The Walls Have Ears: Optimize Sharing for Visibility and Privacy in Online Social Networks

Thang N. Dinh*, Yilin Shen*, and My T. Thai
Department of Computer and Information Science and Engineering
University of Florida, Gainesville, Florida, 32611
{tdinh, yshen, mythai}@cise.ufl.edu
*The first two authors T. N. Dinh and Y. Shen contributed equally to this work.

ABSTRACT

With a rapid expansion of online social networks (OSNs), millions of users are tweeting and sharing their personal status daily without being aware of where that information eventually travels to. Likewise, with a huge magnitude of data available on OSNs, it poses a substantial challenge to track how a piece of information leaks to specific targets. In this paper, we study the problem of smartly sharing information to control the propagation of sensitive information in OSNs.

In particular, we formulate and investigate the *Maximum Circle of Trust* problem of which we seek to construct a circle of trust on the fly so that OSN users can safely share their information knowing that it will not be propagated to their unwanted targets (whom they are not willing to share with). Since most of messages in OSNs are propagated within 2 to 5 hops, we first investigate this problem under 2-hop information propagation by showing the hardness of obtaining an optimal solution, along with an algorithm with proven performance guarantee. In a general case where information can be propagated more than two hops, the problem is #P-hard i.e. the problem cannot be solved in a polynomial time. Thus we propose a novel greedy algorithm, hybridizing the handy but costly sampling method with a novel cut-based estimation. The quality of the hybrid algorithm is comparable to that of the sampling method while taking only a tiny fraction of the time. We have validated the effectiveness of our solutions in many real-world traces. Such an extensive experiment also highlights several important observations on information leakage which help to sharpen the security of OSNs in the future.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms, Performance, Theory

Keywords

Social networks, circle of trust, algorithms, complexity

1. INTRODUCTION

Recent rapid development of Online Social Networks (OSNs) has revolutionized the way of human interaction and drastically changed the landscape of communications and information sharing in the cyberspace nowadays [9, 25]. One of the most important characteristics of OSNs is the "word-of-mouth" exchanges, in which information can be propagated from friends to friends of friends and eventually widely spread over the network. By leveraging this power, many organizations and companies have been using OSNs as an effective medium to increase their visibility and advertise their products [1]. Likewise, millions of OSN users are sharing their personal status daily with a hope to keep many of their friends, near or far, updated. Unfortunately, the fast information propagation is a double-edged sword, that said, it not only quickly propagates information to our friends but at the same time, it also spreads the information to many unwanted targets whom we do not want to share with.

Let us consider the following simple example which highlights a basic need for any organizations or companies who use OSNs. Suppose that Bob wants to share with his friends some of his personal pictures and stories in Facebook, yet he is reluctant to let Chuck know about them. Being careful, Bob just shares to the list of his friends in Facebook where Chuck is not in that group with a belief that Chuck cannot see those pictures. Unfortunately, Alice, who is a friend of both Bob and Chuck, replied to the post, and thus Chuck will see the message from Alice and learn about Bob's sharing. Assume that Bob is extra careful by using the Custom Privacy function provided by Facebook to hides his sharing from Chuck. Unfortunately, this function only tracks and hide the message based on the message-ID, not on its propagation. Therefore, when Bob's friend Alice posts a new message *mentioning* Bob's pictures and stories, this new message cannot be hidden from Chuck anymore since its ID is no longer the same as the original message from Bob. Consequently, Chuck will still learn about Bob's pictures and stories. Thus it raises a practical question: Is there any mechanism for Bob to share his pictures and stories to as many friends as possible without reaching to Chuck?

What Bob really needs is that right before he is ready to share his stories, he should have an opportunity to construct on the fly a subset of his friends to share these stories with so that the probability of Chuck knowing them is very small (for example see Fig. 1). We refer to this subset as a

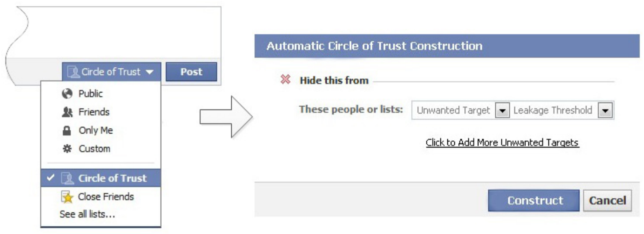


Figure 1: Constructing the Circle of Trust

circle of trust. Of course, we cannot filter out most of his friends because one of the main purposes of posting message on OSNs is to share the information with as many friends as possible. Therefore, we formulate a new optimization problem, called *Maximum Circle of Trust* (MCT), to construct a circle of trust for Bob so that once Bob posts a message to this CT, the probability of such friends in this CT spreading the message to *unwanted targets* is bounded under some input thresholds and the *expected visibility* of the message is maximized. Here the message is *visible* to a friend of Bob if the message appears on the wall of the Bob's friend.

The realization of this function in OSNs, unfortunately, requires us to study many fundamental problems in the large-scale networks. With a huge magnitude of OSN users and data available on OSNs, it poses a substantial challenge to understand how information reach to specific targets. In addition, it requires a significant work in data sampling to verify whether a message can be leaked from one user to another one in the network. Unfortunately, the time complexity of the sampling method is high, therefore, it does not suitable for an on-the-fly construction.

In this paper, we develop solutions to the MCT problem and address the above two fundamental problems. More specifically, our contributions are summarized as follows:

- We are the first to study the maximum circle of trust problem tackling the information leakage so that OSN users can conveniently and safely share their information in OSNs knowing that these information will not be propagated to their unwanted targets.
- In order to understand how information leak to specific targets, we develop a Sharing-Mentioning Leakage (SML) model presenting different leaking features in distinct OSNs.
- Since information mostly can be propagated within 2 to 5 hops [6], we first investigate this problem under 2-hop information propagation, called 2-MCT. We show how hard to obtain a near optimal solution to 2-MCT by proving the inapproximability and provide a randomized algorithm with an asymptotic tight theoretical guarantee. This randomized algorithm can be further derandomized to obtain a deterministic solution.
- In a general case where information can be propagated more than two hops, the problem is #P-hard, ruling out the existence of the finite ratio approximation algorithms. One of the major problems to develop an effective heuristic is to validate solutions' feasibility which is typically done with the expensive sampling method. Thus we propose a novel cut-based estimation which has a comparable result with the sampling method but tremendously improves the running time. This new cut-based estimation can help to address several related problems which requires sampling. Based on the new estimation, we devise an effective greedy algorithm for the general MCT problem.
- The performance of our proposed solutions are validated on many real-world traces, including Facebook, Twitter, and Foursquare. The experiments also reveal some important observations on information leakage and how to better control the information propagation in OSNs.

The rest of this paper is organized as follows. In Section 2, we introduce a SML propagation model and the formal definition of the MCT problem. Section 3 includes the complexity results and our randomized algorithm that comes with a performance guarantee for the 2-MCT problem. For general MCT problem, the IGC approach are provided in Section 4. The experimental evaluation is illustrated in Section 5 and related work is presented in Section 6. Finally, Section 7 provides some concluding remarks.

2. MODEL AND PROBLEM DEFINITION

In order to solve the MCT problem, we first introduce in this section a novel information leakage model, namely *Sharing-Mentioning Leakage* (SML) model. The model is based on the well-known probabilistic model *Independent Cascade* (IC) [11, 16, 17, 7].

