11-16-2015

Content Abuse and Privacy Concerns in Online Social Networks

Md Imrul Kayes
University of South Florida, imrul@mail.usf.edu

Content Abuse and Privacy Concerns in Online Social Networks

by

Md Imrul Kayes

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Adriana Iamnitchi, Ph.D.
Swaroop Ghosh, Ph.D.
Yao Liu, Ph.D.
John Skvoretz, Ph.D.
Kingsley A. Reeves, Jr., Ph.D.

Date of Approval:
October 19, 2015

Keywords: Community Question Answering, Crowdsourcing, User Behavior, Cross-cultural Variations, Contextual Integrity

DEDICATION

To my mother Nasrin Sultana.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my heartfelt gratitude to my major professor Dr. Adriana Iamnitchi for her invaluable guidance and support. She has always encouraged me and allowed me to explore new research directions that drove my interests.

I extend my gratitude to Dr. Swaroop Ghosh, Dr. Yao Liu, Dr. John Skvoretz, and Dr. Kingsley A. Reeves, Jr., for being a part of my defense committee, and Dr. Lingling Fan for serving as the chair in my doctoral defense.

I thank my lab mates during my time in the Distributed Systems Group. Xiang, in particular, was always available for any discussion. Thank you for your opinion and suggestion. I would like to acknowledge Yahoo Labs, Barcelona for the internship and Yahoo Answers dataset. I am thankful to Dr. Nicolas Kourtellis and Dr. Daniele Quercia for their guidance and encouragement during the internship.

Lastly, I would like to thank Tanvir, Eftay, Subhranil, Wasie, Subhabrata, Bashar, Reza, Nashid, and Ruhin for giving me emotional and practical support during my doctoral studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Online Social Networks (OSNs) have seen an exponential growth over the last decade, with Facebook having more than 1.49 billion monthly active users and Twitter having 135,000 new users signing up every day as of 2015. Users are sharing 70 million photos per day on the Instagram photo-sharing network. Yahoo Answers question-answering community has more than 1 billion posted answers. The meteoric rise in popularity has made OSNs important social platforms for computer-mediated communications and embedded themselves into society's daily life, with direct consequences to the offline world and activities. OSNs are built on a foundation of trust, where users connect to other users with common interests or overlapping personal trajectories. They leverage real-world social relationships and/or common preferences, and enable users to communicate online by providing them with a variety of interaction mechanisms.

This dissertation studies abuse and privacy in online social networks. More specifically, we look at two issues: (1) the content abusers in the community question answering (CQA) social network and, (2) the privacy risks that comes from the default permissive privacy settings of the OSNs. Abusive users have negative consequences for the community and its users, as they decrease the community's cohesion, performance, and participation. We investigate the reporting of 10 million editorially curated abuse reports from 1.5 million users in Yahoo Answers, one of the oldest, largest, and most popular CQA platforms. We characterize the contribution and position of the content abusers in Yahoo Answers social networks. Based on our empirical observations, we build machine learning models to predict such users.

Users not only face the risk of exposing themselves to abusive users or content, but also face leakage risks of their personal information due to weak and permissive default privacy policies. We study the relationship between users' privacy concerns and their engagement in Yahoo Answers social networks. We find privacy-concerned users have higher qualitative and quantitative contributions, show higher retention, report more abuses, have higher perception on answer quality and have larger social circles. Next, we look at users' privacy concerns, abusive behavior, and engagement through the lenses of national cultures and discover cross-cultural variations in CQA social networks.

However, our study in Yahoo Answers reveals that the majority of users (about 87%) do not change the default privacy policies. Moreover, we find a similar story in a different type of social network (blogging): 92% bloggers' do not change their default privacy settings. These results on default privacy are consistent with general-purpose social networks (such as Facebook) and warn about the importance of user-protecting default privacy settings.

We model and implement default privacy as contextual integrity in OSNs. We present a privacy framework, Aegis, and provide a reference implementation. Aegis models expected privacy as contextual integrity using semantic web tools and focuses on defining default privacy policies. Finally, this dissertation presents a comprehensive overview of the privacy and security attacks in the online social networks projecting them in two directions: attacks that exploit users' personal information and declared social relationships for unintended purposes; and attacks that are aimed at the OSN service provider itself, by threatening its core business.

CHAPTER 1: INTRODUCTION[1]

Online Social Networks (OSNs) have become a mainstream cultural phenomenon for millions of Internet users. Combining user-constructed profiles with communication mechanisms that enable users to be pseudo-permanently “in touch”, OSNs leverage users’ real-world social relationships and/or common interests and blend even more our online and offline lives. As of 2015, Facebook has 1.49 billion monthly active users and it is the second most visited site on the Internet [7]. Twitter, a social micro-blogging platform, claims over 314 million monthly active users, who send 500M Tweets per day in more than 35 languages [280]. According to the GlobalWebIndex’s 2014 survey, users on average spend 1.69 hours each day doing social networking [25].

In general, OSNs are a digital representation of the sets of relations, which the registered users exercise in the physical world [61]. This digital representation forms a network of users, which spans all enclosed parties and their contacts. Moreover, there are interest-based social networks, where users get connected to people they may not even know to cater common interests. For example, in Pinterest, users tend to find photos of shared passions. In Yahoo Answers, people follow users who provide questions and answers on a topic of common interest. Boyd and Ellison’s widely used definition captures the key elements of an OSN, according to which Social Network Sites or Online Social Network Services are:

“web-based service that allows individuals to (i) construct a public or semi-public profile within the bounded system, (ii) articulate a list of other users with whom they share a connec-

[1] Much of the work in this chapter was first published in [145, 146, 148–150]. Permission is included in Appendix A.

tion, and (iii) view and traverse their list of connections and those made by others within the service" [37].

