3LP: Three Layers of Protection for Individual Privacy in Facebook

Khondker Jahid Reza, Md Zahidul Islam, Vladimir Estivill-Castro

3LP: Three Layers of Protection for Individual Privacy in Facebook

Khondker Jahid Reza*[1], Md Zahidul Islam[1], and Vladimir Estivill-Castro[2]

[1] School of Computing and Mathematics, Charles Sturt University,
Panorama Avenue, Bathurst 2795, NSW, Australia.
[2] DTIC Universitat Pompeu Fabra
Roc Boronat, 138, Barcelona 08018 Spain.
Email:{kreza,zislam}@csu.edu.au vladimir.estivill@upf.edu

Abstract. The possibility that an unauthorised agent is able to infer a users' hidden information (an attribute's value) is known as attribute inference risk. It is one of the privacy issues for Facebook users in recent times. An existing technique [1] provides privacy by suppressing users attribute values from their profile. However, suppression of an attribute sometimes is not enough to secure a users' confidential information. In this paper, we experimentally demonstrate that (after taking necessary steps on attribute values) a user's sensitive information can still be inferred through his/her friendship information. We evaluated our approach experimentally on two datasets. We propose 3LP, a new three layers protection technique, to provide privacy protection to users of on-line social networks.

Keywords: Privacy enhancing technologies, Attribute inference.

1 Introduction

Humans naturally keep themselves connected with friends, colleagues and families but due to geographical distances, people may not be able to meet their friends regularly. Hence, online social networks (OSNs) play a vital role to connect and share contents among people. Now, all over the world, citizens and organisations make extensive use of OSNs such as Facebook, Twitter, LinkedIn, and Google+. In recent years the usage of OSNs, particularly the usage of Facebook, has increased extensively [2, 3].

Facebook is currently the third (after Google and Youtube) most viewed website [3] with 1.09 billion average active users every day [2]. Users typically store and share various personal data on Facebook resulting in the possibility of privacy breaches [4]. Privacy is a crucial element of society. Social scientists have provided several definitions. Tavani defines privacy as our ability to restrict access to our personal information and to have control over the transfer of our information [5]. Rachel [6] argues that privacy is the individuals' ability to disclose selectively personal information related to themselves. What is private for

* Corresponding author.

one may not be private for some others. For example, some may consider their political affiliation to be private while some others may not consider important to disclose their political alignment.

Data stored on Facebook about other users can be analysed for link prediction and attribute value prediction to learn sensitive and private information of victim users and hence compromise their privacy [7, 8]. Sophisticated data mining techniques can breach individual privacy [9] on Facebook.

It was empirically demonstrated [9] that a data set built from other users' data that do reveal what one user U considers confidential can be used by an attacker M to build a classifier that predicts U's private information with high confidence. The fundamental idea of the first techniques to guard against the attribute inference attack (*NOYB* [10], *TOTAL_COUNT*, and *CUM_SENSITIVITY* [1]) is to identify a user's publicly available attribute values which are high predictors of a sensitive attribute value and recommend to the user to obfuscate the predictors. While *NOYB* [10] randomly selects visible attribute values to obfuscate, *TOTAL_COUNT* and *CUM_SENSITIVITY* [1] heuristically identify which public data is highly informative and very likely to be influential in any classifier built by data mining techniques; therefore, recommending to the victim to modify or suppress the visible attribute values those are high predictors. The difference between *TOTAL_COUNT* and *CUM_SENSITIVITY* is in the ranking of the predictors, but both of them are very similar, so we encapsulate them into the global name of *PrivAdv* for short.

The protection technique *PrivAdv* does not consider friendship links among the users as information that M can use to infer the sensitive value of U. The information from on-line social networks can often be organised as a social attribute network (SAN) [11]. The SAN model integrates both users' attribute information and their friendship network. Although *PrivAdv* has been extended to evaluate risks of the inference attack that derive from connections in the social network [12], the easiness of such an attack was not illustrated. Moreover, no concrete suggestions of what shall users do when their privacy is at risks because of social connections. That is, in such extensions [12], the algorithms recommend to unfriend or befriend a user from the victim's friend list randomly if such friend discloses any information which is sensitive to the victim. In those methods, the number of added or deleted friends may be large, and the victim may not be interested in this frequent addition and deletion of friends. We experimentally show that friendship links can be a useful piece of information for M. We also show that naively extending the existing technique [1] may not be effective to ensure privacy protection against M usage of this information. Here, we also propose a new technique (which we name *3LP*) with three layers of protection in order to protect the sensitive value of U even if M uses the friendship links. We also experimentally demonstrate the effectiveness of *3LP*.

This paper is organised as follows. Section 2 discusses some limitations of existing techniques as evidenced by our initial experiments. Section 3 presents *3LP*. Finally, Section 5 gives concluding remarks.

2 The Importance of Friendship Links

We now argue that any protection technique that does not take into consideration both, the attribute values of a user and links of a social network, is not able to ensure sufficient protection. We justify our argument since a real-life attacker can try to infer the sensitive information using whichever of the two aspects (attribute values and links) is ignored by the protection technique, dodging the single focused privacy mechanism. For example, we demonstrate that a previous work [1] that does not take the link information into consideration may not be able to secure sensitive data of users when an attacker uses the connection information of a social network.