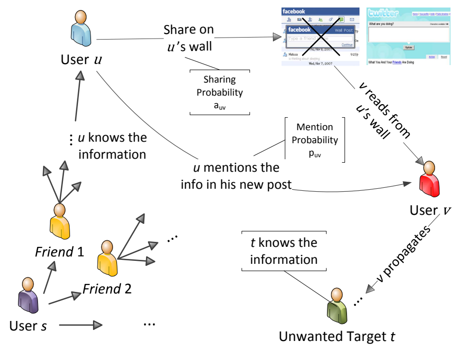


Figure 2: SML Leakage Model

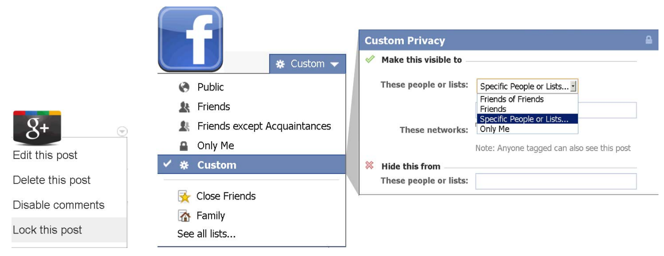
2.1 SML Leakage Model

The social network is modeled as a directed graph $G = (V, E)$, where vertices in V represent users and edges in E represent social links among the users. In the IC model [16], each edge (u, v) is associated with a real number $w_{uv} \in [0, 1]$ that indicates how likely user u would influence user v. When u becomes active, it has a single chance to activate v with the probability of success w_{uv}.

In the SML model, we consider two different information propagation (leakage) mechanisms, namely sharing and mentioning as follows.

- *Sharing*: u shares the message using functions provided by OSNs. For example, u can use "retweet" or "reply" to share on Twitter and "share", "comment" or "like" to share on Facebook. In this case, the message will appear on u's own wall and be visible to v;
- *Mentioning*: u can also propagate the information to v by retyping using his own words or mentioning it in a post.

Correspondingly, each edge (u, v) is associated with two probabilities: the *sharing probability* a_{uv} and the *mentioning*



(a) Google+ (b) Facebook

Figure 3: Sharing options in OSNs

probability p_{uv} that indicate how likely u will make the information visible to v via sharing and/or mentioning. Note that while the sharing propagation can easily be blocked at will, e.g. by using privacy options in Facebook or building manual sharing circles in Google+, it is much harder to block the message propagation via mentioning as there are many different ways to convey a same message.

We now describe the leakage propagation process in G which is illustrated in Fig. 2. Assume that a source user $s \in V$ shares a message m to a subset $C = \{v_1, v_2, \ldots, v_l\}$ of his/her friends. In the first round, a user $v \in C$ will learn about the message m with the sharing probability a_{sv} i.e. there is no propagation via mentioning in the first round. In the second round, if v learn about m in the first round, then each friend w of v can learn about message m from v via sharing with probability a_{vw} or via mentioning with probability p_{vw}. The propagation continues up to δ rounds, where $\delta > 0$ is a predefined integral constant in the model. Our SML model assumes the independence propagation between different links as in the IC model.

Leakage Path. A leakage path $(s = v_0, v_1, \ldots, v_l = t)$ is a path between the source s and an unwanted target t, in which $(v_i, v_{i+1}) \in E$ and v_{i+1} learns about the message m from v_i for all $i = 0, 1, \ldots, l-1$. If v_{i+1} learns about the message via sharing, then (v_i, v_{i+1}) is called a sharing edge, otherwise (v_i, v_{i+1}) is a mentioning edge. The first edge on a leakage path is always a sharing edge. In addition, at least one edge on the path must be a mentioning edge as we shall discuss next.

Sharing Options in OSNs. We present the privacy settings that help users control who see the message in OSNs and the constraint on the leakage paths in those networks.

Facebook. Facebook provides options to share a post to a specific group of friends. Furthermore, it provides the 'hide this from' feature which hides posts from certain users (see Fig. 3b). Regardless of how the friends of the source and the unwanted targets reshare the message, Facebook keeps track of the original message and make it invisible from the unwanted targets. In other words, the unwanted targets cannot learn about the message through a path with all sharing edges. Thus every leakage path must have at least one mentioning edge.

Google+. This social networks allows users to share the information to a circle of friends. Each circle is a *predefined* group of friends which can be overlapping and hierarchical. However, the 'hide this from' option in Facebook is not available. Instead, Google+ provides the 'lock this post' option which prohibits resharing the message (see Fig. 3b). If the message is locked, mentioning is the only way to propagate the information further. Hence, the *second* edge in every leakage path in Google+ must be a mentioning edge.

Twitter. Twitter does not allow to hide messages (tweets) from some users. There are only two options to hide the tweets: either deleting the tweet permanently or hide all of the tweets at once by setting the profile to private. However, if Twitter allows users to share tweets to specific followers in future, then the circle of trust concept can be applied to give Twitter's users more control over sharing their tweets.

Leakage Propagation model. We now provide some intuitive explanation and justification of our model. The SML model is defined based on the Independent Cascade (IC) model for influence propagation [16]. However, the SML model is different: 1) the propagation happens in two different channels: sharing and mentioning 2) the IC model is for influence propagation starting from a seed set, while our model is for the message originated from a single user. Note that other diffusion models e.g. the *Linear threshold model* [16, 17] can also be used to characterize the information propagation in the network.

For our study of the circle of trust concept, we consider that the social network with sharing and mentioning probabilities is given. A number of researches provide methods in extracting the social network and influence probabilities [13, 3, 23, 12]. Those methods can be applied to extract the sharing and mentioning probabilities. In addition, the EdgeRank values which indicate how likely two users interact with each other in Facebook [8] can be used to infer those probabilities. However learning those sharing-mentioning probabilities as well as the combination of the sharing-mentioning mechanisms with other information propagation models are not the focus of this work. Instead, this paper focuses on finding solutions for the MCT problem. The solutions for MCT, proposed in this paper, are relatively independent with the leakage propagation model. That is they can be easily modified to work with other leakage propagation models.

2.2 Problem Definition

In an OSN, when a user s posts some message m, his aim is usually to share it with as many friends as possible, i.e., to maximize the visibility of his message to his friends while preventing it from reaching to some *unwanted targets.* Considering the spread of information within δ hops from the source, we study the following *δ-Hop-Propagation Maximum Circle Of Trust* (δ-MCT) problem, which constructs a circle of trust, a subset of the source's friends. The goal is to *maximize the expected visibility of the message* and to *restrict the leakage probability of each unwanted target to a certain degree.*

PROBLEM 1 (δ-MCT PROBLEM). *Given a directed graph $G = (V, E)$ with $|V|$ users and $|E|$ edges underlying an OSN, where each edge (u, v) is associated with a tuple of sharing probability and mention probability $\langle a_{uv}, p_{uv}\rangle$. Let s be the source user with $|N(s) \setminus T| = S_n$ neighbors, $T = \{t_1, \ldots, t_k\}$ be the set of $k = |T|$ unwanted targets, and k real numbers $0 \leq \tau_1, \ldots, \tau_k < 1$ be the leakage thresholds. The δ-MCT problem constructs a circle of trust (CT) $C \subseteq N(s)$, with the maximum expected visibility, measured as $\sum_{u \in C} a_{su}$, such that the probability of each unwanted target t_j learns about m, after at most δ hops propagation, is at most τ_j.*

Note that in our model, the source user s will neither share nor mention the message to his unwanted targets in case the unwanted targets are the source's friends.