Their definition captures some basic elements of an OSN, however, OSNs offer some additional elements depending on the type of the OSN. For example, message exchange functionality enables users to communicate with other users, multi-media features allow users to upload and share photos, audio and videos, users can annotate contents using tag functionality, group features enable users to join various social groups and collaborate with people who have common preferences.

Perhaps more than previous types of online applications, OSNs are blending in real life: companies are mining trends on Facebook and Twitter to create viral content for shares and likes; employers are checking Facebook, LinkedIn and Twitter profiles of job candidates [230]; law enforcement organizations are gleaning evidence from OSNs to solve crimes [153]; activities on online social platforms change political regimes [185] and swing election results [141].

1.1 Abuse and Privacy in Social Networks

As users in OSNs are typically connected to friends, family, and acquaintances, a common perception is that OSNs provide a more secure, private and trusted internet-mediated environment for online interaction [61]. In reality, however, abuse and privacy issues are common in OSNs.

Generally speaking, abusive behavior is a pervasive and serious problem on the Internet, perhaps since the day it has become popular among the common public. This problem has been aggravated in OSNs, as it has become easier to interact with others and get more exposure of the published content. Moreover, similar to other online environments (e.g., chat rooms, forums), anyone can create an anonymous or fake identity in OSNs and can exhibit abusive behavior without fear of damaging reputations. With more people joining the OSNs, even the most popular social network sites are overwhelmed by the degree and consequences of abuse. Recently, Twitter's CEO has admitted the failure and frustration of dealing with abuse and trolling [126].

Community Question Answering (CQA) sites, such as Yahoo Answers and Quora, are crowd-sourced services for sharing user expertise on various topics, from mechanical repairs to parenting. These services are suitable platforms for studying content abuse at scale for multiple reasons. First, social networks in CQA platforms are built on common interests, that carry a very different purpose from Facebook-like social networks, where users "hang-out" with their friends. As real-world social relationships are less emphasized in the CQA online relationships, users may care less about interacting with others compared to Facebook-like social networks where most of one's friends are personally known. Second, similar to general-purpose OSNs (e.g., Facebook), CQA platforms are rich and mature repositories of user-contributed content (questions and answers), thus, they enable to study content abuse at scale. Third, as the usefulness of CQA platforms depends heavily on fair user contributions (questions and answers), they employ strict community rules and regulations and human moderators. As such, ground truth data of abusive content are reliable, as the content is user reported and human verified.

Not only users are facing the risk of exposing themselves to abusive content or users, but also users' privacy is also at risk in social networks because of the availability of an astonishing amount of personal user data which would not have been exposed otherwise. More importantly, OSNs expose new information from multiple social spheres – for example, personal information on Facebook and professional activity on LinkedIn – that, aggregated, leads to uncomfortably detailed profiles [218]. This high volume of personal data, either disclosed by the technologically-challenged average user or due to OSNs' failure to provide sophisticated privacy tools, have attracted a variety of organizations (e.g., GNIP) that aggregate and sell user's social network data. Unwanted disclosure of user information combined with the OSNs-induced blur between the professional and personal aspects of user lives allow for incidents of dire consequences. The news media covered some of these, such as the case of a teacher suspended for posting gun photos [62] or employee fired for commenting on her salary compared with that of her boss [190]), both on Facebook.

In general, OSN privacy problems are of three types: *surveillance*, *institutional* and *social privacy* [120]. The surveillance privacy threat arises when users' personal information and social interactions are leveraged by authorities or service providers for unintended purposes. For example, facial-recognition technologies have made social tagging easier—users can conveniently tag their friends in pictures. However, at the same time, this has aggravated surveillance risks. Dictatorial and authoritarian regimes could identify protesters from demonstration pictures [68]. Not only that, even democratic governments might be overly curious and keep the mass public under state surveillance. The US government's PRISM project targeting various Internet service providers, including top OSNs has generated much controversy in recent years.

Institutional privacy refers to those privacy problems related to users losing control and oversight over the institutional aggregation, processing and mining of social information [237]. For example, professional data aggregators build databases using public views of social media profiles and social relationships and sale the databases to insurance companies, background-check agencies and credit-ratings agencies [31].

"Social privacy" problems emerge when OSN mediated social interactions disrupt social boundaries [120]. Contrary to surveillance and institutional privacy problems that come from established and consolidated group entities such as government or organizations, social privacy problems are related to other individuals in the social network. The consequences of social privacy problems include, but are not limited to, information oversharing (e.g., using default permissive OSN privacy settings expose more information), unwanted contacts (e.g., privacy settings might allow anyone to contact or send messages), damaged reputations, and context collapse (e.g., coworkers, family members, close friends, or acquaintances are flattened into a single homogeneous group such as Facebook "Friends").

1.2 Research Questions

This dissertation focuses on abuse in community question answering (CQA) social networks and the social privacy risks that come from the default privacy settings of the OSNs. The thesis statement is as follows:

Lessons from large-scale analysis of user behavior in online social networks have the potential to limit content abuse, improve user engagement, and define appropriate privacy settings.

