How to share your favourite search results while preserving privacy and quality

George Danezis[1], Tuomas Aura[2], Shuo Chen[1], and Emre Kıcıman[1]

[1]Microsoft Research,
One Microsoft Way, Redmond, WA 98052-6399, U.S.
{gdane, shuochen, emrek}@microsoft.com.

[2] Helsinki University of Technology,
P.O.Box 5400, FI-02015 TKK, Finland.
tuomas.aura@tkk.fi

Abstract. Personalised social search is a promising avenue to increase the relevance of search engine results by making use of recommendations made by friends in a social network. More generally a whole class of systems take user preferences, aggregate and process them, before providing a view of the result to others in a social network. Yet, those systems present privacy risks, and could be used by spammers to propagate their malicious preferences. We present a general framework to preserve privacy while maximizing the benefit of sharing information in a social network, as well as a concrete proposal making use of cohesive social group concepts from social network analysis. We show that privacy can be guaranteed in a k-anonymity manner, and disruption through spam is kept to a minimum in a real world social network.

1 Introduction

A fundamental problem contemporary web-based information retrieval (IR) face is *ranking*. Given a user query, the IR system has to produce a ranked subset of documents that are most likely to satisfy the user's information needs. To achieve this, techniques beyond simple indexing are required and there are benefits in taking into account social structure when searching for information [15]. Recent research [20] suggests that users' information needs are correlated: it is likely that a document that has been accessed by Alice will also be relevant to her friend or colleague Bob. If only Alice and Bob were able to make use of this information, their search results could be improved.

Two key security problems have to be addressed to enable the sharing of *preferences* about search results and documents in a social network, namely *privacy* and *quality*.

Privacy is necessary to ensure that users do not learn about each others' exact search patterns or retrieved documents. It is unacceptable to allow particular query items or documents to be linked with certainty to a user by third parties. In this work we consider privacy guarantees against both adversarial sybil nodes that infiltrate the network, as well as curious coalitions of the users' friends.

Quality in the context of security means that the ranking system should not be overly influenced by nodes that maliciously inject information to manipulate the ranking of certain resources. Search engine spamming is a serious problem, and any sharing of information has the potential to provide the spammers with an additional tool. The key goal of our scheme is to limit the influence of spammers to mostly those nodes that consider them as 'friends' and limit any further spread of their poisoned preferences.

Our approach to solving this problem involves propagating the user's useful search results—more generally we call this the user's *preferences*—within a random subgroup of the user's social network. We create those subgroups carefully to ensure they are *cohesive*, i.e., with very high density of links between all nodes. The subgroups form a core anonymity set, and are infiltration resistant to prevent spammers from being able to send their preferences to everyone. We present a general model that can be instantiated in many ways depending on the choice of cohesive subgroup – our concrete solution uses the *k-plex* definition [24].

We note that the problem of anonymously propagating information with a social network is far from unique. Similar systems are required for viral marketing, where products are recommended to users according to whether someone socially related to them bought them. Restaurant or movie recommendations are another example of systems that benefit from users socially sharing their preferences, without leaking specifics about what they see or where they are. Generally our solution applies to any system that (a) collects user preferences, (b) aggregates them centrally or locally on a social graph, (c) does some processing operation on the aggregate, (d) and returns the result, or influences the output to users. We will use the concrete example of personalised social search throughout this work, while engineering our solution to be general to the full class of problems.

After reviewing the literature on personalised social search in Section 2, we define an abstract model of our problem and the families of solutions we consider in Section 3. Then in Section 4 we propose a concrete strategy for sharing information in personalised search using cohesive social sub-groups and study the extent to which it satisfies our goals. In the final section we discuss some nuances of such system and offer conclusions.

2 Related Work

Personalized search, that tailors web search results based on preferences of users, is already widely deployed by major search engines [27, 30]. Personalized social search goes a step further and determines the ranking of documents based on the preferences within the social network of users. It is already piloted by smaller online search engines, like Eurekster [29]. Google is currently piloting a mechanism that allows users to re-rank results [16]. The re-rankings are not directly shared but used centrally to increase the quality of the overall results. The Microsoft Research U Rank prototype [18] allows users to re-rank their results, and share them with their direct friends, without any further provision for privacy.

Many studies have looked into the privacy preferences of users, in relation to information they share over social networks [21, 1]. They conclude that search preferences are considered sensitive, and the controversy surrounding AOL search data leak confirms this[1].

Eurekster [29] allows users to designate search mates, with whom they share their search preferences. Effectively any query and subsequent information is shared within this group of friends. Some primitive privacy features are provided through the ability to perform private searches, as well as the ability to delete past searches from being visible to others. Our approach, on the other hand, allows users to share, to some degree, their preferred search results, without compromising their privacy. Additional privacy controls, based on opt-outs like in Eurekster, are orthogonal to our scheme and can be applied independently.

Social networking site, like Facebook,[2] have also tried to share user preferences amongst friends, but for the purposes of viral marketing. The "Facebook Beacon" system caused controversy by sharing user's preferences, often generated outside of the Facebook site, with their network of friends. The initial privacy strategy of an opt-out mechanism was turned into an opt-in mechanism after some pressure [10].