3. ALGORITHM FOR 2-MCT PROBLEM

We study the special case when only vertices within two-hop from the source are considered. We believe this special case covers most of real-world scenarios for two reasons. First, the source must be aware of the unwanted targets, thus, the unwanted targets are often within a close neighborhood of the source. Second, acquiring more than two-hop neighbors of the source might be computationally expensive to implement in distributed applications in OSNs.

The 2-MCT problem can be shown to be NP-hard by a reduction from the subset sum problem. Thus, we are left with two choices: designing heuristics which can have arbitrary bad worst-case performance or designing approximation algorithms which can guarantee the produced solutions are within a certain factor from the optimal. Formally, a ρ-approximation algorithm for a maximization (minimization) problem always returns solutions that are at most ρ times smaller (larger) than an optimal solution.

In this section, we first formulate the *Integer Linear Programming* (ILP) for MCT problem. Then, we present an approximation algorithm based on rounding the fractional solution of a relaxation of the ILP. We end the section with the proof on the hardness of approximation that shows the tightness of our obtained approximation factor.

3.1 Mathematical Programming Approach

We define an indicator variable x_u for each friend $u \in N(s)$ of s where $x_u = 1$ if u is included to CT, and $x_u = 0$ otherwise. Our objective is to maximize the expected visibility of the message, i.e. the expected number of s' friends that the message is visible to. Thus, it can be written as the sum of sharing probabilities of s's friends except unwanted targets, that is, $\sum_{u \in N(s)\setminus T} a_{su} x_u$.

```
   Input  : 2-MCT instance
   Output: Friends in CT $C$
1     Solve the LP relaxation to get the fractional solution $x$
2     Set $\alpha \leftarrow 4k, C \leftarrow \emptyset$
3     Select $u$ into $C$ with probability $\frac{x_u}{\alpha}$, $\forall u \in N(s) \setminus T$
4     $C' \leftarrow C$
5     for each $j = 1$ to $k$
6         while (the $j^{th}$ constraint is violated)
7             $v \leftarrow \arg\min_{x_u=1} w_{uj}$
8             Remove $v$ from $C'$ i.e. set $x_v \leftarrow 0$
9         end while
10    return $C'$
```

Algorithm 1: Randomized Rounding for MCT

For an unwanted target t_j, the message will be leaked to t_j iff a neighbor u of s is informed with probability a_{su} and u further leaks to t_j via mentioning with probability p_{ut_j}. Therefore, all leakage paths of length two will consist of one sharing edge following by one mentioning edge, which is true for both Facebook and Google+. Therefore, the constraint w.r.t. each unwanted target t_j can be written as

$$1 - \prod_{u \in N(s)\setminus T} (1 - a_{su} p_{ut_j})^{x_u} \le \tau_j.$$

Rearrange and take the logarithm of both sides, we obtain the following Integer Linear Programming (ILP):

$$\begin{array}{lll} \text{maximize} & \sum_{u \in N(s)\setminus T} a_{su} x_u & \\ \text{subject to} & \sum_{u \in N(s)\setminus T} w_{uj} x_u \le c_j, & \forall j = 1..k \\ & x_u \in \{0,1\} & u \in N(s) \setminus T, \end{array} \quad (1)$$

where $w_{uj} = -\log(1 - a_{su} p_{ut_j})$ and $c_j = -\log(1 - \tau_j)$.

3.2 Approximation Algorithm

We first solve the LP relaxation of (1) obtained by replacing the integral condition on x_u with the condition $x_u \in [0,1], \forall u$. Let x be an optimal fractional solution of the LP relaxation and $\mathsf{OPT}_{\mathsf{LP}} = \sum a_{su} x_u$ be its objective. Our randomized rounding algorithm, as shown in Algorithm 1, consists of two phases. In the first phase, we independently select vertex u into a CT C with probability $\frac{x_u}{4k}$ for each $u \in N(s) \setminus T$. In the second phase, for each violated constraint j where $\sum_{u \in C} w_{uj} x_u > c_j$, we remove v with the minimum w_{vj}, i.e. set $x_v = 0$, and repeat until the constraint is satisfied. Finally, we return the remaining vertices in the CT denoted by C'.

We can verify that C' is valid CT with probability one. In the rest, we bound the ratio between the visibility of our constructed CT and that of an optimal CT. Since

$$\mathbb{E}\left[u \in C'\right] = \mathbb{E}\left[u \in C\right] \mathbb{E}\left[u \in C' \middle| u \in C\right],$$

the expected visibility of C' will be

$$\begin{aligned} \mathbb{E}\left[\sum_{u \in C'} a_{su}\right] &= \sum_{u \in N(s)\setminus T} \mathsf{Pr}\left[u \in C\right] \mathsf{Pr}\left[u \in C' \middle| u \in C\right] a_{su} \\ &= \sum_{u \in N(s)\setminus T} \mathsf{Pr}\left[u \in C' \middle| u \in C\right] \frac{x_u}{4k} a_{su}. \end{aligned} \quad (2)$$

We present the key lemma for the approximation factor.

Lemma 1. *For any vertex $u \in N(s) \setminus T$, the probability* $\mathsf{Pr}\left[u \in C' \middle| u \in C\right] \ge \frac{1}{2}$.

Proof. Let R_{uj} be the event that vertex u is removed from C'. Consider the set C' at the time u is removed. Since the vertices are removed in non-decreasing order of w_{uj}, all the remaining vertices $u' \in C'$ have $w_{u'j} \ge w_{uj}$. Let $v \in C'$ be the one with the maximum w_{vj}.

Consider two possible cases $w_{vj} > c_j/2$ and $w_{vj} \le c_j/2$. If $w_{vj} > c_j/2$. Then, $\mathsf{Pr}[R_{uj}]$ is no larger than the probability of having a vertex $v \in C'$ with $w_{vj} > c_j/2$. Denote that probability by α_j. We have

$$\alpha_j \le \sum_{w_{vj} > c_j/2} \mathsf{Pr}[v \in C] \le \sum_{w_{vj} > c_j/2} \frac{x_v}{4k} \frac{2 w_{uj}}{c_j} \le \frac{1}{2k}.$$

If $w_{vj} \le c_j/2$, then $\forall u \in C$, $w_{uj} \le c_j/2$. Since constraint j is violated, we have $\sum_{u' \in C, u' \ne u} w_{u'j} > c_j - c_j/2$. Then by the following well-known Markov's inequality,

Markov's inequality *[2]. If X is any random variable and $a > 0$, then* $\Pr(|X| \ge a) \le \frac{E(|X|)}{a}$,

we can upper-bound this probability by $\frac{1}{2k}$. Thus, in both cases, $\mathsf{Pr}[R_{uj}] \le \frac{1}{2k}$. Then by the union bound, we obtain

$$\mathsf{Pr}\left[u \in C' \middle| u \in C\right] \ge 1 - \sum_{j=1}^{k} \mathsf{Pr}[R_{uj}] = \frac{1}{2}$$

This yields the lemma. □

THEOREM 1. *There is an $O(k)$ randomized approximation algorithm for 2-MCT.*

PROOF. From Lemma 1 and Eq. 2, the expected visibility of C' will be at least

$$\sum_{u \in N(s)\setminus T} \frac{1}{2}\frac{x_u}{4k} a_{su} = \frac{1}{8k}\mathsf{OPT}_{LP}.$$

Since OPT_{LP} is an upper-bound on the optimal solution of MCT, Algorithm 1 is an $8k$ approximation algorithm. □

In our experiments, we repeat the rounding algorithm (Algorithm 1) $\log 4k$ times with $\alpha = 1, 2, \ldots, 2^{\log 4k}$; then the best solution is selected.