User behavior is considered abusive if it violates community norms and users feel that it should not belong to that particular online environment [41, 67]. In the real world of face-to-face social interactions, anonymity and deindividuation have been linked to increased likelihood of nonconformity with social norms. These phenomena are more pronounced in general online settings [67, 140]. Abusive behavior has negative effects on the community and its members: it decreases community's cohesion [294], performance [82] and participation [67]. In the worst case, users who are the targets of abusive behavior may leave or avoid online social spaces [67].

Social networks that are built on common interests, like Yahoo Answers CQA social networks, also experience a high volume of abusive content. In order to limit abuse, CQAs define community rules and expect users to obey them. To enforce these rules, published as community guidelines and terms of services, these platforms provide users with tools to flag inappropriate content. In addition to community monitoring, some platforms employ human moderators to evaluate abuses and determine the appropriate responses, from removing content to suspending user accounts. These digital recordings of abusive behaviors enable the study of human behaviors at much larger scale than what is possible in lab experiments and has the potential of guiding the design of mechanisms that foster good behavior. We can leverage them to automatically predict abusive users, that could not only keep fair users safe from abusive users, but also lessen workload of human moderators. More specifically, we ask the following research questions:

1 : How can we use crowd-sourced rule violations reports to understand the position and the contribution of abusive users?

2 : How can we design predictive models to automatically predict abusive users?

Moreover, users expose an unprecedented amount of personal information, and face privacy leakage risks. Hence, it's expected for users to be concerned about their privacy. The important question is how users' privacy concerns are related to their engagement—are privacy concerns turning users off from participating and contributing? On one hand, users might be afraid that their privacy is being compromised, so they might produce less content and show less engagement. On the other hand, users might also properly exercise their privacy rights (for example, by restricting the visibility of their content) and hence feel having more control over their content and contribute more. We focus on community question answering social networks and ask the following question:

3 : What is the relationship between users' privacy concerns and their contribution behavior in CQA social networks?

Users' (un)ethical behavior, privacy perception, and engagement might not be the same across cultures. Researchers sometimes predicted that the online world would be converging into a "one-world culture" [175]. However, research has already shown that the Internet is not a homogeneous subcultural community, significant behavioral differences exist between users from different countries [104, 224, 239]. We might also see differences if we zoom into users' engagement, privacy concerns and (un)ethical behavior to country levels. We focus on community question answering social networks and ask the following question:

4 : Do privacy concerns and contribution behavior vary across cultures in CQA social networks?

Our study of privacy concerns vs. contribution behavior reveals that the majority of the users do not change their privacy settings. Studies in other social networks reveal similar stories. For example, studies [1, 163] show that the majority of Facebook users have default or permissive privacy settings. More worrisome, when the default settings are not matched with user preferences, they almost always tend to be more open, exposing the content to more users than expected [184]. Much of the social privacy problems occur when access to social data is inappropriately protected due to wrong default or personalized privacy settings. Often the default settings serve the business model of the service provider rather than the user's interests, following the "opt out" model. And as we see from the literature and our results on Yahoo Answers social network, although users are allowed to change the default permissive privacy settings, in reality very few do it.

Default privacy related problems have been further aggravated in social ecosystems. Social ecosystems, aggregate user data from various sources (e.g., Facebook, LinkedIn, Twitter) and provide processed and useful information to social applications. This aggregation of data from different contexts presents a more complete profile of a person's life, hence more vulnerable to permissive default privacy settings. As such, appropriate default privacy settings are expected that will allow user information to be shared or transferred appropriately. We ask the following research questions:

5 : How can we limit the vulnerabilities associated with default permissive privacy policies in social ecosystems? Can we generate default privacy policies that restrict user information to be shared or transferred inappropriately?

1.3 Contributions

This dissertation makes the following contributions:

- *Characterizing and predicting the content abusers in CQA social networks* [150]: We investigate the reporting of rule violations (flags) and identify content abusers in Yahoo Answers. The usefulness of CQAs depends heavily on user contributions (questions and answers), but also on respecting the community rules. As a crowd-sourced service, such platforms rely on their users for monitoring and flagging content that violates community rules. Common wisdom is to eliminate the users who receive many flags. Our analysis of a year of traces from Yahoo Answers site shows that the number of flags does not tell the full story: on one hand, users with many flags may still contribute positively to the community. On the other hand, users who never get flagged are found to violate community rules and get their accounts suspended. This analysis, however, also shows that abusive users are betrayed by their network properties: we find strong evidence of homophilous behavior and use this finding to detect abusive users who go under the community radar. Based on our empirical observations, we build a classifier that is able to detect abusive users with an accuracy as high as 83%.

- *An analysis of how privacy concerns correlate with user behavior in the CQA social network* [148]: We analyze one year of recorded traces from Yahoo Answers to understand the association between users' privacy concerns as manifested by their account settings and their activity in the CQA platform. The results show that privacy preference is correlated with behavior in the community in terms of engagement, retention, accomplishment and deviance from the norm. We find privacy-concerned users have higher qualitative and quantitative contributions, show higher retention, report more abuses, and have larger social circles. However, at the same time, these users also exhibit more deviant behavior than the users with public profiles.