We use the naive sharing strategy of simply broadcasting preferences to the sets of friends or friends-of-friends of a node as a benchmark to assess the security benefits of our proposal. Without better privacy and quality preserving techniques, this naive scheme is the one most likely to be deployed, as has been the case in Facebook Beacon and Eurekster.

A serious body of scientific work is concerned with preserving privacy in on-line services. Our schemes borrow privacy notions like k-anonymity from the literature on data sanitation and anonymization [23, 2]. The basic premise of those schemes is that any inference drawn by an observer should be attributable to at least k participants, effectively forming an anonymity set. To our knowledge, this is the first time that k-anonymity is used in the context of data mining on social networks.

The adversary model we consider—an attacker is assumed to control a very large number of nodes in the network—was first introduced in the context of peer-to-peer systems by Douceur as the Sybil attack [13]. Our approach is centralised, and admission control [3] as well as intrusion detection methods could be used to keep the number of corrupt nodes down. Despite this, we aim to resist attacks without such measures, keeping the cost of running the system down and relying on distributed trust decisions for security. These two approaches are complementary and can be combined.

Our security assumptions to combat sybil attacks aiming to degrade privacy and quality are based on the tradition of SybilGuard [33], SybilLimit [32] and SybilInfer [7]. They assume that honest nodes form a connected social graph, and only few misguided nodes introduce an unbounded number of adversary

[1] CNN money included AOL releasing search data as #57 of its "101 Dumbest Moments in Business" for the year 2007.

[2] `http://facebook.com`

nodes. This small number of nodes or links to bad nodes can be used as a 'choke point' to limit the impact of the adversary on the running of the system. The idea of using the social structure itself to fend systems against those attacks was first proposed in [6] and [19].

For privacy we also consider a more traditional threat model, in which a coalition of a user's friends is curious to find out her preferences. The assumption of a limited fraction of dishonest or misguided nodes in a set goes back to work on secret sharing [26], threshold cryptography [28] and double entry book keeping in banking [22].

3 Model of anonymity in preference sharing

Preference sharing has often been implemented with little regard to privacy. In this section we cast the problem of sharing preferences privately against an adversary (sections 3.1 and 3.2). We discuss how to correctly measure anonymity (section 3.3), as well as a generic framework that achieves privacy and utility for preference sharing (section 3.4). Finally, we discuss how quality is preserved in our model (section 3.5).

3.1 Preference sharing

The most basic concepts in our model are the universe of users U and the universe of preferences P. We say that a user $u \in U$ may *set a preference* $p \in P$. The system then *propagates* the preference from the *source* u to a set of users $T \subseteq U$, which is called the *target group*. We also say that the source user has an *initial preference* and the target users have *propagated preferences*.

We assume that users submit their preferences to a trusted centralised system, that is in charge of performing the search and ranking of results, as current search engines are. The target group for the propagated preferences is chosen by the system from *possible target groups* $\mathsf{Groups}(u, p) \subseteq \mathbf{P}(\mathbf{P}(U))$ (a set of sets of users). We also assume that each preference is set by only one user at a time, which simplifies the model greatly, as we will see, without loss of generality.

Note that the source user itself does not decide the possible target groups or the actual target group. The system chooses the target group based on a *propagation policy*, which is partly specified by the function Groups. The goal of this paper is to find a propagation policy that meets several sometimes conflicting criteria:

1. First, the policy should preserve privacy.
2. Second, the policy should take into account social relations between the users to increase the relevance of propagated policies to the target users.
3. Third, the policy should be easy to implement.

The selection of the target group may be deterministic or nondeterministic. With deterministic target selection, $|\mathsf{Groups}(u, p)| = 1$ for all u and p. With nondeterministic selection, there can be multiple possible target groups and the actual target group is chosen randomly from them. (For the time being, let's assume uniform random selection.) An interesting case is one where the target group is selected from multiple possibilities based on a pseudorandom function and a secret key. In that case, the selection process is similar to a random oracle: the target group T is chosen randomly from $\mathsf{Groups}(u, p)$ for each new u, p pair but, if the selection is repeated for the same parameters, the target group will not change.

3.2 Anonymity and the adversary model

After the preference setting and propagation, each user has a set of initial preferences, which remains secret to that user, and propagated preferences, which are considered public[3]. The adversary is a coalition of users that observe the propagated preferences and try to determine which user initially sets each preference. We base our analysis on a rather strong adversary that knows the function Groups and can observe all the propagated preferences. Real-world systems can of course make it difficult for the adversary to observe all preferences through access control, network security and cryptography.

The assumption that each preference is set by at most one user at a time, is explained by the following: we assume that the attacker can observe the target group for each instance of setting the preference, rather than observing only the end result of multiple users setting the preference. This is a kind of worst-case scenario, but also corresponds with the fact that users are unlikely to set their preferences at exactly the same time and each act of setting may affect the propagated preferences for other users.

After observing a preference p propagated to a target group T, the adversary can narrow down the identity of the source to the following set:

$$U_{\mathsf{anon}} = \{u' \in U \mid T \in \mathsf{Groups}(u', p)\} \tag{1}$$

This set is called the *anonymity group*. The adversary knows that one member of the anonymity group initially set the preference. The size of the anonymity group $|U_{\mathsf{anon}}|$ can be used as a measure of anonymity. This is similar to k-anonymity in computer-security literature [4]. Note that here U_{anon} and k depend on u and p. We say that a preference propagation policy *preserves k-anonymity* if $k \leq |U_{\mathsf{anon}}|$ for all u and p.