3.3 Inapproximability Results

To investigate the tightness of the above theoretical bound for 2-MCT, we prove its inapproximability which illustrates that the theoretical guarantee is asymptotic tight in terms of the number of the unwanted targets k.

THEOREM 2. *For any $\epsilon > 0$, 2-MCT is hard to be approximated within a factor*

- $k^{1/4-\epsilon}$*, unless P = NP, and*
- $k^{1/2-\epsilon}$*, unless NP=ZPP.*

PROOF. We show a gap-preserving reduction [4] from the Max-Clique problem, finding a clique with the largest number of vertices, to 2-MCT. The hardness of approximation of 2-MCT is then derived from the following inapproximability results of Max-Clique by Hastad [14]: for any $\epsilon > 0$, there is no polynomial time algorithm that approximates Max-Clique within a factor

- $n^{1/2-\epsilon}$, unless NP=P, and
- $n^{1-\epsilon}$, unless NP=ZPP.

We begin with the gap-preserving reduction. Given an instance $G = (V, E)$ of Max-Clique with $n = |V|$ vertices $v_1, \ldots, v_n$, we construct an instance $G' = (V', E')$ of the 2-MCT problem. The vertices set V' includes a source node s, n vertices $v'_1, \ldots, v'_n$, and a set T of $k = \binom{n}{2} - |E|$ unwanted targets (vertices). The set of edges E' includes n edges from s to all vertices $v'_1, \ldots, v'_n$, and edges from $v'_1, \ldots, v'_n$ to T, constructed as follows. Associate each pair $(v_i, v_j) \notin E$ with a vertex $t_{ij} \in T$ and include both edges (v'_i, t_{ij}) and (v'_j, t_{ij}) into E'. Furthermore, all edges have sharing probabilities *one* and mentioning probabilities 1/2. All unwanted targets t_{ij} have the same leakage threshold $\tau = 1/2$.

The key property of the above construction is that for a pair $(v_i, v_j) \notin E$, the leakage probability at t_{ij} will be less than or equal to the leakage threshold 1/2, iff either v'_i or v'_j (or none of them) is selected into the CT of G'. Thus we can verify that there is a *one-to-one* mapping between the cliques in G to the CT of the same sizes in G'.

We use the fact $k < n^2$ to prove by contradiction that unless $P = NP$, 2-MCT is hard to approximate within a factor $k^{1/4-\epsilon}$. Assume that we have a polynomial time algorithm to approximate the 2-MCT problem within a factor $k^{1/4-\epsilon}$. Then for each instance $G = (V, E)$ of Max-Clique, we can use that approximation algorithm to find a solution of 2-MCT in $G' = (V', E')$ with visibility at least $1/k^{1/4-\epsilon}\mathrm{OPT}_{2-\mathrm{MCT}}$, where $\mathrm{OPT}_{2-\mathrm{MCT}}$ denotes the visibility of an optimal solution of the 2-MCT instance. The solution of the 2-MCT instance induces a clique of the same size in G. Since $k < n^2$ and the size of an optimal solution of Max-Clique is equal to that of 2-MCT, the induced clique has size at most $1/n^{1/2-\epsilon/2}$ time the size of a maximum clique. We can set $\epsilon' = \epsilon/2$ and obtain a contradiction to the Hastad's results [14]. Similarly, we can show that 2-MCT is hard to approximate within a factor $k^{1/2-\epsilon}$ for any $\epsilon > 0$, unless $NP = ZPP$. □

4. GENERAL δ-MCT PROBLEM

The δ-MCT problem becomes much harder when $\delta \geq 3$. In fact, we prove that the problem is $\#P$-complete [22] when $\delta \geq 3$ denying the existence of efficient algorithms to solve the problem. To solve the problem in different settings, we provide a family of greedy algorithms for δ-MCT.

For simplicity, we assume the sharing options of Facebook are in effect. That is all leakage paths must have at least one mentioning edge. The other social networks' sharing models can be tackled in a similar way.

```
   Input  : $\delta$-MCT instance
   Output: Friends in CT $C$
 1 $C \leftarrow \emptyset$;
 2 $L \leftarrow N(s) \setminus T$;
   // Remove all unwanted targets with no risk of leakage
 3 foreach $t_j \in T$ do
 4     if $\tau_j(L) < \tau_i$ then
 5         $T \leftarrow T \setminus \{t_j\}$;
 6     end
 7 end
 8 while $\forall \ell_j(C) < \tau_j$ do
       // Update the set of candidate users $L$
 9     foreach $v \in L$ do
10         if $\exists j : \ell_j(C + \{v\}) > \tau_j$ then
11             $L \leftarrow L \setminus \{v\}$;
12         end
13     end
14     Find $v \in L$ that maximizes $f(v)$;
15     $C \leftarrow C \cup \{v\}$;
16 end
17 return $C$
```

Algorithm 2: IGC Algorithm

4.1 Iterative Greedy Algorithm

We propose an effective algorithm to construct CT, namely *Iterative Greedy Construction* (IGC). In fact, the algorithm is a meta-algorithm, which can be combined with different leakage estimation methods. First, we introduce this meta-algorithm IGC. Then, we present the combinations of a sampling algorithm, a Non-sampling algorithm to estimate leakage with IGC and their hybrid in the sequential subsections.

The IGC algorithm, as shown in Algorithm 2, iteratively adds one of the source's neighbors into the circle of trust until no more nodes can be added without causing the leakage probabilities to exceed the thresholds. Specifically, in each iteration, we first update the set of candidate neighbors L, those whose addition to CT still guarantees that the leakage levels at each unwanted targets t_j does not exceed the threshold τ_j. Denote by $\ell_j(C)$ the probability that the message will be leaked to t_j with respect to CT C. Then, v is a candidate if $\ell_j(C + \{v\}) \leq \tau_j \ \forall j = 1..k$.

We use a greedy function $f(v)$ to evaluate the fitness of user v. The function is defined as follows

$$f(v) = \frac{a_{sv}}{\max_{t_j \in T} \frac{\ell_j^{(v)}\left(\sum_{u\in L} \ell_j^u - \ell_j^{(v)}\right)}{1-\ell_j^{(v)}}} \quad (3)$$

where $\ell_j^{(v)} = \ell_j(C+\{v\})/\tau_j$ is the normalized leakage level at t_j after adding v to the CT. The numerator of $f(v)$ is selected to be a_{sv} so that the algorithm will favor close friends of s, whose adding increase the visibility.