- *An analysis of cross-cultural variations in CQA social networks* [149]: We investigate the influence of national culture on people's online questioning and answering behavior. For this, we analyzed a sample of 200 thousand users in Yahoo Answers from 67 countries. We measure empirically a set of cultural metrics defined in Geert Hofstede's *cultural di-*

mensions and Robert Levine's *Pace of Life* and show that behavioral cultural differences exist in community question answering platforms. We find that national cultures differ in Yahoo Answers along a number of dimensions such as temporal predictability of activities, contribution-related behavioral patterns, privacy concerns, and power inequality.

- *Aegis: A Semantic Implementation of Privacy as Contextual Integrity in Social Ecosystems* [145, 146]: We propose an ontology-based social ecosystem data model to capture users' aggregated social data from diverse sources (e.g., Facebook, LinkedIn etc.). This data model can be used to acquire information from an unrestricted set of social sources and export it to an ever-evolving collection of socially-aware applications and services. We employ semantic web technologies to generate default privacy policies based on Nissenbaum's contextual integrity theory [216]. We provide an architecture and a prototype implementation of our privacy model that automatically enforces access control policies on a social ecosystem knowledge base. Our experimental evaluation on three real-world large networks demonstrates the applicability in practice of our solution.

- *A comprehensive review of privacy and security in online social networks* [147]: We provide an overview of the privacy and security issues that emerged so far in OSNs. We introduce a taxonomy of privacy and security attacks in OSNs, we overview existing solutions to mitigate those attacks, and outline challenges still to overcome.

1.4 Outline

The remainder of this dissertation is organized as follows. Chapter 2 reviews related work that this dissertation builds on. We describe the dataset in Chapter 3. In Chapter 4, our work characterizes the position of the content abusers in the CQA social network and predicts such user. Chapter 5 shows how users' privacy concerns relate to their engagement related contribution behavior in CQA social networks. Chapter 6 zooms in on users' contribution, privacy concerns, and (un)ethical behavior in CQA social networks to the national culture level and investigates cross-country cultural variations in Yahoo Answers. We present *Aegis*, a privacy framework and a reference implementation in Chapter 7. Aegis implements privacy as contextual integrity by using semantic web tools and focuses on defining default privacy policies. Next, in Chapter 8, we present a review of privacy and security attacks in OSNs projecting them on two directions: attacks that exploit the implicit trust embedded in declared social relationships; and attacks that harvest user's personal information for ill-intended use. Finally, we conclude the dissertation in Chapter 9 with a summary of the main results and a discussion on future research directions.

CHAPTER 2: RELATED WORK[1]

In this chapter, we present the previous research that the dissertation builds upon. First, we present research on CQA social networks in Section 2.1. Section 2.2 presents the studies related to the relationship between users' privacy concerns and their behavior in OSNs. We present previous work on OSN cultures in Section 2.3. Finally, we present the access control mechanisms in OSNs related to Aegis in Section 2.4.

2.1 Community Question Answering (CQA) Social Networks

CQAs have attracted much research interest from diverse communities as information science, HCI and information retrieval. We collate past research on community-based question answering in five categories depending on whether it has dealt with content, users, applications, bad behavior in online settings, or CQA communication networks.

Research in the content area has investigated textual aspects of questions and answers. In so doing, it has proposed algorithmic solutions to automatically determine: the quality of questions [177, 269] and answers [4, 245], the extent to which certain questions are easy to answer [81, 241], and the type of a given question (e.g., factual or conversational) [122].

Research on CQA users has explored how users interface with the platform. Dearman et al. [72] asked why users of *YA* do not answer questions and found that active answerers (who contribute most of the answers) do not want to get reported for abuse and potentially lose access to the community. Liu et al. [183] asked why users ask questions. They concluded that a vast majority of the askers are failed searchers; when web search fails they become askers. Liu et

[1] Much of the work in this chapter was first published in [145, 146, 148–150]. Permission is included in Appendix A.

al. [182] also explored the factors (e.g., when users tend to answer and how they choose questions) that influence users' answering behavior in *YA*. Pelleg et al. [226] investigated truthfulness of *YA* users and found that users even post sensitive and accurate information about themselves while asking questions in order to get right answers.

Research on applications has developed techniques and tools to improve system performance and to provide better usability. Researchers have proposed effective ways of recommending questions to the most appropriate answerers [232, 272]. Shtok et al. [249] used the repository of past answers to answer new open questions in order to reduce the number of unanswered questions. Weber et al. [293] derived "tips" (a self-contained bit of non-obvious answer) from *YA* to address "how-to" questions.

Qualitative and quantitative studies of bad behavior in online settings have been done before, including newsgroups [227], online chat communities [268], online multiplayer video games [29], and geosocial networks [124]. A body of work has also investigated the impact of the bad behavior. Researchers have found that bad behavior has negative effects on the community and its members: it decreases community's cohesion [294], performance [82] and participation [67]. In the worst case, users who are the targets of bad behavior may leave or avoid online social spaces [67].

Research on communication networks analyzed users' social networks on the CQA platforms and attempted to understand the interplay between users' social connections and their Q&A activities. Wang et al. [289] analyzed the social network of Quora and found that users who contribute more and better answers tend to have more followers. Panovich et al. [222] evaluated the impact of tie strength in question answers. They found that stronger ties (close friends) contribute a subtle increase in answer quality compared to weak ties. Rodrigues et al. [199] analyzed intent when users ask questions: along with sharing knowledge, do users want to socialize as well in CQAs? They categorized the questions into *social*, i.e., questions that are intended for purely social engagement and *non-social* (e.g., information seeking, asking opinion) categories. Experimenting on *YA* and *MSN* CQA platforms they concluded that users not only share knowledge, but also socialize with others.

