Paci
Department of Information Engineering and Computer Science
University of Trento, Trento,Italy
paci@disi.unitn.it

Smitha Sundareswaran
College of Information Sciences and Technology
The Pennsylvania State University, USA
sus263@psu.edu

ABSTRACT

In this paper, we propose PriMa (Privacy Manager), a privacy protection mechanism which supports semi-automated generation of access rules for users' profile information. *PriMa* access rules are tailored by the users' privacy preferences for their profile data, the sensitivity of the data itself, and the objective risk of disclosing this data to other users. The resulting rules are simple, yet powerful specifications indicating the adequate level of protection for each user, and are dynamically adapted to the ever changing setting of the users' preferences and SN configuration.

1. INTRODUCTION

Web 2.0 revolutionizes how people store and share personal data, allowing for pervasive sharing of personal information on the web. This change has multi-faceted implications, especially on privacy. End users, are often unaware of the size or nature of the audience that could potentially access their data. These issues are particularly pronounced in Social Network sites (SNs from now on), where the false sense of intimacy amongst digital friends often leads to potentially risky disclosures of private data.

Privacy in SNs can be compromised in several ways, by stealing profiles' data, observing the SN graph and by cross-correlating distributed profiles that belongs to multiple sites [19, 7]. One of the main threats to the users' privacy stems from accidental disclosures of data. For example, despite a user may choosing to protect her actual name, another user could accidentally reveal it in a message. Besides, a digital dossier of a user can be built by aggregating partially obfuscated profiles on various SNs.

The privacy protection mechanisms currently provided by most SNs fall short as they enforce access policies set by users. Setting privacy preferences in these policies is a tedious and confusing task for average users having hundreds of connections and extensive profiles [1, 2]. Hence, users often end up with policies which do not protect their personal information well. Further, the anonymization techniques used by some sites to obfuscate users' personal identifying information (PII) may not succeed in protecting it from inappropriate disclosures [19]. Hence, we need privacy protection mechanisms that guarantee SN users protection of their shared data without any tedious policy specification by them. An effective solution for this problem should take into account users' privacy preferences on the desired level protection of their profile information, and adapt the same to the objective risks users face by taking into account the structure of the SN graph and the level of exposure of connected users. In this paper, we propose $PriMa$ (Privacy Manager), a privacy protection mechanism which automatically generates access rules for users' profile information. $PriMa$ access rules are generated on the basis of users' privacy preferences on their profile data, the sensitivity of the data with respect to the privacy settings of the user such as his privacy preferences for his profile data and the degree to which his profile data is at a risk of being exposed to others, and the risk of disclosing such data to other users. These access rules allow users to enforce fine-grained protection, such that the rules can be stated for different levels of granularity ranging from single traits to an entire class of them. Due to this fine-grained control, accidental disclosures are avoided. Hence, $PriMa$ reduces the chance of accidental disclosures due to outdated policies.

The rest of the paper is organized as follows. Section 2 provides a formal representation of SNs and user profiles. Section 3 presents how users' profile data are partitioned based on their sensitivity while Section 4 presents the generation of access rules for the same. Section 5 outlines the related works, with concluding remarks in Section 6.

2. REPRESENTATION OF SOCIAL NETWORKS AND USERS' PROFILES

In this section, we present the concepts that characterize $PriMa$ framework, including the formal definition of SN, profiles, and users' privacy preferences.

2.1 SN Representation

A SN is a labeled graph $\langle U, E, \Phi \rangle$, where U denotes the set of nodes and E the labeled edges. Each node represents a user i, and an edge $E_{i,j}$ represents a relationship between users i and j. Edges are labeled with the social relationship type that connects the two users. The labeling function Φ is defined as $\phi : U \times U \rightarrow \mathcal{R}$, where U is the set of users registered to the SN and $\mathcal{R}$ is the set of the possible relationships connecting the users. We assume the SN supports a finite set of relationships $R = \{R_1, \ldots, R_m\}$, which are explicit and mutually accepted by the involved users. For simplicity we focus on binary user relationships, and denote a relationship as i *:R:* j, being i and j users' unique identifiers, and *R* the

ASIACCS'10 April 13–16, 2010, Beijing, China.

relationship that connects them. The set $deg(i)$ is the set of first degree connected users. The cardinality of $deg(i)$ is denoted as $\#deg(i)$. Users can join groups within the SN, where each group has a unique name, without any approval from other SN members. Each user i has a web space or profile denoted by $prof_i$. A profile, $prof_i$, is a collection of *traits* $[T_1(i), ..., T_w(i)]$ sorted by order of appearance (i.e., the older the traits, the lower the index). Traits can be data posted by the user i or by other users. The user i has control over the access of the traits posted on his/her profile regardless of whether the traits have been posted by i or by the other users.