In addition, the proposed greedy function takes into the account the following quantities

- $1 - \ell_j^{(v)}$: the remaining leakage tolerance at the unwanted target t_j;
- $\sum_{u \in L} \ell_j^u - \ell_j^{(v)}$ reflects the future potential leakage to target t_j when adding user v to CT.

By maximizing $f(v)$ the algorithm will prefer user v with small denominator. Thus, the algorithm will select the users that have higher remaining leakage tolerances but lower future potential leakage.

Note that in IGC, we assume the existence of a procedure to estimate leakage $\tau_j(C)$, which can be invoked up to $O(S_n^2|T|)$, where $S_n = |N(s) \setminus T|$. Thus, the procedure to estimate leakage is the critical component of IGC. It decides both the solution quality and the running time of the meta-algorithm. Due to the #P-hardness of the estimating leakage, exact solutions require exponential time. Hence, we have to resort to approximation methods to estimate $\ell_j(C)$.

4.1.1 Bi-state Sampling Leakage Estimation

The most natural method to estimate $\ell_j(C)$ is *Monte Carlo Sampling*, in which the accuracy depends on the number of sampling times. The method stimulates the random process following the SML propagation model. Each edge $(u, v) \in E$ will be generated independently and assigned label 'sharing' with probability a_{uv} or generated and assigned label 'mentioning' with probability p_{uv}. The estimation of $\ell_j(C)$ is calculated as the ratio between the number of time the message reaches t_j via a *leakage path* to the number of sampling times n_S, which is chosen as 20,000 in our experiments. Recall that in a leakage path, there must be at least one mentioning edge; in addition the path must have length at most δ. Thus after the sampled graph is generated, we use a modification of the breadth-first search algorithm to find such leakage paths from the source to each unwanted target.

Note that in the sampling process, each edge is either sharing or mentioning edge. This implies that each user in the network forwards the message by either sharing or mentioning but not both. Alternatively, we can allow both sharing and mentioning edges between a pair of users. In our experiments, we follow the first approach.

To speed up the estimation, we can estimate the leakage at all targets at the same time using a single (modified) bread-first search. Then the IGC algorithm, in the worst case, runs in time $O(n_S S_n^2(m + n))$.

4.1.2 Non-sampling Leakage Estimation

Sampling methods are rather expensive for large-scale networks, thus, alternative leakage estimation methods are desired to handle the estimate the leakage $\ell_j(C)$. Here, we provide a fast method to calculate a non-trivial upper-bounds on the leakage $\ell_j(C)$ at unwanted target t_j. The upper bound is based on identifying minimal pseudo-cutset, as defined later, between the source and the unwanted target. In the rest of the section, we assume that a trusted circle C is already given without further mention; and each edge (u, v) is associated with a combined sharing-mentioning propagation probability $q(u, v)$.

We first define some necessary notations. Let $t \in T$ be an arbitrary unwanted target that we wish to estimate the leakage level $\tau(t)$, defined as the probability of having a path between s and t. For each edge (u, v), define event $X_{u,v} = 1$ if the information can propagate through the edge (u, v) and $X_{u,v} = 0$ otherwise i.e. $\Pr[X_{u,v} = 1] = q(u, v)$. We define a cutset $\langle S, T\rangle$ as $\langle S, T\rangle = \{(u, v) \in E \mid u \in S, v \in T\}$. For a subset of edges $A \subseteq E$, let A^* be the event that all the edges of A do not let the information propagate through i.e. $\Pr[A^*] = \Pi_{e \in A}(1 - q_e)$. Then, the exact leakage $\tau(t)$ can be computed as follow

$$1 - \Pr[\bigcup_{\langle S,T\rangle \in \mathcal{C}_{s,t}} C^*],$$

where $\mathcal{C}_{s,t}$ is the set of all cutsets $\langle S, T\rangle$ satisfied $s \in S$ and $t \in T$. However, computing this exact formulation is intractable. We first relax the formulation by replacing $\mathcal{C}_{s,t}$ with a collection of cutsets $\mathcal{B} \subset \mathcal{C}_{s,t}$. This yields the following upper-bound on the leakage $\tau(t)$

$$1 - \Pr[\bigcup_{\langle S,T\rangle \in \mathcal{B}} C^*] \geq \tau(t)$$

Note that when $\mathcal{B}$ contains only disjoint cutsets, the upper bound can be efficiently computed in polynomial time. Thus, in our non-sampling estimation method, we seek a maximal collections of disjoint cutsets. The selection of good disjoint cutsets that gives close estimation is governed by two factors. First, we prefer small cutsets; at the same time, we prefer a collection of many cutsets. Note that the number of cutsets is bounded by the distance between the source and the unwanted target. Since if P is a s, t path of minimum length l, every cutset must contains at least one edge in P. Thus, we have the following theorem.

THEOREM 3. *The maximum number of disjoint cutsets that separate s and t cuts equals the minimum length of a path between s and t.*

Instead of using cutsets, we propose the use of *pseudo-cutsets*, which are the set of edges whose removal makes the distance from the source s to destination t at least $\delta + 1$ hops. The reason is that disrupting all paths of length at most δ is sufficient to prohibit the message m to propagate to t. Our algorithm to find the pseudo-cutsets and compute the upper-bound is given in Algorithm 3.

```
  Input  : Graph $G = (V, E)$, $s, t \in V$, and probabilities $q(u, v)$
           for $(u, v) \in E$.
  Output: Bound on the leakage probability at $t$.
1 foreach $v \in V$ do
2     Compute $d(s, v)$ and $d(v, t)$ the hop distance from $s$ to $v$
      and from $v$ to $t$, respectively;
3 end
4 Let $d_0 = d(s, t)$;
5 for $i = 1$ to $d_0$ do do
6     $C_i \leftarrow \{(u, v) \mid d_s(u) = i \wedge d_s(u) + d_t(v) \leq \delta - 1\}$
7 end
8 return $1 - \Pi_{i=1}^{\delta}\left(1 - \Pi_{e \in C_i}(1 - q_e)\right)$
```

Algorithm 3: Disjoint Cutset Upper-bound

The algorithm first applies Breadth-First Search algorithm to construct layers of vertices. Layer L_i consists of vertices at distance i from the source in term of hops. The cutset C_i is constructed by including all edges (u, v) with $u \in L_i$ and $v \in L_{i+1}$ but only if $d_s(u) + d_t(v) \leq \delta - 1$ i.e. there is a path of length at most δ going through the edge (u, v). The upper-bound can be computed efficiently, since it only requires visiting nodes within δ-hop from the source instead of doing so n_S times as in the sampling method.

Although the Disjoint Cutset Upper-bound is not as accurate as the sampling method, provided a large number of sampling times, it is the only scalable method to estimate the leakage. Moreover, it is also rather effective in comparing the neighbors to find out the ones causing the leaks to the unwanted targets. We name the use of the disjoint cutset upper-bound to estimate the leakage in the IGC algorithm as the *Non-sampling Method*.

4.2 Hybrid Method

Since the Non-sampling method can overestimate the leakage at the unwanted targets, it often stops early even more neighbors can still be added to CT without causing excessive leakage risks. Thus, the solution of the IGC algorithm using the upper-bound via disjoint cutsets can be further fine tuned.