Our work in Chapter 4 sheds light on abusive behavior in CQA communities by studying *YA*, one of the largest and oldest such communities. It quantifies how *YA*'s networks channel user attention, and how that results in different behavioral patterns that can be used to limit abusive behavior.

2.2 Relationships Between Privacy Concerns and User Behavior in OSNs

A number of studies [63, 235, 259] on social networks like Facebook have shown the correlation between users' self-reported privacy concerns and their self-reported behavior. For example, Staddon et al. [259] showed that users who express concerns on Facebook privacy controls and find it difficult to comprehend sharing practices also report less engagement such as visiting, commenting, and liking the content. At the same time, users who report more control and comprehension of privacy settings and their consequences are more engaged with the platform. Similarly, the frequency of visits, type of use, and general Internet skills are shown to be related to the personalization of the default privacy settings [63]. Acquisti and Gross' [1] survey on Facebook found that a user's privacy concerns are only a weak predictor of his joining the network: that is, despite expressing privacy concerns, users join the network and reveal great amounts of personal information. Young et al. [304] used surveys and interviews on Facebook users to show that Internet privacy concerns and information revelation are negatively correlated. Tufekci's study [279] on a small sample (704) of college students showed that students on Facebook and Myspace manage privacy concerns by adjusting profile visibility but not by restricting the profile information.

Wang et al.'s [291] demographic study on privacy concerns among American, Chinese, and Indian social network users showed that American respondents are the most privacy concerned, followed by Chinese. However, all of these studies are survey based and are subject to bias. Moreover, there has been no research on privacy concerns and user behavior in CQA platforms. In this dissertation, we focus on understanding how the users' behavior—characterized by broad engagement, accomplishments and deviance metrics—relates to their privacy concerns in CQA platforms using large-scale real activity traces of the users.

2.3 Cultures in Social Networks

Golder and Macy [111] studied collective mood in Twitter across countries from 509 million Twitter posts by 2.4 million users over a 2-year period. Despite having different cultures, geographies, and religions, all countries (USA, Canada, UK, Australia, India, and English-speaking Africa) in their study showed similar mood rhythms—people tended to be more positive on weekends and early in the morning. Park et al. [224] examined the variation of Twitter users' emoticon usage patterns in cross cultures. They used Hofstede's national culture scores of 78 countries and found that collectivist cultures favor vertical and eye-oriented emoticons, where people within individualistic cultures favor horizontal and mouth-oriented emoticons. Hofstede's cultural dimensions have also been used to study whether culture of a country is associated with the way people use Twitter [105]. In another study on cross-country Twitter communication, Garcia et al. [104] showed that cultural variables such as Hofstede's indices, language and intolerance have an impact on Twitter communication volume.

Silva et al. [250] used food and drink check-ins in Foursquare to identify cultural boundaries and similarities across populations. They showed that online footprints of foods and drinks are good indicators of cultural similarities between users, e.g., lunch time is the perfect time for Brazilians to go for slow food places more often, whereas Americans and English people go for slow foods more at dinner time. Extracted features like these allow them to apply simple clustering algorithms such as K-means to draw cultural boundaries across the countries.

Quercia [234] used *Satisfaction With Life* tests and measured happiness of 32,787 Facebook users from 12 countries (Australia, Canada, France, Germany, Ireland, Italy, New Zealand, Norway, Singapore, Sweden, UK, USA). He found that despite comparative economic status, country-level happiness significantly varies across the countries and that it strongly correlates with official well-being scores.

Reinecke et al. [239] used about 1.5 million Doodle polls from 211 countries and territories and studied the influence of national culture on people's scheduling behavior. Using Hofstede's cultural dimensions, they found that Doodle poll participants from collectivist countries find more consensus than those from predominantly individualist societies.

However, there has been no empirical cross-cultural analysis of CQA platforms. This dissertation is a first step in this direction and it verifies whether cultural differences are manifested in one such platform, *YA*.

2.4 Access Control Mechanisms in OSNs

Different solutions have been proposed to control access to users' data on social networking applications in response to increasing popularity in this type of applications.

Trust-based access control policies are inspired by research and development in trust and reputation in social networks. Kruk [165] proposed Friend-of-a-friend (FOAF)-Realm, an ontology-based access control mechanism. FOAF uses RDF (*Resource Description Framework*) to describe relations among users. The D-FOAF system [166] is a FOAF ontology-based distributed access control management system for social networks, where information inherent in social networks is used to provide community-driven access rights delegation. Both systems use a generic definition of relationships ("knows") as a trust metric and generates rules that control a friend's access to resources based on the degree of separation in the social network. This approach that uses the degree of separation as the only way to quantify the level of relationship between two users ignores the relationship type. Choi et al. [53] considered named relationships (e.g., worksWith, isFriendOf, knowsOf) in modeling trust. A more nuanced trust-related access control model is proposed by Carminati et al. [47] based on relationship type, degree of separation, and trust levels among users in the network.

An inherent problem with trust-based privacy models is that the trust threshold values should be smoothed as much as possible. In practice, it is difficult to comprehend and specify appropriate trust thresholds without prior threshold value tuning experiments. Our approach, Aegis (Chapter 7), avoids this problem by not using trust (always difficult to define), but by capturing instead the information semantics using an ontology-based access control policy.