We assume that attackers have access to a large data set which has the structure of an undirected social network (or graph) G having a N number of users, each with A attributes. Here, without loss of generality, each attribute value is considered as a distinct binary attribute. This standard data representation converts a categorical attribute like *hometown* (with possible values *Sydney*, *Melbourne*, and *Brisbane*) into a characteristic vector: the value is true if and only if the user's residential city correspond to that attribute-value pair. Under the SAN model, not only members of the OSN are vertices, but attribute-values are also vertices. For each user-vertex u with an attribute-value pair $a = v$, the SAN places an edge between u and the attribute-value pair $a = v$.

The SAN model also places an edge between two users if they are friends.

The SAN data model can be used by attackers to estimate the influence of a user on another user. The idea here is that linked users who have a small number of friends are strongly connected and have a high influence on each other. For example, if a user Tom White is linked to Rob Black and each of them has only two other friends then Tom and Rob have a high influence on each other meaning that if Rob supports the Labour party, then there is a greater chance that Tom will also support the Labour party. On the other hand, if a user is linked to another user who has a huge number of friends then the two users are relatively weakly connected and have low influence on each other. For example, if Tom is linked to Mel Gibson who has thousands of friends, then the fact that Tom supports the Liberal party does not give a strong clue on whether or not Mel Gibson also supports the Liberal party. Such influence of a user on another user u can be computed through a metric that represents the strength of the connection between u and an attribute-value pair $a = v$, where the strength of a connection is proportionate to the number of common users (who are friends of u and have the attribute-value pair $a = v$) and inversely proportionate to the numbers of friends of the common users.

We first need to introduce some notations before we formally present the metric function for a user and an attribute-value pair. We denote by $\Gamma_{s+}(u)$ the set of all social users linked to a user u. Similarly, $\Gamma_{s+}(a = v)$ is the set of all users having the attribute-value $a = v$. Also, $\Gamma_{a+}(u)$ is the set of all attribute-value pairs linked to user u. Thus, the neighbourhood of u in the SAN is, $\Gamma_{+}(u) = \Gamma_{s+}(u) \cup \Gamma_{a+}(u)$. On the other hand, $w(u)$ is the weight of any social

Table 1. Attributes of Facebook data set D_{FB}.

Attribute name	Attribute values
Gender	female; all 616 users are female aged over 18 years
Profile image	contains 12 categories represented by 1 to 12, as follows, image shows the user: 1-alone, 2-with one or more friends, 3-at a special occasion, 4-with their special partner, 5-smiling, 6-in a unique location not in their hometown, 7-shows only face/head, 8-playing or watching sport, 9-with family, 10-depicts an object with apparent meaning to the user, 11-having a unique visual effect, 12-Reveals too much skin
Relationship status	contains 11 categories: 1 to 10 and null, as follows. 1-single, 2-in a relationship, 3-engaged, 4-married, 5-it's complicated, 6-in an open relationship, 7-widowed, 8-separated, 9-divorced, 10-in a civil union, null
Interested in	null, men,women,both
Family on FB	absent, present
Hometown	absent, present
Show sex	no, yes
High school	absent, present
Year-graduated from high school	absent, present
University or college	absent, present
Year graduated from university or college	absent, present
Timeline	absent, present
Work	absent, present
Friend	high, medium, low, null
Album	high, medium, low, null
Photo	high, medium, low, null
Language	english, english+, other, other+, null
Religion	absent, present
Activities	absent, present
Email	absent, present
Date of birth (DOB)	1-full dob is revealed, 2- only day and month are revealed, 3- dob is not revealed
Political view	absent, present
People who inspire	absent, present
Class attribute	connected, lonely

node (i.e. a user) $u \in G$. In this study, we assume the weight of each social node is constant and is set to 1. The equation [13] for the metric $m(u, a = v)$ is

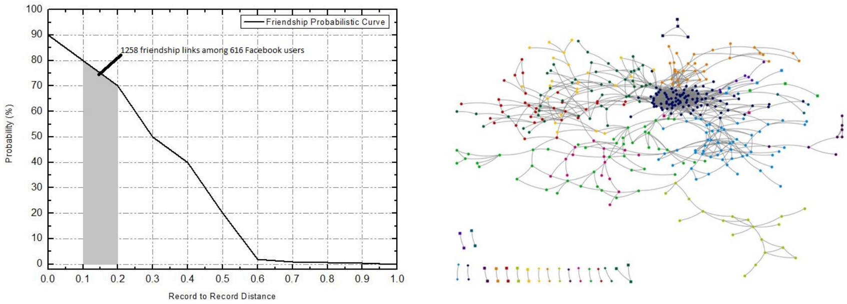
$$m(u, a = v) = \sum_{t \in \Gamma_{s+}(u) \cap \Gamma_{s+}(a=v)} \frac{w(t)}{log|\Gamma_{+}(t)|}. \tag{1}$$

An interesting property of this metric is that, if friendship information is available, then $m(u, a = v)$ can be calculated for any attribute-value pair $a = v$ whether the user u has that value or not. A high $m(u, a = v)$ suggests that u has a high chance of having value v for attribute a since, u is connected to many other users who have $a = v$. Since $m(u, a = v)$ is computed by taking the network link information into account, we will add $m(u, a = v)$ information for each user and each attribute in a data set (having a number of users and a number of attributes for the users) [12] to demonstrate that an existing technique [1] (that does not take the network link information into account) may not provide protection against an attack using the link information.