Our hybrid method combines the Sampling method and Non-sampling method. The hybrid method consists of two phases. In the first phase, the Non-sampling method is used to quickly construct a CT. In the second phase, the following simple heuristics is used together with the sampling algorithm to add more neighbors to the circle:

1. Sort the neighbors that are not included in the circle in non-decreasing order of their visibilities;
2. In that order, include each neighbor into CT and check if the leakage below the thresholds using *Sampling algorithm*. If so, add the node to CT.

The hybrid method achieves both the speed of the Non-sampling method and the advantage of the Sampling method, the high solution quality. Since, it can involve at most S_n calls to the sampling estimation algorithm, the running time of the hybrid method is very competitive for large networks.

We observe in our experiments that often only one to three neighbors are added using this heuristic. Thus, the first phase using the Non-sampling method is rather effective in term of detecting trusted friends. Overall, this hybrid approach turns out to be excellent performance in our experiments and gives a great trade-off between the time and the solution quality.

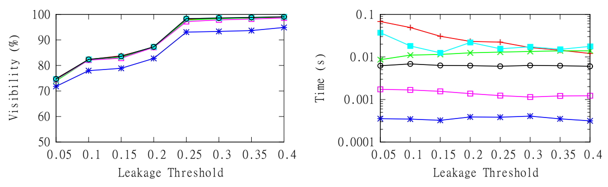
5. EXPERIMENTAL STUDY

In this section, we perform the experiments on a collection of large OSNs to measure the efficiency of our proposed algorithms and apply those algorithms to provide better insights into the information leakage process in social networks. We begin with describing the experimental setup. Then we analyze the solution quality and performance of the proposed algorithms. Finally, we seek the answers to the questions around how “celebrities” and their friends can be in control of their shared information.

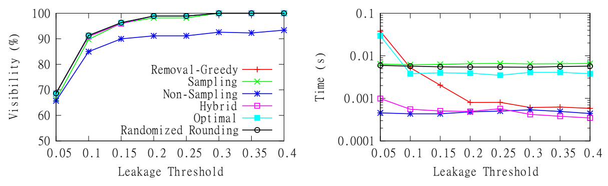
5.1 Experiment Setup

Dataset. We perform experiments on three real-world OSNs, including Facebook, Twitter and Foursquare. The statistics of the used networks can be found in Table 1. The Facebook dataset is obtained thanks to authors in [24]. The Twitter dataset is extracted from the Twitter network provided by Cha *et al.* [5] by applying the unbiased sampling method in [10]. Foursquare dataset is our crawled dataset in which a set entrepreneurs and investors is selected as the seeding. After that, we use Foursquare API to get the users and their connections within two hop from the seeds.

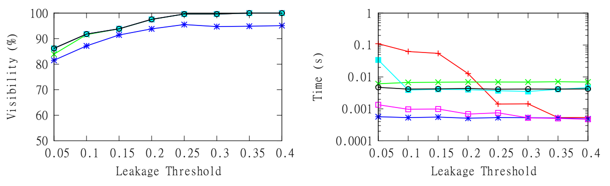
Due to the lack of ground truth, we independently assign uniform random numbers between 0 and 1 to sharing probability a_{uv} and mention probability p_{uv} of each edge. Unless otherwise mentioned, we select the source randomly, then pick up the unwanted targets randomly among the nodes that are within δ hops from the source.



(a) Facebook: Visibility (left), Time (right)



(b) Foursquare: Visibility (left), Time (right)



(c) Twitter: Visibility (left), Time (right)

Figure 4: Performance comparison when $\delta = 2$

Table 1: Dataset

Dataset	Nodes	Edges	Density	Source
Facebook	63,731	905,565	4.46%	Ref [24]
Foursquare	44,832	1,664,402	8.28%	Our data
Twitter	88,484	2,364,322	3.02%	Sampling in [5]

[*] Facebook are undirected networks; Twitter and Foursquare are directed networks.

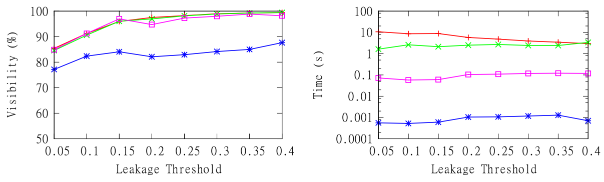
We implemented and compared the following algorithms

1. *Sampling*: IGC algorithm using Monte-Carlo Sampling in section 4.1.1;
2. *Non-Sampling*: IGC algorithm using Disjoint Cut-sets in section 4.1.2;
3. *Hybrid*: IGC-Hybrid algorithm in Section 4.2;
4. *Removal-Greedy*, in which users are removed one by one to construct the CT [22];
5. *Randomized-Rounding*: The approximation randomized algorithm in Section 3. We run the algorithm 20 times and report the average results;
6. *Optimal*: The optimal solution found by solving the Integer Programming (1).

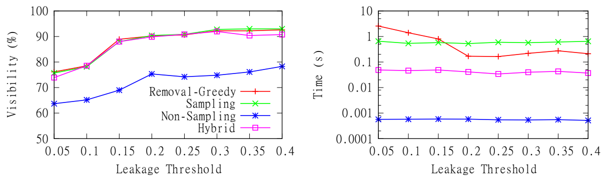
In the sampling subroutine, we stimulate the random process 20,000 times. The optimal solutions and the fractional LP solutions in the randomized algorithm are obtained by using CPLEX optimization package. All algorithms are performed on a cluster of 20 PCs with CPU AMD Opteron 2.00Ghz and 32GB RAM.

Measurements: We measure the following quantities:

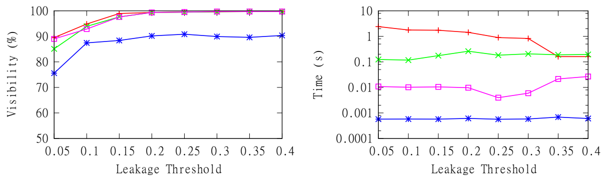
(1) *Visibility(%)*: The expected visibility measured as the percentage of the maximium visiblity when all neighbor of s are included in the circle. Formally, it is defined as $\frac{\sum_{j \in CT} a_{sj}}{\sum_{j \in N(s) \setminus T} a_{sj}}$.



(a) Facebook: Visibility (left), Time (right)



(b) Foursquare: Visibility (left), Time (right)



(c) Twitter: Visibility (left), Time (right)

Figure 5: Performance comparison when $\delta = 3$

(2) *CT Size(%)*: The fraction of the source's friends included in CT, which is defined as $\frac{|CT|}{|N(s)\setminus T|}$. Often, the larger visibility, the bigger CT size, however, it is a more intuitive metric than visibility in some cases.

(3) *Running Time*: The running time in seconds.

5.2 Performance

We measure the visibility(%) and running time when the leakage threshold ranges from 0.05 to 0.4. For each leakage threshold, we randomly choose 100 sets of random sources and five unwanted targets and average the results. The results for $\delta = 2$ and $\delta = 3$ are shown in Figs. 4 and 5, respectively.