Semantic rule-based policies have also emerged as a promising choice to control access to users social data. Rule-based policies represent the social knowledge base in an ontology (e.g., OWL) and define policies as Semantic Web Rule Language (SWRL) rules[2]. Access request related authorization is provided by reasoning on the social knowledge base. Systems that leverage OWL and SWRL to provide rule-based access control framework are [46, 86]. Although conceptually similar, [46] provides richer OWL ontology and different types of policies; access control policy, admin policy and filtering policy. The practicality of these solutions is difficult to evaluate in the absence of a proof-of-concept implementation. A more detailed semantic rule-based model is [192], which also provides a proof-of-concept implementation.

Rule-based privacy models have several limitations. First, authorization is provided by forward reasoning on the whole knowledge base, challenging scalability with the size of the knowledge base. Second, all authorizations must be recomputed if a change occurs in the social knowledge base. And finally, rule management is complex and requires a team of expert administrators [88]. In our approach the social knowledge base can be easily distributed, such that a user's trusted peer handles the user-related social data requests (like in [158]). Furthermore, recomputation of all policies is not required in case of knowledge base changes.

Role and Relationship-Based Access Control (ReBAC) are other types of privacy models that employ roles and relationships in defining privacy policies. Fong [98] proposed a ReBAC

[2] http://www.w3.org/Submission/SWRL/

model based on the context-dependent nature of relationships in social networks. This model targets social networks that are poly-relational (e.g., teacher-student relationships are distinct from child-parent relationships), directed (e.g., teacher-student relationships are distinct from student-teacher relationships) and tracks multiple access contexts that are organized into a tree-shaped hierarchy. When access is requested in a context, the relationships from all the ancestor contexts are combined with the relationships in the target access context to construct a network on which authorization decisions are made. Our work of Aegis is similar in that we also model relationships in a social context as the means to access and distribute social data. But our objective is different, as we do not target particular social networks, but generate default policies for aggregated social data that could be accessed by diverse social applications.

Giunchiglia et al. [108] proposed RelBac, another relation-based access control model to support sharing of data among large groups of users. The model defines permissions as relations between users and data, thus separating them from roles. The formalization of the RelBac model as an entity-relationship model allows for its direct translation into description logics, which also allows reasoning. The model, however, does not provide any precise social aspect and lacks auto generation of default policies.

The work conceptually closest to our work Aegis is PriMa [257]. PriMa also auto generates access control policies for users, acknowledging the fact that due to growing complexity and diversity of user content, it is difficult for regular users to manually configure their access control settings. The policies in PriMa are generated based on intuitive factors such as average privacy preference of similar and related users, accessibility of similar items in similar and related users, the popularity of the owner (i.e., popular users have sensitive profile items), closeness of owner and accessor (such as the number of mutual friends), etc. Access control policies for profile items are finally generated aggregating these factors. This approach is vulnerable to highly volatile policies due to changes in these factors. Moreover, a large number of factors and their parametrized tuning contribute to longer policy generation and enforcement time. Unfortunately, these limitations are not addressed, so it is difficult to judge their impact in practice. Another auto-generated policy framework is PolicyMgr [248]. PolicyMgr is based on supervised machine learning techniques. It uses user-contributed example policy configurations as training sets and builds classifiers to predict auto-generated policies for users' profile objects. Again, its practicality in terms of response time has not yet been shown.

Our privacy model differs from the above solutions with the focus on generating default policies for a social ecosystem that deals with users' aggregated social data from different domains; the existing solutions target single application scenarios. Moreover, most of those solutions do not take target default policy generation as a primary goal. Furthermore, to the best of our knowledge, we are the first to consider a privacy framework proposed by social theorists and translate it into an architecture and proof-of-concept implementation.

CHAPTER 3: DATASET: YAHOO ANSWERS[1]

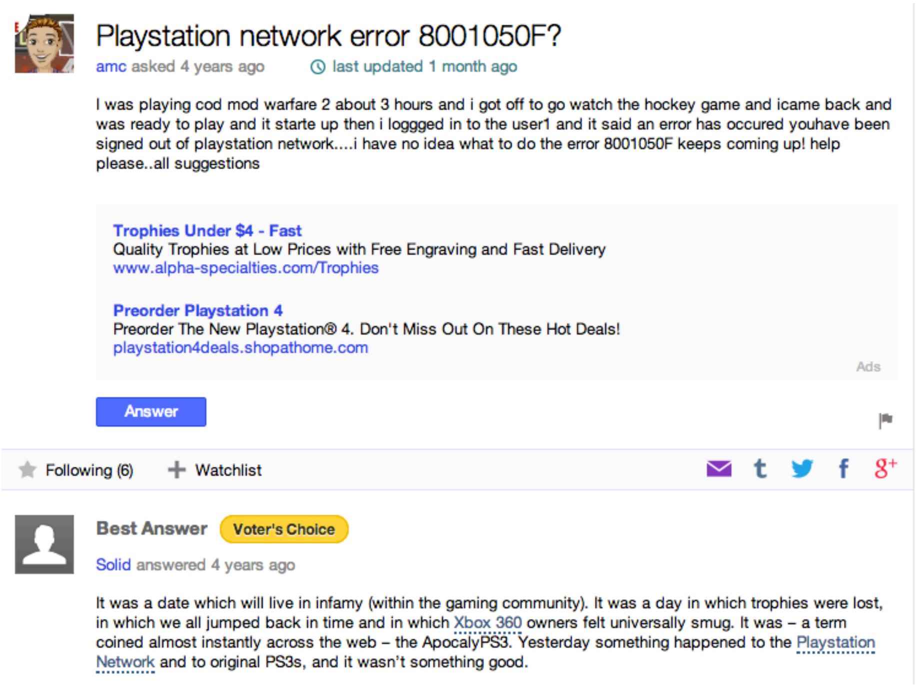
CQA platforms like Yahoo Answers, Quora and Stack Overflow have become rich and mature repositories of user-contributed questions and answers over the past decade. For example, Yahoo Answers, launched in December 2005, has more than one billion posted answers [195]. Quora, one of the fastest growing CQA sites, has seen three times growth in 2013 [277]. A study on Yahoo Answers revealed that about 2% of web searches performed by Yahoo Answers users lead to a question posted to the community [183]. These CQA platforms have a social networking component, where users follow each other based on common interests. Given the social network component, access control mechanisms and the popularity and specificity of the content (e.g., questions and answers), CQAs are suitable platforms for studying content abuse, privacy and culture at scale.