2.1 Data sets

We use the same data set D_{FB} that was used in some previous studies [1, 9]. The data set D_{FB} has 616 records where each record contains information of a female Facebook user who is either feeling *lonely* or *connected* as it is explicitly mentioned in their recent posts. Out of 616 records, 308 users are *lonely*, and 308 users are *connected*. As in the previous studies [1, 9], we also assume that the *emotional status* is confidential. A malicious data miner will try to learn this information of a user who has not revealed this information. Hence, *emotional status* is the class attribute while building a classifier to learn the patterns for discovering the *emotional status* of members of the social network.

Thus, the structure of the data set D_{FB} consists of 23 non-class attributes and the class attribute *emotional status*. Table 1 provides details of these attributes.



(a) Friendship probability graph (b) Friendship Network of 616 users

Fig. 1. The friendship links simulation

For example, the *Profile Image* attribute contains 12 categories based on the image. If the image shows the user alone, then the value of the attribute is 1, if the image shows the user with one or more family members, then the value of the attribute is 2 and so on. The attribute *Hometown* contains two values *absent* and *present*. If the hometown of a user is revealed, then the value of the attribute is *present*; otherwise, *absent*. The attribute *Friend* has four possible values: *high*, *medium*, *low* and *null* depending on the user's number of friends. If the friendship information is not available, then the attribute has *null*.

However, D_{FB} does not have any information relating the social network links (i.e. friendship information). Therefore, we first simulate the connections among users to construct a data set D'_{FB} that contains information relating social network links. We set the probability of a link between two users inversely proportional to the Hamming distance between the two users. We set the record-to-record distance (or *R2RD* $\in [0, 1]$) between two users as the Hamming distance divided by 23 (the number of non-class attributes).

Users having similar attribute values (i.e. low Hamming distance) are likely to have common interests and thus are likely to have friendship links (social links) between them [14]. A link between two users will be a Bernoulli trial with probability p where we set p as a high probability of a friendship link when the *R2RD* is low. In particular, when the value of *R2RD* between two users is within the range of 0.0 and 0.2, then we set the link probability p linearly between 0.9 and 0.7. When *R2RD* is between 0.2 and 0.3, then the link probability p is linear between 0.7 and 0.5. Thus, for example, if the *R2RD* between two users is 0.3, then we draw a link between them with probability 1/2; that is, it is equally likely there is no connection. Fig.1(a) provides the plot that determines the link probability p as a function of *R2RD*. In this model, even if the *R2RD* is large, between 0.6 and 1, there is still some probability that the users are linked as friends. Fig.1(a) shows that 1258 friendship links were created among users whose *R2RD* is between 0.1 and 0.2. Fig. 1(b) shows a social network drawn in the way, where the dots represent the users and the links represent the friendship between the users.

Once the friendship links are simulated we can compute the $m(u, a = v)$ for every user u and attribute-value pair $a = v$. Recall that the data set D_{FB} has 23 non-class attributes and a class attribute. A user u is represented by a record $r \in D_{FB}$ that has 24 attribute values. For each attribute value of u we compute $m(u, a = v)$. Thus, for each attribute-value pair, we create a new attribute containing $m(u, a = v)$ for each user u. Let us call these newly created attributes "link attributes" and the original 23 attributes "regular attributes". Therefore, when we consider the link information, the expanded data set D'_{FB} has now altogether 24+24 = 48 attributes. That is, in the expanded data set D'_{FB}, we have 47 non-class attributes and a class attribute containing two possible values: *lonely* and *connected*.

We also utilize a synthetic data set as per those synthetic OSN data sets [15]. This data set consists of 11 non-class attributes which are given in Table 2. The data contains 1000 records (489 male users and 511 female users) and 50,397 friendship links. These are also synthetically generated friendship links [15]. We shall consider two version of this data set. In the first, we take *political orientation* as the confidential attribute of the data set and it is denoted by $D_{Political}$. In the second one, now D_{Sexor} we consider *sexual orientation* as the confidential attribute. Both of this will have 10 non-class attributes (but they exchange *sexual orientation* and *political orientation* as the class attribute).

After preparing $D_{Political}$ and D_{Sexor}, we calculate SAN metric values for each attribute as we did for D'_{FB}. This results in expanded data sets $D'_{Political}$ and D'_{Sexor} respectively with 11+11=22 attributes one of which is the confidential class attribute.

2.2 Empirical Demonstration

We now empirically demonstrate the impact of considering social links on individual's privacy. For a data set D, in our experiments, we split the users in 10 disjoint groups: $\{D^1, D^2, D^3, \ldots, D^{10}\}$. For example, for D_{FB} $|D^i_{FB}| = 61$ for $i = 1, \ldots, 9$ and $|D^{10}_{FB}| = 67$. For the i-th iteration the users in D^i are considered those users who wish to keep their confidential attribute unpredictable from the

Table 2. Attributes of Synthetic Dataset.