Each trait is a pair $T_k(i) = (tn_k(i), tv_k(i))$, where tn_k is the trait's name, and tv_k the trait's value. Traits can belong to one of the following categories: T_{attr}, denoting users' attributes, T_{comm}, denoting comments and posts, T_{mm}, representing group membership, and T_{rel}, representing users' relationships. [1] Depending on their semantics, some traits have a single unique value, like first name and last name, whereas others, like address or telephone number could have multiple values. For simplicity we assume traits are normalized [18]. If a trait is of type T_{attr}, the trait name denotes the attribute type and the trait value denotes the value assumed by the attribute. For example, a user John Doe has the trait ("LastName","Doe") of type T_{attr}, where the trait name $tn_k(i)$ is "LastName" and the trait value $tv_k(i)$ is "Doe". Traits of the type T_{mm} model groups' memberships. In this case, the trait name is "Group" while the value is the given group's name. For example, when Jane Doe joins the group "Fashionista", a new trait with the trait name "Group" and the trait value "Fashionista" is added to her profile. Traits of type T_{rel} are represented as tuples where the trait name is equal to "Relationship" and the trait value is $i : R : j$. Traits of type T_{comm} represent streaming data, that is, posts, comments, and other html content that users post over the web. They are modeled by a tuple where the trait name is "Comment" and the trait value is the comment or post text. Comments (or posts) can be associated with other traits. For example, the post "How are you Jane? How is Austin?" left on Jane's profile is associated with the traits of type T_{attr} ("First Name","Jane") and ("Location", "Austin").

2.2 Traits and User's Privacy preferences

When a user i registers to the SN, he specifies a coarse-grained privacy preference for each of the four main trait categories T_{attr}, T_{comm}, T_{mm} and T_{rel}, or some default settings are applied by the SN site. These initial values are used to bootstrap $PriMa$ with finer grained privacy preferences, denoted as $\alpha(T_k(i))$ (α, when the trait is not relevant) for each trait $T_k(i)$ in the profile of user i.

If a trait has no specific α value, $PriMa$ derives the user's privacy preference for that trait, by leveraging the same for other traits that have some commonalities with the given trait, such as the value or the name. Otherwise, if this is not possible due to lack of user input, it calculates the α expected value based on the corresponding α values specified by other users having in their profile a trait $T_k(i)$.

$PriMa$ dynamically evaluates whether a recently updated privacy preference $\alpha(T_k(i))$ can be applied to other traits in the user's profile. To do this, it considers the frequency of updates of the privacy preferences and infers the user's privacy inclination. If such inference is not possible, due to lack of data or user's input, $PriMa$ computes an expected value for $\alpha(T_k(i))$ leveraging the values of $\alpha(T_k(j))$ specified by other users for traits similar to $T_k(i)$. The idea is to group together those users who share the same trait types, are linked by a relationship and have displayed similar privacy preferences. Such a group of users, referred to as *Crowd*, serves as a first indicator of user's i privacy behavior and expectation.

[1]For simplicity, we do not consider images among the trait's types.

DEFINITION 2.1 (CROWD). *Let U_{set} be a subset of users in U. Let i be a user in U. U_{set} is a* Crowd *for a trait $T_k(i) = (tn_k(i), tv_k(i))$ of i'profile and for user i if and only if the following conditions hold:*

- *Given a trait value $tv_k(i)$, $\forall j \in U_{set}$ there exists $T_k(j) = (tn_k(j), tv_k(j))$ in $prof_j$ s.t. $tn_k(i) = tn_k(j)$,*
- *Given a not empty set of relationship types $\{R_1, \ldots, R_m\}$ $\forall\ j \in U_{set}$ there exists a relationship $i : R : j$, where $R \in \{R_1, \ldots, R_m\}$.*
- *for every trait $T_k(i)$ in $prof_i$*

$$\alpha(T_k(i)) \approx \sum_{j=1, i\neq j, j\in U_{set}}^{|U_{set}|} \alpha(T_k(j))p(\alpha(T_k(j))$$

where $p(\alpha(T_k(j))$ is the p.m.f of $\alpha(T_k(i))$; $\alpha(T_k(i))$ is a discrete random variable.