In this dissertation, we use a dataset from Yahoo Answers. After 10 years of activity, Yahoo Answers has 56M monthly visitors from the U.S. only [171]. The functionalities of the *YA* platform and the dataset used in this analysis are presented next.

3.1 The Platform

YA is a CQA platform in which community members ask and answer questions on various topics. Users ask questions and assign them to categories selected from a predefined taxonomy, e.g., *Business & Finance*, *Health*, and *Politics & Government*. Users can find questions by searching or browsing through this hierarchy of categories. A question has a title (typically, a short summary of the question), and a body with additional details.

[1] Much of the work in this chapter was first published in [150]. Permission is included in Appendix A.



Playstation network error 8001050F?

amc asked 4 years ago last updated 1 month ago

I was playing cod mod warfare 2 about 3 hours and i got off to go watch the hockey game and icame back and was ready to play and it starte up then i loggged in to the user1 and it said an error has occured youhave been signed out of playstation network....i have no idea what to do the error 8001050F keeps coming up! help please..all suggestions

Best Answer Voter's Choice

Solid answered 4 years ago

It was a date which will live in infamy (within the gaming community). It was a day in which trophies were lost, in which we all jumped back in time and in which Xbox 360 owners felt universally smug. It was – a term coined almost instantly across the web – the ApocalyPS3. Yesterday something happened to the Playstation Network and to original PS3s, and it wasn't something good.

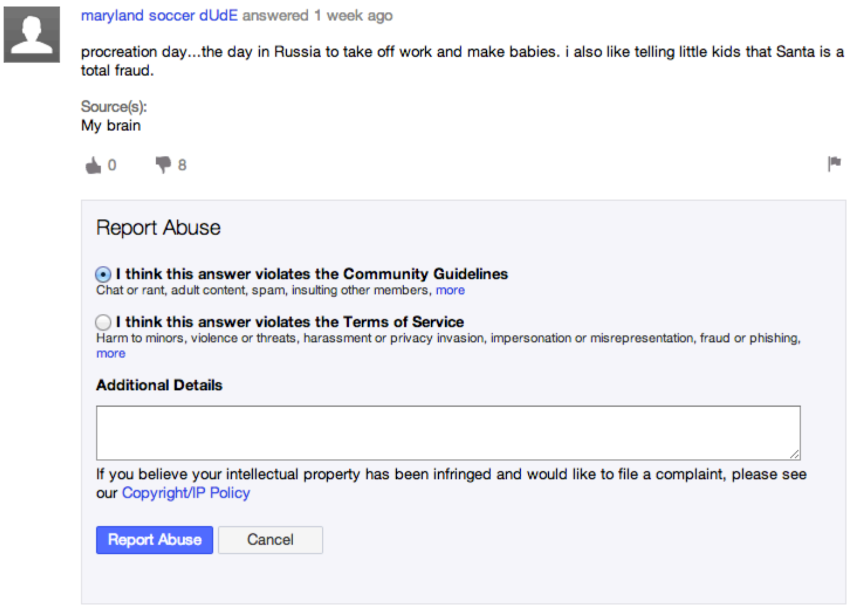
Figure 1 An answer (truncated and selected as best) for a question.

A user can answer any question but can post only one answer per question. Figure 1 shows an example of a question and an answer in *YA*. Questions remain open for four days for others to answer. However, the asker can select a best answer before the end of this 4-day period, which automatically *resolves* the question and archives it as a *reference* question. The best answer can also be rated between one to five, known as *answer rating*. If the asker does not choose a best answer, the community selects one through voting. The asker can extend the answering duration for an extra four days. The questions left unanswered after the allowed duration are deleted from the site. In addition to questions and answers, users can contribute comments to questions already answered and archived.

YA has a system of points and levels to encourage and reward participation [9]. A user is penalized five points for posting a question, but if she chooses a best answer for her question, three points are given back. A user who posts an answer receives two points; a best answer is worth 10 points.

A leaderboard, updated daily, ranks users based on the total number of points they collected. Users are split into seven levels based on their acquired points (e.g., 1-249 points: level 1, 250-999 points: level 2, ..., 25000+ points: level 7). These levels are used to limit user actions, such as posting questions, answers, comments, follows, and votes: e.g., first level users can ask 5 questions and provide 20 answers in a day.