5.2.1 *Solution quality*

As shown in Figs. 4, all algorithms Removal-Greedy, Sampling, Hybrid, Randomized Rounding produce optimal or close-to-optimal results in term of visibility. The hybrid algorithm performs slightly worse than the others, however, the gap is usually negligibly small. As expected, non-sampling method shows the worst results among all. The Visibility obtained with Non-sampling methods are 5% to 10% smaller than the others'. The reason is that when the Non-sampling method examines if any nodes can be added to the CT, it is often pessimistic about the current leakage situation and stop early. Clearly, the non-sampling method should not be used as a standalone method to construct CT, but rather as a method to construct an early version of CT. Similar observations for $\delta = 3$ can be seen in Fig. 5.

Moderately surprising, not too many friends need to be blocked to prevent information leakage. Even when the leakage threshold is 5%, the visibility still ranges from 70% to 90%. In addition, the visibility increases quickly when the threshold increases.

We notice that when comparing the cases $\delta = 2$ and $\delta = 3$, the visibility in Foursquare decreases, while those in Facebook and Twitter show the reverse trend. Moreover, in case $\delta = 3$ the visibility in Fourquare does not approach 100% when the leakage threshold increases. The main reason for smaller visibility in Foursquare is the closer distance between sources and targets in Foursquare. This is an artifact caused by the way we crawled the Foursquare network. As we only query the users within two-hops from the seeding in Foursquare, the randomly generated targets in Foursquare are often 2-hops away from the source, regardless of δ. In contrast, there is a high probability that the distance in hop from the source and the target in Facebook and Twitter equal exactly the value of δ. This observation, more or less, supports the hypothesis that the closer the source and the unwanted target, the smaller the CT.

5.2.2 *Running time*

The running time of the studied algorithms are shown in the last rows of Figs. 4 and 5. All methods are adequately fast when $\delta = 2$. When $\delta = 3$, the Non-sampling and Hybrid method are substantially faster than the other methods. Because the Hybrid avoids most of the calls to the sampling at the early selection steps; it is up to a hundred times faster than the Removal-Greedy, and the Sampling methods.

In comparison to the Sampling method, the running time of Removal-Greedy is excessively high for small leakage threshold, when the algorithm takes many rounds to remove nodes until obtaining the CT. In contrast, Sampling method (as well as Hybrid and Non-sampling method) has the running time slightly increased when the leakage threshold increases, which leads to larger CTs. Since we often desire smaller leakage threshold, in practical the algorithms that construct CT by adding nodes may have higher performance than those that find the CT in a removing nodes fashion.

Clearly, for $\delta = 2$ the optimal solution by solving IP is the best selection. When $\delta \geq 3$, the Hybrid is the method of choice due to both its outstanding solution quality and fast running time, less than 0.1 second. Note that the ability to construct CT quickly is critical, if one wishes to implement the algorithm on social network platforms. For larger networks and δ, the only applicable method is, however, the Non-sampling method.

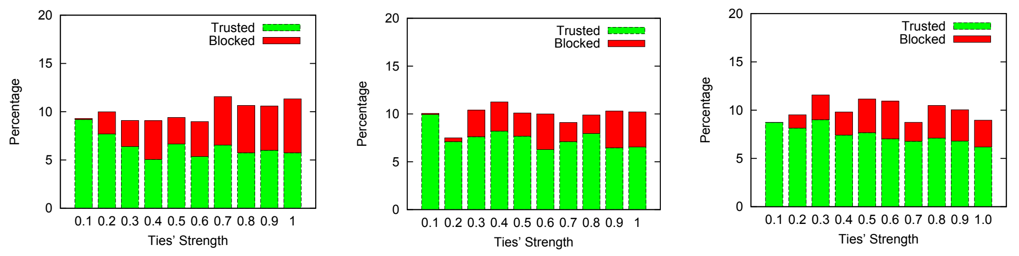
5.2.3 *Ties' Strength of Trusted and Blocked Users*

CT might not be useful if it excludes most of the 'close friends', or strong ties, of the source. Thus we perform experiments to see the distribution of the ties' strength, inferred via the sharing probabilities, of users inside (and outside) CT. In each network, we randomly select 40 source users. For each source user, 100 sets of 5 unwanted targets are randomly selected. Then for each set of unwanted target, we solve IP to find the optimal solution for $\delta = 2$ and the leakage thresholds 0.15. The distribution of ties' strength (sharing probabilities) are shown in Fig. 6.

As shown in Fig. 6, in all the three networks, the number of strong ties inside CT is still greater than the number of strong ties outside CT, even at the highest level of ties' strength. For examples, in Foursquare and Twitter, two third of the strong ties remain in CT. We observe, at the higher levels of ties' strength, there is a tendency of more strong ties being excluded from CT. Nevertheless, the produced CTs are acceptable as they include the majority of the strong ties.

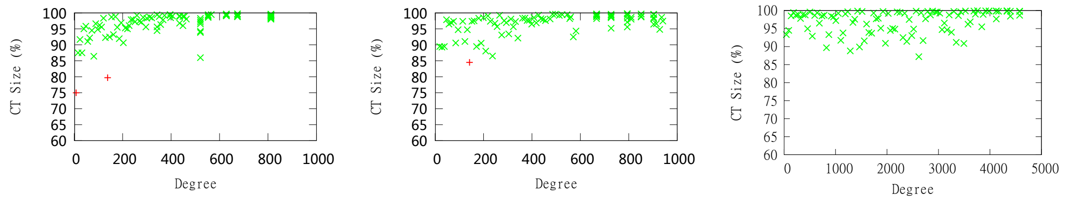
5.3 Case Studies

We provide several observations obtained by solving IP



(a) Facebook (b) Foursquare (c) Twitter

Figure 6: The distribution of ties' strengths between the source and the trusted/blocked users.



(a) Facebook (b) Foursquare (c) Twitter

Figure 7: CT size of sources with different "celebrity level" (degree)

($\delta = 2$) and using the Hybrid method ($\delta = 3$) that give the best performance in general.

Can Celebrities Keep Their Secrets? We investigate the question whether or not "celebrities", users with a large number of friends or followers, can only share their information with a small number of their friends to avoid information leakage. Specifically, we are interested in the relation between the degree of the source and their CT sizes. Intuitively, when a user has more friends, there is a high chance his friends will forward the information further. Thus, to avoid information leakage, the CT of "celebrities" possibly includes only a small fraction of their friends to CT.

To verify the hypothesis, we select the source users with different degrees, increasing by 1% of the maximum degree in each step. For each source, we choose ten random sets of five unwanted targets and compute the average size of CT by solving the IP. The results are shown in Fig. 7.

In contrast of our hypothesis, their is no sharp decrease in the CT size. Indeed, for the source with the highest degree, the CTs contain more than 95% of the neighbors. This suggests that by carefully constructing CT, smartly sharing information to avoid leakage is possible for "celebrities".

Can We Trust Our Popular Friends? There is an incentive not to share our sensitive information to a popular person (celebrity) who will easily leak the information further. In other words, the friends of a celebrity are unlikely to trust him. To investigate the effect of the "celebrity level", we select in each network 10 users with different degree, incrementing by 10% of the maximum degree. For each "celebrity", we select each of his neighbor to be the source and compute the percentage that the "celebrity" is included in his friends' CTs.