At first, there may not be a specific value for $\alpha(T_k(i))$. Hence, in condition 3, we use the generic preference value specified for the category $T_k(i)$ belongs to.
To compute the expected value for a trait $T_k(i)$, we adopt the Expectation-Maximization (EM). The EM algorithm [13] is used to find the maximum likelihood of various parameters in probabilistic models. The algorithm takes as input the recently updated known values of $\alpha(T_k(j))$, where $j \neq i\ \forall\ j \in Crowd$, where $Crowd$ is a Crowd associated with user i. If no changes have been made for the same trait type in i's Crowd, no meaningful updates can be applied to the sensitivity value, and the value of α is not changed.

3. $PRIMA$ TRAITS CLASSIFICATION AND PARTITION

Once a privacy preference value is associated by $PriMa$ to each trait in a user profile, the traits are then classified and partitioned based on the notion of *sensitivity*. Then, for each partition the rules that determine who have access to which traits are derived.

The "sensitivity score", denoted as $\theta(T_k(i))$ (simply referred to as θ when no ambiguity arises) is a measure of how sensitive a trait $T_k(i)$ in a user's profile is with respect to the user's privacy preferences for his data and the exposure of the user's profile data to others; the higher θ is, the higher is the sensitivity of a particular trait. θ takes into account not only the privacy preference of the user on $T_k(i)$, but also objective information about the SN such as the popularity of the profile and of the trait itself. θ is calculated combining three different metrics: the number of users that are part of user i's Crowd that have trait T_k in their profile accessible by i, denoted as $f_{cr}(T_k(i))$; $L(i)$, the looseness of the profile; and $\alpha(T_k(i))$, the privacy preference of i for the trait $T_k(i)$. The looseness gives a measure of the popularity of a profile $prof_i$, and thus of its potential level of exposure.We omit its detailed formulation due to lack of space.

DEFINITION 3.1. *(Sensitivity Value) Let $T_k(i)$ be a trait in the profile $prof_i$ of a user i, and let cr be a Crowd related to user i with respect to $T_k(i)$. Let $f_{cr}(T_k(i))$ be the value of f with respect to cr, $L(i)$ be the looseness of the profile $prof_i$, and $\alpha(T(i))$ the value of the privacy preference for $T_k(i)$. The Sensitivity score for $T_k(i)$ is calculated as follows:*

$$\theta(T_k(i)) = \frac{1}{f_{cr}(T_k(i))} * L(i) * \alpha(T_k(i))$$

Once a sensitivity value θ is assigned to each trait $T_l(i)$ in a profile $prof_i = [T_1(i), \ldots, T_m(i)]$, we can cluster traits into $\gamma_1(i)$,. . ., $\gamma_k(i)$, $k \geq 2$ classes based on traits level of sensitivity computed according to Definition 3.1. We employ the k-**means clustering for discrete objects** algorithm [16] (discrete k-mean for short) in order to perform the partitioning. Using the k-means algorithm, the system administrator can control the granularity of the partitions and, therefore, the granularity of the privacy policies generated by varying the value of k. Once the clustering is finished, the overall sensitivity of each partition, denoted by $\theta(\gamma(i))$, is calculated as the mean of the sensitivity scores of all the traits in that cluster.

4. $PRIMA$ PRIVACY POLICY GENERATION

One of the key features of $PriMa$ is that the policy protecting the traits in users' profiles is automatically suggested by the framework based on users privacy inclinations and the actual risk of exposing the traits to a certain set of users. Such a policy is specified by means of a set of access rules that essentially state who is granted access to the classes of traits generated for a user profile (positive rules) and who is denied access (negative rules). A $PriMa$ access rule is represented as a tuple ($AccRule$) of the form $\langle Pred, Users, \gamma \rangle$ where $Pred$ is a predicate that can assume the values $Share$, and $NotShare$ which denote positive and negative rules respectively. $Users$ is the set of users to which the policy is applied. $Users$ can be a set of users identifiers or a set of relationships, and γ is the partition of traits protected by the rule. At implementation level, only positive access rules are needed because negative rules are complementary to the positive ones. Without loss of generality, we show how the rules are computed for users who have a first-degree relationship with the profile owner and we assume that the other users, who are not related to the profile owner by a first-degree relationship are denied access. It is straightforward to apply the same mechanism to connections of higher degree. Access rules are generated for each class of traits $\gamma_n(i)$ in which a profile $prof_i$ has been partitioned, based on the notion of *user access score*.