The relation between the members of the anonymity group must be *symmetric* in the sense that, for a given preference and target group, if u' is in the anonymity group when the real source is u, then u is in the anonymity group when the real source is u'. This is natural because an anonymity group arises from the fact that any one of them could be the real source.

The adversary defined above corresponds to an outsider who can require all users to reveal their propagated preferences but does not have access to anyone's

[3] This is a modeling assumption, and real world systems may further limit their visibility.

initial preferences. This could, for example, be someone who demands that users show their current search results, which are influenced by propagated preferences.

We are also interested in an adversary that has, additionally, access to the initial preferences of some colluding users. These could, for example, be a set of friends who try to figure out the source of a preference propagated to them. For an adversary with the combined knowledge of a coalition of users U_{bad}, the anonymity set is reduced to $U_{\mathsf{anon}} \setminus U_{\mathsf{bad}}$. In practical situations, however, we expect the size of the coalition to be small, often just a single user. This is because the members of the coalition need to trust each other to tell the truth about their initial preferences, and because sybil attacks will be prevented by the user of social networks (see section 3.4).

From equation 1, we make the important observation that privacy does not depend on the random selection of the target group T. A deterministic algorithm could be just as anonymity-preserving, as long as it picks the same target group for several users. Randomized selection does not guarantee anonymity either: it needs to be carefully designed to produce anonymity sets of sufficient size. This is why we consider both deterministic and nondeterministic propagation algorithms.

Finally, we make a couple of further observations. First, the target groups cannot be selected independently for each source user because they need to coincide, or otherwise the anonymity sets will be small. This has implications to the extent that the target group selection can be distributed. Second, the possible target groups for each preference can be selected independently of other preferences. The parameter p is carried in the notation as a reminder of this fact. Third, if privacy is the only goal, we could just as well select the empty target group (no preference sharing) or the all-users group U (share with everyone). This is in fact the current practice of recommender systems (such as Amazon or Netflix). The reasons for selecting something in between, which will be discussed in section 3.4, are unrelated to privacy, but crucial for adding value to search while preventing spam.

3.3 Probabilistic anonymity model

Above, we have not considered the probability distribution between different choices of target groups. This lead to using k-anonymity as the measure of privacy: the anonymity group includes everyone who might be the source, no matter how unlikely it is. Now, we extend the model to take into account probabilities. Given a source u and a preference p, the probability distribution of target groups is denoted by $P(u, p, T)$. The function Groups can now be defined as

$$\mathsf{Groups}(u, p) = \{T \subseteq U \mid P(u, p, T) > 0\}$$

As established in the literature [25, 11], anonymity in the probabilistic model is measured by entropy, i.e., the adversary's uncertainly about the identity of the source. Entropy is measured in bits, i.e., how many more bits of information would the adversary need to be certain of the source identity. We assume that all users are initially equally likely to be the source (equal a-prior probabilities), and that only one at a time sets the preference. When the adversary observes a preference p propagated to a target group T, the entropy for the source can be calculated as follows.

$$H(u|p,T) = \sum_{u \in U_{\mathsf{anon}}} (\frac{P(u,p,T)}{S}) \cdot (-\log_2(\frac{P(u,p,T)}{S})) \text{ where } S = \sum_{u \in U_{\mathsf{anon}}} P(u,p,T)$$

What can we learn from this? Obviously, the larger the anonymity set, the higher the entropy. Analogous to our earlier comparison of deterministic and nondeterministic propagation policies, we also note that it makes no difference how many different choices $|\mathsf{Groups}(u,p)|$ there are for T. The most important lesson from the above formula is that, given a fixed-size anonymity set, the entropy is maximized when all members of the anonymity set are equally likely to choose the specific target group. It does not matter how or whether this probability is large or small, as long as it is uniform across the possible sources.

3.4 Preference sharing in a social network

The privacy model above does not explain why we want to propagate the preferences in the first place. Our aim is to select a target group that is by some measure *close* to the source, so that the propagated preferences are relevant to the group. This will not only result in more effective use of the preference information but also in *spam resistance*. It is important to note, however, that there is no simple right way for defining closeness between users. Before considering possible definitions, we will consider some general factors in propagation policies that are based on the concept.

Since the preferences set by a user are naturally closest to its own needs, we only consider propagation policies that are *reflexive* in the sense that each user is in all of its own propagation targets:

$$T \in \mathsf{Groups}(u,p) \text{ implies } u \in T. \tag{2}$$

In a reflexive propagation policy, the anonymity group is always a subset of the target group.

For a given target group T, we denote $U_{\mathsf{ext}} = T \setminus U_{\mathsf{anon}}$. Thus, the target group is the disjoint union of the anonymity group and an *extended group*: $T = U_{\mathsf{anon}} \dot{\cup} U_{\mathsf{ext}}$.

The discussion so far gives one possible outline for constructing propagation algorithms. The algorithm can be executed independently for each preference p, or the same groups can be used for many preferences:

1. Select anonymity groups in such a way that they cover all users U. Members of the anonymity set should be close to each other, based on some arbitrary social metric.
2. For each anonymity set, decide on the extended groups. The members of the extended groups should be close to the members of the anonymity set, but not necessarily to each other.