As illustrated in the Fig. 9, the "trust level" of the friends for a user decreases quickly when the degree of the user increases, i.e. when the user becomes more popular. However, even for the user with the highest degree, there are still half of his friends trust him as in the cases of Facebook and Foursquare. For Twitter, only 20% of the friends trust the considered user. This can be explained by the fact that Twitter is a directed network with low reciprocity (A follows B, but B does not follow A) [18]. Thus, the celebrities are not even considered to be friends to be included in the CTs. Roughly speaking, "celebrities" in social networks such as Facebook are relatively "trusted", while the same conclusion does not hold for the "social media" network Twitter [18].

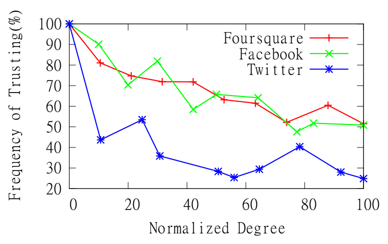


Figure 9: Trusting our Popular Friends

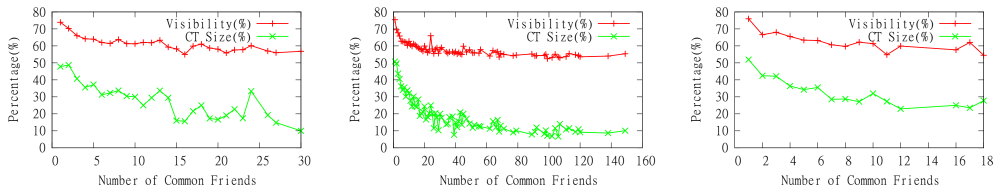
The Unwanted Friend of Friends. Another interesting question is the impact of the relation between source user and the unwanted targets to the CT. We have observed in the Foursquare network $\delta = 3$ that the closer the distance between sources and unwanted targets, the smaller the CT. Here, we study the relation between the number of common friends and the CT size. Is it necessary to block all these common neighbors?

To address these questions, we then plot the average Visibility and CT Size with respect to the number of common friends on Fig. 8. As can be seen, the CT Size decreases sharply from 50% to 10% with the increase of the number of common friends. However, the visibility reduces at a much lower rate. Regardless of the number of common neighbors, the visibility is still nearly 60%.

Therefore, the case when the unwanted targets are close to the source, is not as bad as it appears. The growing gap between the Visibility and CT size reflects that we still can effectively select high visibility neighbors into the CT despite the shrinking of the CT.

6. RELATED WORK

Our work is the first attempt to address the smartly sharing information in OSNs without leaking them to unwanted targets, thus there is not many related work. The most relevant works are the works of selective sharing by Shen et al. [22] and Kairam et al. [15]. Shen et al. [22] address the



(a) Facebook (b) Foursquare (c) Twitter

Figure 8: The impact of common friends between sources and targets

selective sharing problem as an optimization problem and provide various hardness and polynomial-time approximation schemes for the problem. Kairam et al. [15] analyze how users organize and select their audiences and derive interesting observations on how users share across life facets, and ties of vary strength.

There is set of papers studied on the privacy issues in OSNs [19, 20, 21]. Lam *et al.* [19] showed that, in current OSNs, no matter how much efforts a user puts to protect his personal information, it cannot be prevented from being revealed by some malicious users by examining their "public" interactions with friends. Later on, for the sake of such unintentional information spreading, Ngoc *et al.* [20] then presented a new metric to quantify the privacy. Noting the potential risks by disclosing information to OSN companies, Nigusse *et al.* [21] proposed an *information flow model*, which made the existing privacy techniques more practical. However, these studies only focus on the users' personal profile, i.e. name, address, etc., not on the information sharing and posting. In addition, they neglected the information leakage led by multi-hops diffusion.

7. CONCLUDING REMARKS

In this paper, we study a problem of how smartly sharing our information in OSNs without leaking them to unwanted targets. We formulate this problem as the optimization problem, namely maximizing a circle of trust (MCT), of which we construct a circle of trust to maximize the expected visible friends such that the probability of information leakage is reduced to some degree. We have proven the inapproximability and provided a randomized algorithm with theoretical guarantees for the 2-MCT problem. When the information can be propagated more than 2 hops, we provided an iterative greedy algorithms based on a novel disjoint-cut estimation, instead of using sampling method. Extensive experiments in real-world traces show that our solutions can construct a large size of CT within a short amount of time.

There are many future directions emerged from our work. First, the knowledge about how people form 'circles' of friends and share information can be incorporated to identify the proper input parameters for the SML models. Second, the MCT problem can be studied under different propagation models in literature. Furthermore, we believe controlling the flow of the shared information is an important issue which can inspire many other problems.

8. REFERENCES

[1] http://www.marketingcharts.com.
[2] N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley, New York, 1992.
[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. KDD '08, pages 7–15, New York, NY, USA, 2008. ACM.
[4] S. Arora and B. Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.
[5] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *ICWSM*, May 2010.
[6] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. WWW '09, pages 721–730, New York, NY, USA, 2009. ACM.
[7] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD '10*, pages 1029–1038, New York, NY, USA, 2010. ACM.
[8] B. Darwell. Edgerank and graph rank defined. `http://www.insidefacebook.com/2011/12/27/edgerank-and-graph-rank-defined/`, 2011.
[9] M. Fraser and S. Dutta. Barack obama and the facebook election. `http://www.usnews.com/opinion/articles/2008/11/19/barack-obama-and-the-facebook-election`.
[10] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *IEEE INFOCOM 2010*, pages 1–9, Mar. 2010.
[11] J. Goldenberg, B. Libai, and E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, pages 211–223, Aug. 2001.
[12] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. WSDM '10, pages 241–250, New York, NY, USA, 2010. ACM.
[13] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. WWW '04, pages 491–501, New York, NY, USA, 2004. ACM.
[14] J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. In *FOCS '96*, page 627, Washington, DC, USA, 1996. IEEE Computer Society.
[15] S. Kairam, M. J. Brzozowski, D. Huffaker, and E. H. Chi. Talking in circles: Selective sharing in google+. CHI '12, New York, NY, USA, 2011. ACM.
[16] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.
[17] D. Kempe, J. Kleinberg, and ÃL'va Tardos. Influential nodes in a diffusion model for social networks. In *ICALP*, pages 1127–1138. Springer Verlag, 2005.
[18] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.
[19] I.-F. Lam, K.-T. Chen, and L.-J. Chen. Involuntary information leakage in social network services. IWSEC '08, pages 167–183, Berlin, Heidelberg, 2008. Springer-Verlag.
[20] T. H. Ngoc, I. Echizen, K. Komei, and H. Yoshiura. New approach to quantification of privacy on social network sites. AINA '10, pages 556–564, Washington, DC, USA, 2010. IEEE Computer Society.
[21] G. Nigusse and B. D. Decker. Privacy codes of practice for the social web: The analysis of existing privacy codes and emerging social-centric privacy risks. In *AAAI Spring Symposium Series*, 2010.
[22] Y. Shen, Y.-S. Syu, D. T. Nguyen, and M. T. Thai. Maximizing circle of trust in online social networks. HT' 12. ACM, 2012.
[23] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. KDD '09, pages 807–816, New York, NY, USA, 2009. ACM.
[24] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the Evolution of User Interaction in Facebook. In *Proceedings of WOSN'09*, Aug. 2009.
[25] S. T. Walters and C. Neighbors. Feedback interventions for college alcohol misuse: what, why and for whom? *Addict Behav*, 30(6):1168–82, 2005.