Attribute name	Attribute values
Age	contains 7 categories: 18-25, 26-35, 36-45, 46-55, 56-65, 66-75, 76-85
Gender	male, female
Residence	contains 5 categories: palo alto, santa barbara, san jose, boston, winthrop
Religion	contains 7 categories: christian, hindu, jewish, muslim, sikh, other religions, no religious affiliation
Marital status	contains 4 categories: single, married, divorced, widowed
Profession	contains 7 categories: image shows the user: manager, professional, service, sales and office, student, natural resources construction and maintenance, production transportation and material moving
Political orientation	contains 7 categories: absent sexual information, bisexual, heterosexual, homosexual
Political orientation	contains 7 categories: far left, left, centre left,centre, centre right, right, far right
Like 1	contains 5 categories: entertainment, music artist, drink brand, soccer club, tv show
Like 2	contains 5 categories: entertainment, music artist, drink brand, soccer club, tv show
Like 3	contains 5 categories: entertainment, music artist, drink brand, soccer club, tv show
Class attribute	political orientation or sexual orientation (any one at a time)

adversary M, while the adversary has the data of the other users $\cup_{j=1}^{10} D^j \setminus D^i$ who have revealed such confidential attribute.

For each user U in D^i, we use *PrivAdv* repeatedly to identify the sensitive rules R^u. In each iteration, the primary attribute obtained from R^u is suppressed until $R^u = \emptyset$. At this stage, *PrivAdv* considers U's privacy protected. Different users in D^i have different attribute-value pairs suppressed.

How, we complement the columns of $\cup_{j=1}^{10} D^j \setminus D^i$ and D^i with the link information, essentially considering D' instead of D. We impersonate the adversary M who builds a forest from $\cup_{j=1}^{10} D'^j \setminus D'^i$. That is, we assume the adversary uses the *SAN* metric and thus obtains a new set of sensitive rules R'^u for each user U in D^i (the users in D^i and D'^i are the same, D'^i has the *SAN* link information as the metric $m(u, a = v)$ as per Equation (1)).

The assumed strategy of the adversary for each D'^i is a decision-tree forest *SysFor* [16] with the aim of building a forest of 10 trees. Throughout the experiments, we use the standard set of parameters of *SysFor*. *SysFor* sometimes cannot build 10 trees as requested due to various reasons such as not having enough good attributes. Nevertheless, *SysFor* always builds at least 8 trees and 40 rules for D'_{FB} data set (refer to Table 3). The sensitive rules (SR) obtained by the adversary's strategy are of 3 types, SRR tests only regular attributes, SRRL tests both link attributes and regular attributes, SRL are sensitive rules made of only the link attributes.

Table 3 contrasts the types of sensitive rules that are obtained from the link attributes from D'_{FB} versus those that do not. Those users in D^i_{FB} who have at least one sensitive rule $\in R'^u_{FB}$ for which no regular attribute value is suppressed by *PrivAdv* are at risk, and we found that the adversary always found at least 20 of these rules. That is, there are plenty of sensitive rules for which all values tested in the antecedent are *link attributes* (i.e. the attributes that contain $m(u, a = v)$ values). Note again that these values are not suppressed

Table 3. Analysis of sensitive rules with and without social link information

Run	Trees	Rules	SR	SRR	SRRL	SRL
1	9	40	31	1	7	23
2	8	71	41	1	8	32
3	8	50	33	0	9	24
4	8	69	42	3	6	23
5	8	83	43	3	20	20
6	8	81	44	3	13	28
7	9	60	41	3	8	30
8	8	49	35	1	13	21
9	8	61	42	3	13	26
10	8	48	37	0	5	32
Average	**8.2**	**61.2**	**38.9**	**1.8**	**10.2**	**25.9**

by *PrivAdv* since *PrivAdv* only uses *regular attributes* from D_{FB} [1]. Users are not properly secured by *PrivAdv* with respect to the social link information. For instance, consider D'^{1}_{FB}, any records satisfying any of the 23 SRLs for D^{1}_{FB} are not secured by *PrivAdv*. We can see from Table 3 that on an average there are 25.9 SRLs out of 38.9 SRs. This indicates that most of the sensitive rules obtained from D'_{FB} are not taken care of by *PrivAdv*. This should not be surprising, the vast majority of information derived in recommender systems and on-line social networks where information is represented as graph models like the SAN derives from the link information.

On the other hand, *SysFor* generates 9 trees in each component D_i for both $D'_{Political}$ and D'_{Sexor} data sets. The average number of SRRL is comparatively higher than SRL and SRR in each component of $D'_{Political}$. Out of 50 SR (i.e., sensitive rules), on average, the number of SRRL is 41.6 where SRL (i.e., sensitive rules with link attributes only) is approximately 0 in our experiments. In D'_{Sexor} data set, on average, 180 SR are generated in each part D_i and among these 176.4 are SRRL. Only 3.7 (on average) sensitive rules are SRL (i.e., containing both regular and link attributes).

The limitation of *PrivAdv* is further defined by the confidential attribute-value pair is revealed by rules in SRL or SRRL. If a user in D'^{i} has a sensitive rule in SRL or SRRL (*PrivAdv* does not suppress any of the attributes in the antecedent of the rule), then the user's information is considered to be insecure, otherwise the user's information is considered to be secure.