It makes sense to start by fixing the anonymity groups because that is an easy way to guarantee k-anonymity. If we instead expected anonymity to arise probabilistically, it would be difficult to guarantee that they all will be sufficiently large. The members of each anonymity group need to be all close to each other because any one of them could be the source. The members of the extended group, on the other hand, are targets and need to be close to potential sources.

A simpler model would be one where $U_{\mathsf{ext}} = \emptyset$ and $T = T_{\mathsf{anon}}$. In this restricted model, preferences are shared mutually among a sets of users who are close to each other, such as the members of a club or a clique of users who all know each other. An advantage of the more general model, especially when $|T| \ll |U_{\mathsf{anon}}|$, is that the preferences can be propagated to a larger number of target users without any reduction in anonymity. In practical social networks, we are looking at anonymity sets of around ten users and target groups that are one order of magnitude larger (as studied is section 4.3).

We are particularly interested in social networks that are based on a *friendship graph* $G \subseteq U \times U$. The friendship relations in this kind of graph are typically symmetric, which means that any metric of closeness between members will be symmetric as well. In the above outline for propagation algorithms, the first step is to select anonymity sets in such a way that all their members are close to each other.

3.5 Spam resistance

For the purposes of modeling spam resistance, we categorise nodes in the system as being in one of three categories: *honest nodes* genuinely share their preferences, and *dishonest nodes* try to propagate to honest nodes spam preferences. We consider that a class of honest nodes are *misguided* in that they have created friendship links with dishonest nodes.

We can use this intuition to build anonymity sets U_{anon} and broadcast groups U_{ext} that are *infiltration resistant*. This means that once a number of honest nodes are part of a group they are unlikely to form links with dishonest nodes, thus disallowing them from broadcasting their preferences within the group. Part of our security analysis is concerned with validating this property in a real-world social network.

4 Outline of solution

We propose a concrete nondeterministic propagation strategy that is based on broadcasting users' preferences within socially cohesive subgroups. The subgroups can be overlapping, and are formed by k-plexes of some s-minimal size. A k-plex is a sub-graph of size $g_s \geq s$ of the social network in which all nodes link to at least $g_s - k$ other nodes in the sub-graph. It is an established relaxation of cliques (which are a special case for $k = 1$) that defines robust and cohesive subgroups, extensively used in social network analysis [24].

The properties of s-minimal size k-plexes make them a very good fit for supporting our security and functional properties. The parameters k, defining how many links can be missing within a subgroup, as well as s, the minimal size of the subgroup, are naturally related to quality and privacy.

First, k-plexes of a minimum size s are *infiltration resistant*. For a single node of a coalition of c nodes to be part of a k-plex they need to form a large number of links l_c:

$$l_c = \max(s - k, [(s - k) - (c - 1)] \cdot c) \geq s - k \tag{3}$$

This has a direct security implication for quality since a small number of misguided nodes in a k-plex forming links with adversary nodes, will not allow those nodes to infiltrate the k-plex, containing other honest nodes. (Although misguided nodes can be conned into joining k-plexes dominated by corrupt nodes.) Therefore limiting broadcast of preferences within those sub-groups curbs the potential for abuse and spam – an adversary will have to invest a lot of effort to infiltrate them, and a few vigilant members of each group will be able to thwart such actions.

While a set of k-plexes form the anonymity groups U_{anon} each of them is augmented by a set of additional nodes i.e., the extended broadcast group U_{ext}. Membership of nodes to the extended broadcast group is parameterized by a threshold T on the number of friends a node has that belong to the anonymity group U_{anon}. If a node has T or more friends in U_{anon} then it belongs to the extended broadcast group U_{ext}.

4.1 The *preference-sharing* algorithm

The preference sharing algorithm works in two phases. First a pre-computation extracts cohesive sub-groups that are used to form anonymity sets U_{anon}, and their corresponding extended broadcast groups U_{ext}. Only the structure of the social network is required to perform group extraction. In a second phase preferences are continuously set by users and are propagated to other users through the extracted groups. Only propagated preferences are collected to compute the ranking of resources for each user, and the initial preferences can even be forgotten.

The parameters of the *preference-sharing* algorithm are:

- k, the parameters of the k-plexes we use.
- s_a, the size of the anonymity set required.
- T, a threshold that defines the extended broadcast groups membership.

We require $s_a > 2k$ and $T > 1$ (in our analysis we use $k = 2$, $s_a = 8$ and $T = 2$). These conditions ensure that the diameter of the sub-groups extracted is at most 2 [31]. This in turn enforces strong locality and makes the extraction of the cohesive subgroups faster.

Sub-group extraction. First the anonymity sets are extracted. Given the social graph G a set of k-plexes of size s_a is extracted and associated with each user. These are the sets U_{anon} that form the core anonymity sets providing

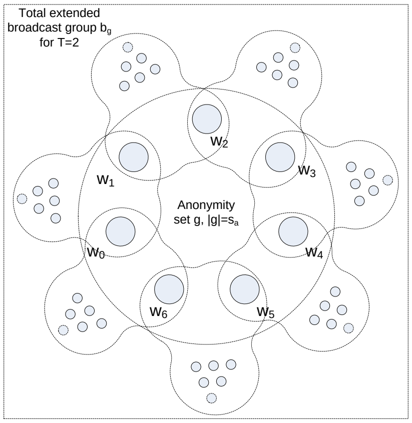

Fig. 1. An illustration the anonymity groups and the broadcast groups selected to propagate a single preference.

privacy for preference propagation. Second the extended broadcast groups for each anonymity set are extracted. For each cohesive subgroup U_{anon} we define a broadcast set U_{ext} containing all nodes that are friends with at least T members of the cohesive subgroup. This defines a 'wider circle' of people around each subgroup to which preferences will also be broadcast.