The *user access score*, denoted as $\delta(\gamma_n(i), j)$, is representative of the adequacy of a given user j to access a given partition $\gamma_n(i), n \leq k$, (where k is the number of classes of traits in which user i profile has been partitioned). We use two metrics to compute the user access score, the *relationship score* and the *risk*. The relationship score rates the strength of the relationship between user i and j; the higher their relationship score, the stronger the relationship. The second dimension estimates the risk of disclosing a class of traits $\gamma_n(i)$ to the user j. Even if a user is trusted, it may be objectively unsafe to disclose certain traits to him. We compute a score that is an indicator of the strength of the relationship between two users, which does not solely depend on the users' input. We calculate the relationship score of two users i and j as

$$rel(i,j) = type * \frac{\#(deg(j) \cap deg(i))}{\#deg(i)} * rep_j$$

where $type$ is a normalized numerical value assigned to each relationship type, such that the closer the relationship, the higher the score. Here, we refer to SNs that have a hierarchical structure for first degree relationships. For example for the relationships "Best Friends", $type = 1$, for "Friends", $type = 0.8$, $FriendsofFriends = 0.6$ and so on. If two users share more than one relationship, then the closest one is taken for assigning the value for $type$. If only one relationship type exists, or if the supported relationships are not hierarchically sorted, the value of $type$ is set to 1 by default.

The parameter rep_j denotes the normalized rating given by user i to j. This value can be directly input by the user, or calculated using approaches such as [6]. The ratio $\frac{\#(deg(j) \cap deg(i))}{\#deg(i)}$ expresses the similarity in terms of common first-degree connections. To estimate the access score, we also consider the specific risk level associated with the disclosure of a trait set $\gamma_n(i)$ to a user j. The risk score of disclosing a class of traits $\gamma_n(i)$ of user i to a user j is calculated as follows:

$$risk(\gamma_n(i), j) = L(j) * \theta(\gamma_n(i)) \quad (1)$$

In the equation, we use $L(j)$, the looseness of user j, to express the information leakage associated with j, and combine it with actual sensitivity of the partition under consideration.

We are now ready to define the *user access score*, which is calculated as the ratio between the relationship and the risk score.

$$\delta(\gamma_n(i), j) = \frac{rel(i,j)}{risk(\gamma_n(i), j)} \quad (2)$$

The user access score is directly related to $rel(i,j)$ because the relationship score indicates the strength of the relationship between two users. On the other hand, the higher the risk of a user with respect to $\gamma_n(i)$, the lower should be the access score of the user.

Once the adequacy scores are computed, generating access rules in $PriMa$ is straightforward. For each class $\gamma_n(i)$, $1 \leq n \leq k$ the set of users in $deg(i)$ is partitioned into two smaller sets, $Users$ and $Users''$. $Users$ is the set of users allowed to access $\gamma_n(i)$ while $Users''$ is the set of users who are denied access to $\gamma_n(i)$. To build sets $Users$ and $Users''$ for a class $\gamma_n(i)$, for each and every user, j, in $deg(i)$, the algorithm checks whether the user has enough privilege to access the partition under consideration by verifying that the *user access score* $\delta(\gamma_n(i), j)$ is greater than the threshold ξ. If this condition is met by the user j, he is added to the set $Users$. Otherwise, he is added to the set $Users''$. Following this approach, the positive and negative access control rules for $\gamma_n(i)$ are generated using the sets $Users$ and $Users''$. The algorithm iterates this process over all the partitions in order to cover all the traits in the user's profile. Such approach helps define the access rules and the policy at a very fine granularity, i.e. on a per user, per partition basis. The algorithm is efficient, with a complexity linear to the number of users connected with the profile owner.

5. RELATED WORK

SNs demand a new approach to access control [8, 4, 9], flexible and based on interpersonal relationships. A first attempt along this direction has been taken by authors in [9], where a social-networking based access control scheme suitable for online sharing is presented. In the proposed approach, authors consider identities as key pairs, and social relationship on the basis of social attestations.In [8], authors proposed a content-based access control model, which makes use of relationship information available in SN for denoting authorized subjects More sophisticated mechanisms have been proposed in [4, 3]. Carminati and colleagues presented a rule-based access control mechanism for SN. Such an approach is based on enforcement of sophisticated policies expressed as constraints on the type, depth, and trust level of existing relationships. Subsequently, the same group of authors has extended the previously proposed model[3] to make access control decisions completely decentralized and collaborative. This work is orthogonal to ours, since PriMa does not only deal with privacy of users' relationships, but also to fine-grained data protection. In an approach parallel to ours, Lindamood et al. [11] have leveraged the data available on SN including relationship and the effect that changing a user's trait has on their privacy. Lindamood et ,al. employ a Naive Bayes classifier to classify the data gathered from SN.The

authors explore the effect of sanitizing both traits and link details. We borrow the idea of representing users' profile in terms of traits from Lindamood's work. However, this work focuses on the effect of trait and link sanitization on private information leakage rather than leveraging the data available on the SN to provide an access control mechanism geared towards preventing the leakage of private information.