In our experiments, we found that among 62 users in each cross fold of D'_{FB} data set, 35 of them (56.62%) have protected information. However, 27 (43.38%) out of 62 users having insecure information. In case of $D'_{Political}$ and D'_{Sexor} data sets, out of all 10 parts D^{i}, on an average 41.7% and 70.5% users, respectively, are having insecure information after *PrivAdv* has been applied. For these insecure users, the attributes suppressed by *PrivAdv* are insufficient to protect their privacy when an adversary uses a data set with link attributes.

3 Our Technique

Our technique *3LP* secures the confidential attribute-value pairs of users even when link attributes (obtained from social links) are taken into consideration. Our technique suggests three layers of protection: Layer 1 suggests to suppress necessary attribute values (and is equivalent to *PrivAdv*: Step 1 and 2 in Algorithm 1), Layer 2 suggests to hide some friendship information and Layer 3 suggests to add new friends.

Step 1 *Compute Sensitivity of Each Attribute for a User.* In Step 1, we invoke the function *GetSensitiveRules()* to create the set of sensitive rules R^s. The set R^s is generic, but the function *GetSensRulesForUser()* uses the attribute values of a particular user U and returns the set R^u of sensitive rules for U. The set $A_u^{r/s}$ of sensitive attributes is the union of all regular attributes in the antecedents of the rules in R^u. The *TOTAL_COUNT* [1] counts how many times each regular attribute A_i appears in the antecedents of set R^u.

Step 2 *Suppress Attribute Values as Necessary (Layer 1). 3LP* identifies the regular attribute A_n with the highest number of appearances in the set R^u and suggests user U shall suppress the value of attribute A_n. As in *TOTAL_COUNT* [1], our first layer only suggests the suppression and leaves the decision up to the user. Either way, the attribute A_n is removed from the set A_u^s of sensitive attributes. If user U suppresses attribute A_n, then all sensitive rules in R^u that have A_n in their antecedent are no longer applicable. In this case, those sensitive rules are no longer in R^u. The treatment is repeated with the next regular attribute with the highest number of appearances in the set R^u until R^u is empty (in which case the algorithm terminates) or the set $A_u^{r/s}$ of regular attributes in R^u is empty (in which case the algorithm continues with **Step** 3. We remark here that in the experiments of this study we assume that a user follows all the suggestions.

Step 3 *Hide Friendship Links as Necessary (Layer 2).* If there are still some sensitive rules $R_j^u \in R^u$, such rules must use only link attributes. We explore if there is any link attribute $m(u, A_n = v)$ whose value can be reduced by deleting or hiding some friendship links in order to reduce the number of sensitive rules in R^u. Unlike the regular attributes, the link attributes cannot be suppressed easily. Moreover, as discussed when Equation (1) was introduced, in many cases $m(u, A_n = v)$ derives from the social links of the user and not the explicit links the user has control.

However, we can offer to the user to carefully change the social links (by deleting/hiding some friendships) and thus alter the values of the link attributes $m(u, A_n = v)$. For example, if we hide the friendship link of the user U with a friend who also shares the same attribute-value pair $A_n = v$, then we can decrease the link attribute value $m(u, A_n = v)$. Moreover, we can see from Equation (1) that if we hide the friendship link of the friend t who has the smallest $\Gamma_+(t) = \Gamma_{s+}(t) \cup \Gamma_{a+}(t)$, then we can maximise the reduction of $m(u, A_n = v)$.

In Step 3, we first find the most sensitive link attribute $m(u, A_n = v)$ for the user U. We then check if the value of $m(u, A_n = v)$ is higher than the split point in a sensitive rule R^u_j, where one of the tests in the antecedent of R^u_j is $A_n \geq \mathit{split\_point}$. If it is, then we suggest user U shall hide the friendship link with a friend who has the smallest $\mathit{\Gamma}_+(t) = \mathit{\Gamma}_{s+}(t) \cup \mathit{\Gamma}_{a+}(t)$ in order to reduce the $m(u, A_n = v)$ value the most. If the user accepts the recommendation, we recompute $m(u, A_n = v)$.

The goal here is to reduce the value of $m(u, A_n = v)$ below the split point so rule R^u_j is no longer applicable to U. We continue the process of hiding friends until we get the a value of $m(u, A_n = v)$ lower than the split point in R^u_j. We then remove R^u_j and any other rules no longer applicable to user U from R^u and repeat the process for another sensitive rule R^u_j that tests $m(u, A_n = v) \geq \mathit{some\ split\_point}$ in it antecedent. At the end of Step 3, if we still have some rules $R^u_j \in R^u$ then we move to Step 4 (Layer 3).

Step 4 *Add New Friends as Necessary (Layer 3).* We again find the most sensitive link attribute $m(u, A_n = v)$ for the user. We check if there is any sensitive rule $R^u_j \in R^u$ that has an antecedent of the from $m(u, A_n = v) \leq \mathit{some\ split\_point}$. If there is such R^u_j, then we aim to add friends and thus increase the value of $m(u, A_n = v)$ so that it eventually becomes greater than the split point and thus R^u_j is no loner applicable to U. Our algorithm *3LP* suggests the adding approach to the user U and the user shall make the decision whether to add the friend or not. Our *3LP* retrieves the possible friend t with the smallest $\mathit{\Gamma}_+(t) = \mathit{\Gamma}_{s+}(t) \cup \mathit{\Gamma}_{a+}(t)$, and recommends to add a friendship link to t. This maximises the increase of the value of $m(u, A_n = v)$ and minimises the number of friendship links to be added.