Sub-group extraction is not necessarily real-time and can be performed periodically depending on how often the social graph changes. The anonymity sets for each user contained in U_{anon} as well as their broadcast groups U_{ext} can be reused for propagating multiple preferences. Sub-group extraction does not need to be exhaustive either. In this work we chose to extract the set of k-plexes for each user that contain at least all neighbours of each node which share a k-plex with the user. This strategy ensures that all the friends that share a cohesive subgroup with a node could possibly be receiving the user's preferences.

Preference-propagation. At some point in time, a user u sets a preference for a p. Our system chooses at random a k-plex containing the user $g \in_R \{U_{\mathsf{anon}}\}$ to act as the anonymity set for this preference. If there is no such k-plex no propagation of results takes place, and the algorithm ends. Otherwise, the preference of node u is broadcast to all nodes in $U_{\mathsf{anon}} \cup U_{\mathsf{ext}}$, i.e., the anonymity set and the extended broadcast group corresponding to the selected anonymity set U_{anon}.

Each node $v \in b_g$ aggregates all preferences broadcast in a multiset of preference P_{vi} relating to a resource i. Each broadcast updates the multiset with the received preference $P'_{vi} = \{f(i,u)\} \biguplus P_{vi}$. A simple function can then be applied to this multi-set of preferences to determine the final preference of this each node relating to each resource $g(i,v)$.

4.2 Privacy Analysis

Our first task is to evaluate the privacy offered by the preference-sharing algorithm, against two types of adversaries. The first is a very powerful global adversary, that can see the preferences output by the preference sharing algorithm for every single node in the network. Yet this adversary is passive in that it does not know the private inputs to the algorithm and tries to infer them. The second threat we consider is a curious coalition of a user's friends, that wants to infer what her preferences are.

Global passive adversary. We assume that an eavesdropper can see all the propagated preferences. Through those they can extract the sub-group g (of size s_a) that formed the core of the anonymity set used to propagate a particular preference $f(i, u)$.

Any of the members of g *could* have been the originators of the preference. This already ensures some plausible deniability and privacy to the real originator. To be more specific one has to calculate the probability a user set a preference given that it was broadcast in sub-group g, that we will denote as $\Pr[u|g]$. By applying Bayes theorem we can express it in terms of known quantities:

$$\Pr[u|g] = \frac{\Pr[g \in_R G_u] \Pr[u]}{\sum_{w \in g} \Pr[g \in_R G_w] \Pr[w]} \tag{4}$$

$\Pr[g \in_R G_u]$ is the probability that a user u chooses group g and $\Pr[u]$ is the a-prior probability we assign to user u being the originator of a preference i. If we assume that the a-prior probability over all users is uniform, and that they all choose the sub-groups $g \in_R G_u$ uniformly out of the sets G_u we get:

$$\Pr[u|g] = \frac{1}{|G_u| \sum_{w \in g} \frac{1}{|G_w|}} \tag{5}$$

In case all users chose amongst a set of fixed size $|G_u| = c$, this expression simplifies, and the sought probability becomes: $\Pr[u|g] = 1/s_a$. This related nicely the parameter s_a of the algorithm with the privacy provided. The larger s_a the larger the anonymity provided, when measured information theoretically.

Yet there is likely to be an imbalance between the sizes of the sets G_u for different users. We try to establish what the worse case scenario is, assuming that we have some maximal size of $\max |G_u| = c_{\max}$ as well as some minimal size $\min |G_u| = c_{\min}$. In those cases we still have that:

$$\Pr[u|g] < \frac{c_{\max}}{c_{\min}(s_a - 1) + c_{\max}} \tag{6}$$

This expression makes it possible to compute the probability a preference is associated with a user. A system can either try to keep it low by choosing carefully sub-groups to guarantee $c_{\min} \leq |G_w| \leq c_{\max}$, or simply not propagate preferences in case this probability is higher than a threshold.

The adversary model assumed is extremely conservative, assuming that most information in the system is available to pinpoint g. It is most likely that coalitions of dishonest nodes will receive much less information. In particular a single node in the system will not be able to distinguish which of the nodes in its set b_u was the originator of a preference.

Yet an important concern is the possibility that nodes in the anonymity set g are in fact corrupt. We assume this is very difficult since k-plexes are infiltration resistant. Sharing a k-plex of size s_a with s_a dishonest nodes, requires a misguided node to make $s_a - k + 1$ bad friends. In any case such an attack would only affect misguided nodes in the system, which we assume are in a minority.

Honest nodes (with mostly honest friends) will never find themselves in a k-plex dominated by dishonest nodes. Even a misguided node with fewer than $s_a - k + 1$ dishonest friends will never have their privacy totally compromised through infiltration.

Coalition of curious friends. The second key threat to the privacy to users are their very own friends. A collection of a user's friends may exchange information about their private preferences in an attempt to infer the preferences of a user.