Regarding the evaluation of users' relationships, a number of researchers have investigated trust metrics in SN[6, 20, 4]. The trust metrics are a means to predict the trustworthiness of a user - often unknown to the focus user. Our work uses a metric, the user access score, that elegantly leverages the user's perceived local trust -expressed by the reputation value- with supervised SN centric dimensions, such as the objective risk associated with certain users.

Finally, Ziegler et al. [20] argue that there is strong evidence for correlation between user similarity and trust. Our learning approach for predicting users inadequacy on specific data sets leverages this idea of similar user profile attributes being an indicator of their adequacy to access the data. In addition, we go one step further by leveraging the focus user' social graph network metrics as input to predicting the appropriate protection of user's data.

6. CONCLUSION

This paper explores an adaptive policy generation framework, $PriMa$, as a first step towards providing flexible, adaptive and powerful access control to SN users. $PriMa$ access rules are generated on the basis of users privacy preferences on their profile data, the sensitivity of the data and the risk of disclosing such data to other users.

There still exist many shortcomings to be overcome before $PriMa$ can be regarded to be completely sufficient in protecting the user's information. For example, the tuning of the thresholds used for the rule generation processes will play an essential role in real-world settings.

7. REFERENCES

[1] A. Acquisti. Privacy in electronic commerce and the economics of immediate gratification. In A. Press, editor, *Proceedings of the 5th ACM Electronic Commerce Conference*, pages 21–29, 2004.

[2] A. Acquisti and J. Grossklags. Privacy and rationality in decision making. *IEEE Security and Privacy (January/February)*, pages 26–33, 2005.

[3] B. Carminati and E. Ferrari. Privacy-aware collaborative access control in web-based social networks. In *DBSec*, pages 81–96, 2008.

[4] B. Carminati, E. Ferrari, and A. Perego. Private relationships in social networks. In *ICDE Workshops*, pages 163–171, 2007.

[5] Facebook. http://www.facebook.com.

[6] J. A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, College Park, MD, USA, 2005. Chair-Hendler, James.

[7] R. Gross, A. Acquisti, and H. J. Heinz, III. Information revelation and privacy in online social networks. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, New York, NY, USA, 2005. ACM.

[8] M. Hart, R. Johnson. A. Stent. More content - Less control: Access control in the Web 2.0. In *IEEE Web 2.0 Privacy and Security Workshop*, 2007.

[9] K. Kollu, S. Saroiu, and A. Wolman. A social networking-based access control scheme for personal content. In *21st ACM Symposium on Operating Systems Principles. Work in Progress*, October 2007.

[10] S. R. Kruk, A. Gzella, and S. Grzonkowsk. D-FOAF distributed identity management based on social networks. In *First Asian Semantic Web Conference*, pages 140–154, 2006.

[11] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring private information using social network data. In *18th International World Wide Web Conference (WWW2009)*, 2009, ACM.

[12] P. Massa and P. Avesani. Controversial users demand local trust metrics: an experimental study on epinions.com community. In *25th American Association for Artificial Intelligence Conference (AAAI)*, 2005.

[13] G. McLachlan and T. Krishnan. The EM algorithm and extensions. Wiley series in probability and statistics.

[14] B. Monahan. Gnosis: HP labs modeling and simulation framework. Systems Security Lab, 2009.

[15] M. E. J. Newman. The mathematics of networks. In *Blume, L.E., Durlauf, S.N. (eds.), The New Palgrave Encyclopedia of Economics, 2nd edn.* Palgrave Macmillan, Basingstoke, 2008.

[16] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, 2000.

[17] P. Resnick and R. Zeckhauser. The value of reputation on eBay: A controlled experiment. 9(2):79Ð101, 2006.

[18] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56, Kent, UK, 1981. Butterworth & Co.

[19] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 531–540, 2009. ACM.

[20] C.-N. Ziegler and J. Golbeck. Investigating interactions of trust and interest similarity. *Decis. Support Syst.*, 43(2):460–475, 2007.