Table 4. Number of insecure users after applying *3LP* on expanded dataset D'_{FB}.

Run	Number of users in test data set	Number of insecure users			
		After using *PrivAdv*	After using Layer 1 of *3LP*	After using Layer 1 and Layer 2 of *3LP*	After using Layer 1, Layer 2 and Layer 3 of *3LP*
1	61	18	18	15	0
2	61	40	40	36	0
3	61	35	35	19	0
4	61	9	9	9	0
5	61	35	35	33	0
6	61	29	29	27	0
7	61	20	20	17	0
8	61	10	10	10	0
9	61	34	34	22	0
10	67	38	38	28	0
Average	**61.6**	**27**	**27**	**22**	**0**

4 Experimental Results and Discussion

We now present experimental results that validate our algorithm *3LP*. We apply *3LP* on the expanded data sets named D'_{FB}, $D'_{Political}$ and D'_{Sexor} separately. We again partition the data sets into 10 disjoint parts, using one part as the potential victims and 90% of the dataset as the data available for inferring confidential attributes. Table 4 shows experimental results for D'_{FB}.

Algorithm 1 3LP()

```
Input      : User $U$, attribute $C$ that $U$ considers confidential is the class attribute, dataset $D$
             having $N$ records, $A$ is the set of non-class attributes where $A^r \subset A$ is the set
             of regular attributes and $A^l \subset A$ is the set of link attributes, $C$ denotes the class
             attribute $C$ and $G$ the graph information.
Output     : Recommendations for $U$ to act on some attributes in $A$.
Variables  : $A_n$=the $n^{th}$ attribute
             $R^s$= set of sensitive rules
Step 1: Compute Sensitivity of Each Attribute for a User
    $R^s \leftarrow GetSensitiveRules(D, C)$
    $R^u \leftarrow GetSensRulesForUser(R^s, U)$
    $Counter_i \leftarrow 0; \forall Counter_i \in Counter$ /*$Counter_i$ shall total the number of appearances of
    $A_i \in A^r$ in the set of sensitive rules*/;
    $A_u^{r/s} \leftarrow \phi$ /*Initially $A^s$ is set to null*/;
    foreach $R_j^u \in R^u$ do
        foreach $attribute\ A_n \in A^r$ do
            if $A_n$ is in the antecedent of $R_j^u$ then
                $Counter_n \leftarrow Counter_n + 1$
                $A_u^{r/s} \leftarrow A_u^{r/s} \cup \{A_n\}$ /* Add the $n^{th}$ attribute in $A^s$ */
            end
        end
    end
end
Step 2: Suppress Attribute Values as Necessary for the User
    while $R^u \neq \phi$ $OR$ $A_u^{r/s} \neq \phi$ do
        $A_n \leftarrow maxarg(Counter)$ /*Indentify the attribute that appears the most in $R^u$*/
        SuggestSuppress($A_n$) /*Suggest the user to suppress the attribute value for $A_n$*/
        $A_u^{r/s} \leftarrow (A_u^{r/s} \setminus \{A_n\})$
        $Counter \leftarrow (Counter - Counter_n)$ /*The counters are kept aligned with the attributes*/
        if $A_n$ is suppressed then
            $R^u \leftarrow (R^u \setminus FindRules(R^u, A_n))$ /*Rules using $A_n$ in preconditions are removed*/
        end
    end
end
Step 3: Hide Friendship Links as Necessary for the User
    $A_n \leftarrow FindMostSensitive(A^l, R^u, U, G)$/*$A_n = Val(m(u, a))$*/
    while $A_n \neq null$ do
        $a \leftarrow WhichAttr(A_n, A^l)$
        foreach $R_j^u \in R^u$ do
            if $A_n \in IsTested(R_j^u)$ and $Val(A_n) \geq SplitPoint(R_j^u, A_n)$ then
                while $Val(A_n) \geq SplitPoint(R_j^u, A_n)$ and $MoreFriends(U, G)$ do
                    $f \leftarrow FriendWithLeastDegree(G, D, U, a)$
                    SuggestHide($f$)
                    if $t \in IsHidden(f)$ then
                        $G \leftarrow RemoveLink(G, U, f)$
                        $Recompute(A_n, G, D, U)$
                    end
                end
            end
            $R^u \leftarrow R^u \setminus \{R_j^u\}$
        end
        $A_l \leftarrow A_l \setminus \{A_n\}$
        $A_n \leftarrow FindMostSensitive(A^l, R^u, U, G)$/*$A_n = Val(m(u, a))$*/
    end
end
Step 4: Add New Friends as Necessary for the User
    $A_n \leftarrow FindMostSensitive(A^l, R^u, U, G)$/*$A_n = Val(m(u, a))$*/
    while $A_n \neq null$ do
        $a \leftarrow WhichAttr(A_n, A^l)$
        foreach $R_j^u \in R^u$ do
            if $A_n \in IsTested(R_j^u)$ and $Val(A_n) \leq SplitPoint(R_j^u, A_n)$ then
                while $Val(A_n) \leq SplitPoint(R_j^u, A_n)$ and $MoreUsers(U, G)$ do
                    $f \leftarrow UserWithLeastDegree(G, D, U, a)$
                    SuggestAdd(f)
                    if $t \leftarrow IsAdded(f)$ then
                        $G \leftarrow AddLink(G, U, f)$
                        $Recompute(A_n, G, D, U)$
                    end
                end
            end
            $R^u \leftarrow R^u \setminus \{R_j^u\}$
        end
        $A_l \leftarrow A_l \setminus \{A_n\}$
        $A_n \leftarrow FindMostSensitive(A^l, R^u, U, G)$/*$A_n = Val(m(u, a))$*/
    end
end
```