First we note a very strong privacy property against such attacks. In any case coalitions of fewer than $s_a - 1$ users will fail to attribute a preference with certainty to a single user. This is a very strong result that sets a lower bound on the size of the conspiracy.

At least $s_a - 1$ nodes are necessary to fully de-anonymize a preference, but this condition is not sufficient to perform an actual attack. It is also necessary that the coalition of node coincides exactly with the members of the cohesive sub-group used as an anonymity set to broadcast the preference. This places additional restrictions and difficulties in creating such a malevolent coalition.

Through simulations we try to estimate the quality of anonymity remaining after such an attack. For those we use about 100000 user profiles downloaded through the `livejournal` public interfaces using snowball sampling. Only symmetric links were kept to form a social graph. We assume that a fraction f of all users collude to deanonymize users. These users are curious but make no special effort to place themselves in the social graph to maximise the information they receive (they are not as such sybil nodes – just curious friends.) Therefore we assume they are randomly distributed across the network.

Figure 2 summarises the results of attack simulations on a real-world social network. The Preference Propagation (PP.) algorithm (yellow, right hand size columns) is compared with the naive strategy (Std.) of broadcasting preferences to all friends (red, left hand side columns.) When a very low number of nodes collude to infer a user's preference ($f = 1\%$) the naive scheme provides good anonymity, since on average a corrupt users cannot narrow down the originator of a preference beyond his full circle of friends. Yet as the fraction of curious nodes grows ($f = 10\%, 20\%, 30\%$) the anonymity sets for the standard strategy shrink to zero aside from some exceptional cases. On the other hand the anonymity sets of the Preference Propagation algorithm remain large with high probability.

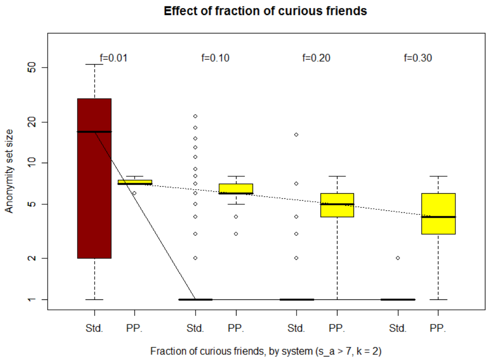

Fig. 2. The sizes of the anonymity sets remaining after a colluding coalition of friends tries to de-anonymize a preference. The fraction f represents the probability a friend is participating in the adversary coalition.

Their reduction is only due to the fraction of curious nodes actually being in the anonymity set of the propagated preference.

4.3 Quality Analysis

The second security objective of the proposed preference propagation algorithm is to limit the potential for the propagation of spam. Our objective is to limit the propagation of preferences from dishonest nodes mostly to the misguided nodes, but to make it difficult for such preferences to travel any further in the social graph.

The simple minded 1-hop propagation algorithm, in which users only broadcast their preferences to their neighbours, by definition achieves this property. Its down side is that the number of nodes that could benefit from the shared preference is limited to the number of friends. The simple extension of this scheme to a 2-hop broadcast extends the reach of the shared preferences but also makes it very likely that nodes are the recipients of some spam. Figure 3 (right) plots the number of nodes that are affected by users with different degrees in each mechanism. As expected the preference propagation algorithm affects a wider circle per node than the simple 1-hop propagation. At the same time the number of nodes included in a extended broadcast group is smaller than the reach of the 2-hop naive propagation.

Despite the order of magnitude increase in the nodes affected by the preference propagation algorithm compared with the 1-hop scheme, quality is to a large extent maintained, even for larger fractions of misguided nodes all connected to collaborating dishonest nodes. Figure 3 (left) illustrates the probability that a node receives spam for all systems, as the fraction of misguided nodes grows, in a real social network. In the 2-hop scheme receiving spam becomes quasi-certain even when a small minority of users are misguided ($f = 1\% - 10\%$). For

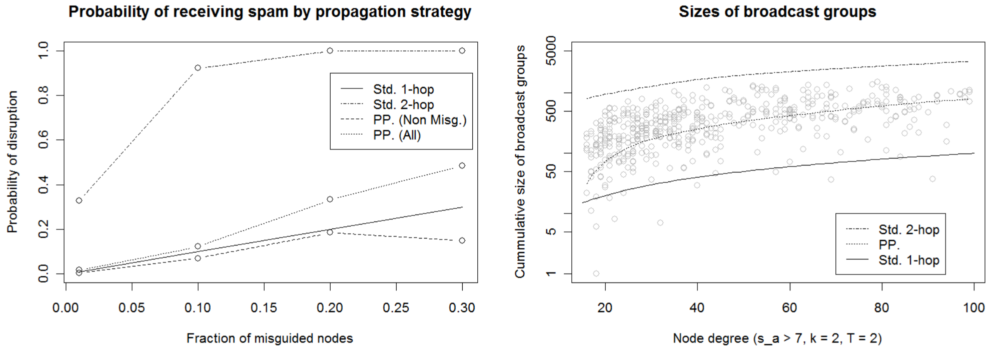

Fig. 3. The probability of honest nodes receiving malicious preferences, depending on the fraction of malicious nodes in the system (Left). The size of the naturally occurring broadcast groups as a function of node degree, compared with the 1-hop and 2-hop neighborhood (right).

the preference propagation scheme on the other hand the probability of receiving spam remains low even for large fractions of misguided nodes. It is in fact closely tracking the probability of being misguided for low rates of infiltration ($f = 1\% - 10\%$). For higher rates of infiltration ($f = 10\% - 30\%$) the probability non-misguided honest nodes receive spam increases slowly (marked at "PP. (Non Misg.)" on the illustration.)

