Mitigating Sybil attacks on content rating systems

Arash Molavi Kakhki[†§]

Aniko Hannak[†§]

Alan Mislove[†]

Ravi Sundaram[†]

[†]Northeastern University

[§]Student

1. INTRODUCTION

Online content sharing services allow users to find and share content ranging from news articles (Digg) to videos (YouTube) to URLs (StumbleUpon). Generally, such *social content* sites allow users to create accounts, declare friendships, upload and rate content, and locate new content by leveraging the aggregated ratings of others. For example, most highly rated content typically appears on the front page of the site (or on the user's front page after logging in), garnering significant attention and traffic.

However, the user accounts on these sites are not verified and are essentially free to create. For example, creating an account on most sites only requires proving ownership of an email address and solving a CAPTCHA. Unfortunately, this combination of free accounts and content rating privileges—hereafter known as voting—associated with each account is leading to unintended consequences: Malicious users are naturally incentivized to create multiple accounts (known as a Sybil attack [1]), and have been observed to use multiple accounts to manipulate the voting system in order to have advertisements or other malicious content rated highly [2].

Recent work has attempted to defend against Sybils by leveraging the social network [3]. However, such systems do not directly apply to voting, as a typical piece of content only receives a small number of votes, and even a few malicious identities can quickly outvote the honest users. DSybil [4] has taken an alternate approach of finding trusted users in the network, but can only provide guarantees for users who have submitted a sufficient number of votes (often a small fraction of the population in practice). SumUp [2] uses tokens passed over the social network (and inspired our design), but has two subtle weaknesses: (a) SumUp assumes that the region of the social network surrounding the user requesting the vote is free of Sybils, and (b) SumUp incentivizes users with multiple social network links to create Sybils, as this can lead to greater influence on the vote outcome.

In this proposal, we explore a new approach for mitigating the impact of Sybils in online content sharing services. In brief, we assign *weights* to each user's vote. To mitigate Sybil attacks, we ensure that the total weight of votes placed by a user controlling multiple identities is the same as the weight of the vote placed if the user instead had only a single identity. As a result, users gain no additional influence on the outcome of a vote by creating multiple identities.

2. MODEL AND ASSUMPTIONS

Similar to prior work, we assume that the identities are connected by a social network $G = (V, E)$. We further assume that links to an honest user take effort to form and maintain; in other words, a malicious user cannot obtain an arbitrary number of links to honest users. Note that we make no assumptions about the number of identities malicious users possess, or the structure of the links between malicious identities.

The purpose of our system is to summarize the votes placed by other identities on a given piece of content. For simplicity, we assume that the vote of a user u is a real-valued function $v(u) \in [0, 1]$, with 1 representing "good" and 0 representing "bad". To summarize the votes, we assign each voting user a weight $w(u) \in [0, \infty)$ and take the weighted average to be the summarized vote on the object. Specifically, if $W \subset V$ is the set of voting users, the aggregated vote is

$$\frac{\sum_{u \in W} w(u)v(u)}{\sum_{u \in W} w(u)}$$

Unfortunately, calculating a global summarization of the votes is both error-prone and undesired. A global assignment assumes that all users value all content and all other users' votes equally; this is clearly not the case for sites like YouTube and Digg, where different users share wildly different types of content. Instead, we create a *personalized* content rating for each user by calculating the weights *per user*. Thus, the summarized vote that one user sees for a given piece of content may be different from the one another user sees, depending on

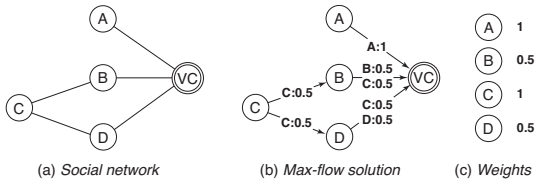

Figure 1: Shown are a (a) social network, (b) resulting max flow solution, and (c) resulting user vote weights for an example with voting users $A...D$ and vote collector VC.

where the two users are located in the social network and the set of users near them who vote. For the remainder of this proposal, we refer to the user for whom we are calculating the vote as the *vote collector* [2].