Earlier we saw that *PrivAdv* [1] could secure the confidential attributes of only 56.62% users from the attribute inference attack that uses link information on the D'_{FB} dataset. However, using algorithm *3LP* the remaining 43.38% users are protected. Later 1 is essentially *PrivAdv*, none of the information of the users at risk is secured further. Typically, for a group of 61 users, 27 users are still at risk after Layer 1. But, on average, 5 of them can prevent a breach of privacy by hiding friends. In percentage terms, users whose confidential attribute is secure increases to 64.52% after Layer 2, with a 7.9% increment with respect to Layer 1. Although hiding a particular friend from user profile is currently unavailable on Facebook these results suggest that the operators of OSN such as Facebook may consider adding this option to a user profile. That is, enable users to select the automatic masking of some friendships to any data analyst so their confidential attribute (already not present) can not be inferred.

Moreover, to secure the data of the remaining users, our experimental results show that on an average 22 users need to add more friends to prevent a breach of privacy. (i.e., Layer 3 of *3LP*). Of the users who are not protected by previous approaches (Layer 1), equivalently 83.84% (22 out of 27) need to do it by adding friends. While choosing the friend during addition, lower degree friends carry more impact on the metric function values.

Although adding more friends may seem unrealistic in OSNs settings, and other risks may derive from linking with strangers, we believe the operators of OSNs would be able to perform this. Certainly ensuring the privacy of their users is in the operators' best interest, Thus, our results here suggest that operators can suggest to users the addition of some synthetic friends. Alternatively, they could use such technique to sanitise the data before releasing it to data analysts. We plan to focus on this in our future work. On the other hand, in Table 5 we present respectively the experimental results with $D'_{Political}$ and D'_{Sexor}. The average results show that, for a group of 100 users, about 23 and 3 (after rounding) users are still insecure after applying the first layer of *3LP* on $D'_{Political}$ and D'_{Sexor} data sets respectively.

In order to secure these users we then apply Layer 2 of *3LP* (i.e., obfuscate friends from friend lists) and we notice that no more users are at risk (after applying Layer 2 of *3LP*) in both $D'_{Political}$ and D'_{Sexor} data sets. Hence Layer 3 of *3LP* is not required in our experiments for both of these data sets.

Table 5. Number of insecure users after applying *3LP* on the expanded dataset $D'_{Political}$ and D'_{Sexor}.

$D'_{Political}$		Number of insecure users			
Run	Number of users in D_i	After using *PrivAdv*	After using Layer 1 of *3LP*	After using Layer 1 and Layer 2 of *3LP*	After using Layer 1, Layer 2 and Layer 3 of *3LP*
1	100	17	0	X	X
2	100	15	0	X	X
3	100	12	0	X	X
4	100	100	99	0	X
5	100	20	0	X	X
6	100	35	0	X	X
7	100	29	0	X	X
8	100	97	95	0	X
9	100	53	39	0	X
10	100	39	0	X	X
Average	**100**	**41.7**	**23.3**	**0**	**X**

D'_{Sexor}		Number of insecure users			
Run	Number of users in D_i	After using *PrivAdv*	After using Layer 1 of *3LP*	After using Layer 1 and Layer 2 of *3LP*	After using Layer 1, Layer 2 and Layer 3 of *3LP*
1	100	69	1	0	X
2	100	69	0	X	X
3	100	67	0	X	X
4	100	75	7	0	X
5	100	86	15	0	X
6	100	77	0	X	X
7	100	60	0	X	X
8	100	74	0	0	X
9	100	63	5	0	X
10	100	65	0	X	X
Average	**100**	**70.5**	**2.8**	**0**	**X**

Table 6. Required number of attribute Suppression, Friend Deletion or Addition for each insecure user suggested by *3LP* in expanded data set D'_{FB}.

Run	Average Number of attribute suppression (per user) in Layer 1 of *3LP*	Average number of friends needed to be hidden (per user) in Layer 2 of *3LP*	Average number of friends needed to be added (per user) in Layer 3 of *3LP*
1	0	1	2
2	0	1	1
3	0	1	1
4	0	0	1
5	1	2	2
6	0	1	2
7	1	1	1
8	0	0	1
9	1	1	2
10	0	1	2
Average	0	**1**	**2**

The Column 2 of Table 6 shows the number of attributes needed suppression in Layer 1 of *3LP*. Please note that these are the suppressions made in addition to the suppressions suggested by the regular *PrivAdv*. The average number of attribute suppression (Layer 1 of *3LP*), on the other hand, is higher both in $D'_{Political}$ and D'_{Sexor} compared to D'_{FB}. The reason may be the number of generated SRR (i.e., sensitive rules with regular attributes) is much lower for D'_{FB}.