There is a further fine, but important, difference between the proposed preference propagation algorithm and the traditional 1-hop or 2-hop schemes. In our approach the dishonest nodes, connected to the misguided honest nodes, must all be acting in a coordinated way to spam the system. They need to form cohesive subgroups between themselves and the users to broadcast their preferences. In effect it means that *a single adversary* must be connected to a fraction f of the honest nodes, unless they start applying social engineering to target related nodes to form cohesive subgroups.

The standard 1-hop and 2-hop propagation on the other hand does not require adversaries to coordinate in any way to spam. This means that the total fraction of misguided nodes, connecting to even unrelated adversaries, needs to be f for the probability of attack illustrated in figure 3 (left) to hold. It is much more likely that the total number of misguided nodes reaches a fraction f, than the number of misguided nodes connected to a single adversary's nodes reaches the same fraction. Unless there is a conspiracy at a massive scale it is difficult to imagine a single adversary connecting sybils to more than 10% of honest nodes in a larger network, at which point purpose built sybil attack defenses based on social networks should be employed [7].

Even in the absence of other sybil defenses the proposed system offers excellent guarantees against spam, as 10% of misguided nodes would lead to barely more than 10% of nodes being spammed (Figure 3 (left)). At the same time our strategy affords honest preferences a wide reach, of an order of magnitude above simply propagating preferences to friends (Figure 3 (right)).

4.4 Future work: adversarial profiling

As preferences are propagated in groups it might be possible for adversaries to modify established disclosure attacks [17, 8] to try to de-anonymize or profile users. For example if a user keeps receiving preferences about rare comic books, or another relatively rare subject, from many anonymity sets they might assume a single user is the originator and try to intersect the anonymity sets to de-anonymize them. The general attack considers users on one side, each with some abstract interests, and propagated preferences on the other side. Every time a preference is propagated to a broadcast group, this is modeled as a communication though a mix with the same anonymity set. Then the statistical disclosure attacks can be applied to extract user profiles in the long term.

The effectiveness of this attack in this new context is not clear, as the adversary has to ascribe preferences to categories – a fuzzy step that was not previously necessary. Subgroups are also likely to be coherent in their preferences which creates dependencies in the anonymity sets not previously considered by disclosure attacks. Adapting those traffic analysis techniques to extract preference profiles could be a valuable contribution to the literature. Bayesian models of such attacks are likely to be the most amenable to this setting [9].

5 Conclusions

We presented a general framework for anonymously sharing information in a social network. Our framework guarantees some k-anonymity, maintains high value by allowing information to be shared based on social proximity, and increases the cost of spamming the network. Our approach, extracting special cohesive social structures to protect users, adds to a body of work that uses social network information for security, as SybilInfer does for sybil defenses [7], and other proposals for automatically extracting privacy policies in social networks [5]. In the absence of a top-down trust structure we believe that the hints the users provide as to who they know and trust are the only way to bootstrap such policies, even though they might not be as bullet proof as traditional mandatory access control systems. Notions of differential privacy [14] can also be used to show that a published statistic leaks no identifiable information, and the application of this framework to our problem would be an interesting avenue for future work.

The framework we provide can be extended through alternative definitions of broadcast groups, that may provide a different anonymity, quality and spam-resistance trade-offs. Some structures could make use of explicit user hints of groups and communities, or even try to route preferences to groups that would most benefit from those (i.e., preferences about technical searches staying within technical communities). A further open question remains: how can traditional long term traffic analysis attacks be adapted, from inferring patterns of communications, to inferring users profiles despite the anonymization?

Acknowledgments. Dogan Kesdogan was the first to relate to the authors the idea that anonymity and privacy really comes down to "anonymity set engineering", which this work contributes to.

References

1. Alessandro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 36–58. Springer, 2006.
2. Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. Efficient k-anonymization using clustering techniques. In Kotagiri Ramamohanarao, P. Radha Krishna, Mukesh K. Mohania, and Ekawit Nantajeewarawat, editors, *DASFAA*, volume 4443 of *Lecture Notes in Computer Science*, pages 188–200. Springer, 2007.
3. Miguel Castro, Peter Druschel, Ayalvadi J. Ganesh, Antony I. T. Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *OSDI*, 2002.
4. Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. -anonymity. In Ting Yu and Sushil Jajodia, editors, *Secure Data Management in Decentralized Systems*, volume 33 of *Advances in Information Security*, pages 323–353. Springer, 2007.
5. George Danezis. Inferring privacy policies for social networking services. In Dirk Balfanz and Jessica Staddon, editors, *AISec*, pages 5–10. ACM, 2009.
6. George Danezis, Chris Lesniewski-Laas, M. Frans Kaashoek, and Ross J. Anderson. Sybil-resistant dht routing. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS*, volume 3679 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 2005.
7. George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*. The Internet Society, 2009.
8. George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In Jessica J. Fridrich, editor, *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2004.
9. George Danezis and Carmela Troncoso. Vida: How to use bayesian inference to de-anonymize persistent communications. In Ian Goldberg and Mikhail J. Atallah, editors, *Privacy Enhancing Technologies*, volume 5672 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2009.
10. Wendy Davis. Facebook hit with privacy complaint. The Online Media Post, June 2 2008.
11. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Dingledine and Syverson [12], pages 54–68.
12. Roger Dingledine and Paul F. Syverson, editors. *Privacy Enhancing Technologies, Second International Workshop, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers*, volume 2482 of *Lecture Notes in Computer Science*. Springer, 2003.
13. John R. Douceur. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *IPTPS*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.
14. Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Ding-Zhu Du, Zhenhua Duan, and Angsheng Li, editors, *TAMC*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.