3. ASSIGNING WEIGHTS

In order to assign the vote weights to users, we use flow over the social network. Consider the social network shown in Figure 1 (a), with the vote collector labeled VC and the voting users labeled $A...D$. We create a multi-commodity max flow problem, with all social network links having unit capacity and each user's vote being a flow with a source of the user placing the vote and a sink of the vote collector.

The output of the linear solver is shown in Figure 1 (b), with the amount of flow that each voting user is able to send shown on each link. For example, user C is able to send 0.5 units of flow along the path $C \rightarrow B \rightarrow VC$ and 0.5 units of flow along the path $C \rightarrow D \rightarrow VC$. In order to determine each user's weight, we simple sum up the total amount of flow each user is able to send. For our example problem, the resulting weights are shown in Figure 1 (c).

We now demonstrate that multi-commodity max flow, described above, ensures that users who create multiple identities are unable to gain any additional weight. To see why, consider the alternate social network shown in Figure 2 (a). This network is identical to the one in Figure 1 (a), with the exception that user A has created two Sybil nodes A' and A'' and attached them to himself (recall that we made no assumptions about the links between malicious users). Now, we wish to show that the aggregate weight of the identities controlled by A is the same as if A had only a single identity.

Let us examine the output of the max flow problem on our modified social network, shown in Figure 2 (b). Note that, regardless of the number of identities that lie "behind" node A, the total flow to VC is restricted by the cut between these identities and the honest region of the network (namely, the $A \leftrightarrow VC$ link). Thus, A can share his weight with his other identities, but in total, the weight they receive is the same as A received before he created the additional identities.

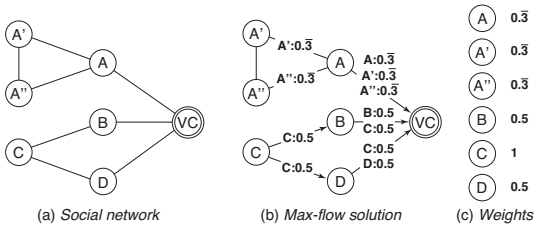

Figure 2: A modified version of Figure 1, with user A creating additional voting identities A' and A''. Shown in (c), the total weight received by A's identities is the same as when A had a single identity.

4. PRELIMINARY RESULTS

We demonstrate our approach on a data set from Yelp containing 152K reviews on 6.9K businesses from 65K users. We construct social network links between users who write at least 3 reviews on the same business, resulting in 159K links. We simulate Sybil attacks by attaching a Sybil network to the social network with varying numbers of *attack links*. The honest votes are from the Yelp dataset, with an average of 4.2 stars; all of the Sybils issue votes with 5 stars.

The results are shown in Figure 3. Without any Sybil defense, the aggregate vote approaches 5 stars as more Sybils vote; with our approach, the aggregate vote stays constant, regardless of the number of Sybils that vote.

5. REFERENCES

[1] J. Douceur. The Sybil Attack. *IPTPS*, 2002.
[2] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-Resilient Online Content Voting. *NSDI*, 2009.
[3] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A Near-Optimal Social Network Defense Against Sybil Attacks. *IEEE S&P*, 2008.
[4] H. Yu, C. Shi, M. Kaminsky, P. B. Gibbons, and F. Xiao. DSybil: Optimal Sybil-Resistance for Recommendation Systems. *IEEE S&P*, 2009.

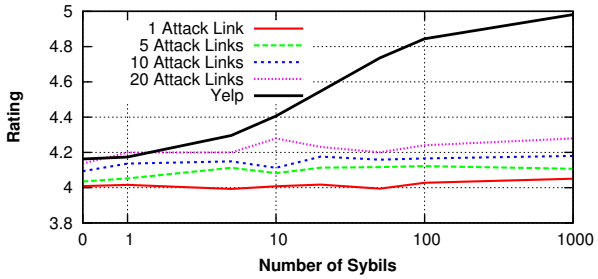

Figure 3: Aggregate vote for Yelp and for our approach (with varying numbers of attack links). For a given number of attack links, the Sybils only receive a fixed influence on the aggregate vote, regardless of the number of votes they cast.