Our results also show that the burden of additions and obfuscations of friends is not that large. For example, in D'_{FB} data set, we need to hide/add at most 1-2 friends, on average, in each partition D^i to secure the confidential attribute (refer to Table 6).

Table 7. Required number of attribute Suppression, Friend Deletion or Addition for each insecure user suggested by *3LP* in expanded data sets $D'_{Political}$ and D'_{Sexor}.

$D'_{Political}$ Run	Average Number of attribute suppression (per user) in Layer 1 of *3LP*	Average number of friends needed to be hidden (per user) in Layer 2 of *3LP*	Average number of friends needed to be added (per user) in Layer 3 of *3LP*
1	1	X	X
2	1	X	X
3	1	X	X
4	1	10	X
5	1	X	X
6	1	X	X
7	1	X	X
8	1	6	X
9	1	11	X
10	2	X	X
Average	**1.1**	**2.7**	**X**

D'_{Sexor} Run	Average Number of attribute suppression (per user) in Layer 1 of *3LP*	Average number of friends needed to be hidden (per user) in Layer 2 of *3LP*	Average number of friends needed to be added (per user) in Layer 3 of *3LP*
1	2	11	X
2	2	X	X
3	2	X	X
4	1	9	X
5	2	7	X
6	1	X	X
7	1	X	X
8	3	X	X
9	1	14	X
10	2	X	X
Average	**2**	**4.1**	**X**

In case of the data sets $D'_{Political}$ and D'_{Sexor}, we need to hide 3-4 friends, on average, in each partition whereas, Layer 3 is not required in our experiments (refer to Table 7).

5 Conclusion

We proposed *3LP*, a privacy-preserving technique in order to protect the privacy of Facebook users from attribute inference risks. Previous works did not consider friendship network information which may create vulnerability to users' privacy. Our technique provides suggestions, to a user to suppress necessary attribute values and fabricate friendship links, in order to protect sensitive attribute values of the user. The technique can also enable a social network provider to query a user whether to fabricate such links to preserve his/her privacy. Our experimental results show that by hiding or adding a few friends in a user's profile can protect the user's sensitive information from being inferred. Though hiding a particular friend from the user's profile is currently unavailable on Facebook, the approach here suggests that such feature could be added in order to protect users' privacy.

In this paper, we have considered that only the user or few others in user's network are consumers of *3LP*. If all friends in a user's friend list continuously use and adopt the recommendations of *3LP*, then the calculation will be dynamic and different. We believe this is an exciting avenue for further research.

References

1. V. Estivill-Castro, P. Hough, and M. Z. Islam. Empowering users of social networks to assess their privacy risks. *2014 IEEE Int. Conf. Big Data*, p. 644–649, 2014.
2. Facebook stats. `newsroom.fb.com/company-info/`. [Online; 03-June-2016].
3. The top 500 sites on the web. `//www.alexa.com/topsites`. [Online; 03-June-2016].
4. A. Ho, A. Maiga, and E. Aimeur. Privacy protection issues in social networking sites. *IEEE/ACS Int. Conf. Computer Systems and Applications*, p. 271–278, 2009.
5. Herman T. Tavani. *Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing.* Wiley Publishing, Hoboken, NJ, 3rd edition, 2011.
6. J. Rachels. Why privacy is important. *Philosophy & Public Affairs*, 4(4):323–333, 1975.
7. R. Heatherly, M. Kantarcioglu, and B. M. Thuraisingham. Preventing private information inference attacks on social networks. *IEEE Trans. on Knowl. and Data Eng.*, 25(8):1849–1862, August 2013.
8. E. Ryu, Y. Rong, J. Li, and A. Machanavajjhala. curso: Protect yourself from curse of attribute inference: A social network privacy-analyzer. *SIGMOD Workshop on Databases and Social Networks*, DBSocial '13, p. 13–18, NY, 2013. ACM.
9. Y. Al-Saggaf and M. Z. Islam. Privacy in social network sites (SNS): The threats from data mining. *Ethical Space: Int. J. Communication Ethics*, 9(4):32–40, 2012.
10. S. Guha, K. Tang, and P. Francis. NOYB: Privacy in online social networks. *First Workshop on Online Social Networks*, WOSN '08, p. 49–54, NY, USA, 2008. ACM.
11. N. Z. Gong, A. Talwalkar, L. W. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song. Joint link prediction and attribute inference using a social-attribute network. *ACM Trans. Intell. Syst. Technol.*, 5(2):27, 2014.
12. V. Estivill-Castro and D. F. Nettleton. Can on-line social network users trust that what they designated as confidential data remains so? *2015 IEEE Trustcom/BigDataSE/ISPA*, p. 966–973, Washington, 2015. IEEE Computer Soc.
13. L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.
14. T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. *19th Int. Conf. on World Wide Web*, WWW '10, p. 601–610, NY, 2010. ACM.
15. D. F. Nettleton. Generating synthetic online social network graph data and topologies. *3rd Workshop on Graph-based Technologies and Applications*, Graph-TA '15, UPC, Barcelona, Spain, 2015.
16. Z. Islam and H. Giggins. Knowledge discovery through SysFor: A systematically developed forest of multiple decision trees. *9th Australasian Data Mining Conference V 121*, p. 195–204, Darlinghurst, 2011. Australian Computer Soc.