15. Brynn M. Evans and Ed H. Chi. Towards a model of understanding social search. In *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 485–494, New York, NY, USA, 2008. ACM.
16. Brian Horling and Matthew Kulick. Personalized search for everyone. The Official Google Blog, December 4 2009.
17. Dogan Kesdogan, Dakshi Agrawal, Dang Vinh Pham, and Dieter Rautenbach. Fundamental limits on the anonymity provided by the mix technique. In *IEEE Symposium on Security and Privacy*, pages 86–99. IEEE Computer Society, 2006.
18. Emre Kiciman, Chun-Kai Wang, Shuo Chen, and Konstantin Mertsalov. U rank. Project web-page `http://research.microsoft.com/en-us/projects/urank/`, 2008.
19. R. Levien. Attack resistant trust metrics. *Draft available at http://www. levien. com/thesis/compact. pdf*, 2003.
20. Alan Mislove, Krishna P. Gummadi, , and Peter Druschel. Exploiting social networks for internet search. In *Proceedings of the 5th ACM Workshop on Hot Topics in Networks (HotNets)*, Irvine, CA, November 2006. ACM.
21. Judith S. Olson, Jonathan Grudin, and Eric Horvitz. A study of preferences for sharing and privacy. In Gerrit C. van der Veer and Carolyn Gale, editors, *CHI Extended Abstracts*, pages 1985–1988. ACM, 2005.
22. Luca Pacioli. Summa de arithmetica, geometrica, proportioni et proportionalita. Manuscript circulated in Venice, 1494.
23. Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, page 188. ACM Press, 1998.
24. J. Scott. Social network analysis. *Sociology*, 22(1):109, 1988.
25. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Dingledine and Syverson [12], pages 41–53.
26. Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
27. Chris Sherman. Yahoo bolsters personal search. Search Engine Watch Blog, April 26 2005.
28. Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *J. Cryptology*, 15(2):75–96, 2002.
29. Danny Sullivan. Eurekster launches personalized social search. Search Engine Watch Blog, January 21 2004.
30. Danny Sullivan. Google relaunches personal search - this time, it really is personal. Search Engine Watch Blog, June 28 2005.
31. S. Wasserman and K. Faust. *Social network analysis: Methods and applications.* Cambridge Univ Pr, 1994.
32. Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17. IEEE Computer Society, 2008.
33. Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In Luigi Rizzo, Thomas E. Anderson, and Nick McKeown, editors, *SIGCOMM*, pages 267–278. ACM, 2006.

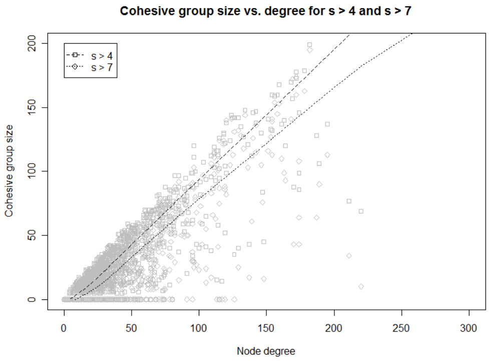

Fig. 4. Reach of cohesive groups per node degree and s_a value.

A Measuring *k*-plexes in the wild.

The privacy offered by our scheme against passive adversaries as well as dishonest nodes is closely related to the minimal size of the cohesive subgroups defining our anonymity sets, namely s_a. This parameter is not up to the designer of the system to tune, and is heavily dependant on the natural sizes of cohesive subgroups appearing within real-world social networks. Choosing s_a to be too large means that few nodes can broadcast their preferences, but choosing it to be too small results in lower degrees of anonymity for preferences.

To better understand the range of possible subgroup sizes s_a we measure the number of nodes reachable through a k-plex with parameters $k = 2$, $s_a > 4$ and $s_a > 7$. We use the Live Journal (LJ) data set[4], where edges represent the mutual consent of two LJ users to read each others' private journal entries. Figure 4 illustrates the number of users reachable for these two parameters. It is clear that the number of nodes sharing cohesive subgroups with a user grows roughly linearly with the degree of the node. As expected the a higher s_a leads to fewer nodes being in cohesive subgroups of that size. We use $s_a > 7$ throughout all our experiments, since it seems to offer a good trade-off between privacy and reachability.

The natural emergence of social structures that are large and cohesive could be of great importance for other security designs. Traditional threshold cryptosystems, or secret sharing schemes, assume that their processes are distributed across a number of participants out of whom some are honest. Yet there has been little research in measuring the natural sizes of subgroups in a social network over which such functions could be distributed. Our work is the first to inform the debate with such figures.

[4] `http://www.livejournal.com/`