YA requires its users to follow the Community Guidelines that forbids users to post spam, insults, or rants, and the Yahoo Terms of Service [8] that limits harm to minors, harassment, privacy invasion, impersonation and misrepresentation, and fraud and phishing. Users can flag content (questions, answers or comments) that violates the Community Guidelines and Terms of Service using the "Report Abuse" functionality. Figure 2 shows how reporting is done on an answer. Users click on a flag sign embedded with the content and choose a reason between violation of the community guidelines and violation of the terms of service. Reported content is then verified by human inspectors before it is deleted from the platform.



maryland soccer dUdE answered 1 week ago

procreation day...the day in Russia to take off work and make babies. i also like telling little kids that Santa is a total fraud.

Source(s):
My brain

Report Abuse

● **I think this answer violates the Community Guidelines**
Chat or rant, adult content, spam, insulting other members, more

○ **I think this answer violates the Terms of Service**
Harm to minors, violence or threats, harassment or privacy invasion, impersonation or misrepresentation, fraud or phishing, more

Additional Details

If you believe your intellectual property has been infringed and would like to file a complaint, please see our Copyright/IP Policy

Figure 2 Abuse reporting on an answer. Users can click on the "flag" sign of the answer and report an abuse.

Users in *YA* can choose to follow other users, thus creating a follower-followee relationship used for information dissemination. The followee's actions (e.g., questions, answers, ratings, votes, best answer, awards) are automatically posted on the follower's newsfeed. In addition, users can follow questions, in which case all responses are sent to the followers of that question.

3.2 Dataset

We studied a sample of 10 million abuse reports posted between 2012 and 2013 originating from 1.5 million active users. These users are connected via 2.6 million follower-followee relationships in a social network (referred to as FF in this study) that has 165,441 weakly connected components. The largest weakly connected component has 1.1M nodes (74.32% of the nodes) and 2.4M edges (91.37% of the edges). Out of the 1.5 million users, about 9% of the users have been suspended from the community.

Figure 3(a) and Figure 3(b) plot the complementary cumulative distribution function (CCDF) for the degree of followers (indegree) and followees (outdegree), respectively. The indegree and outdegree follow power-law distributions [21], with an exponential fitting parameter α 3.53 and 2.95 respectively.

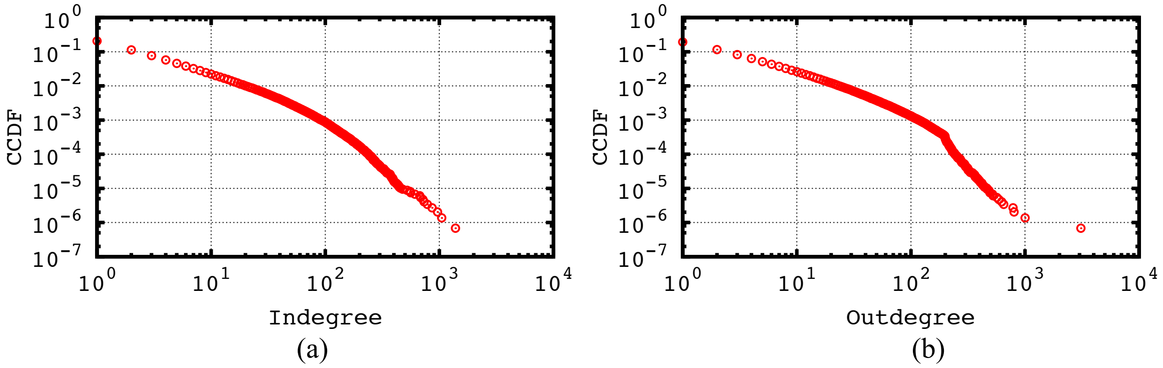


Figure 3 (a) Indegree distribution; (b) Outdegree distribution.

Along with the follower-followee social network, we built an activity network (AN) that connects users if they interacted with each other's content. In the *AN* network, nodes are users who answered other users' questions, directed edges point from the answerer to the asker, and edge weights show the number of answers provided over the source user to the questions posted by the destination user. The activity network has 1.2M nodes and 45M edges, thus being 141 times denser than the FF network.

CHAPTER 4: THE SOCIAL WORLD OF CONTENT ABUSERS IN COMMUNITY QUESTION ANSWERING SOCIAL NETWORKS[1]

This chapter investigates the reporting of rule violations in Yahoo Answers (*YA*), one of the oldest, largest, and most popular CQA platforms. We characterize the contribution and position of the content abusers in Yahoo Answers CQA social networks, and predict such users. In order to preserve the health and usefulness of online communities, CQA platforms define community rules and expect users to obey them. These rules are published as *community guidelines* and *terms of services*. To enforce community monitoring, CQAs provide users with tools to *flag* inappropriate content. Moreover, CQAs employ human moderators to evaluate the reported flags. Based on the report the moderator might keep or remove the content. In the worst case, users are suspended for persistent posting of abusive content.

Our dataset contains about 10 million editorially curated abuse reports posted between 2012 and 2013 (the dataset has been described in Chapter 3). Out of the 1.5 million users who submitted content during the one-year observation period, about 9% of the users got their accounts suspended. We use suspended accounts as a ground truth of bad behavior in *YA*, and we refer to these users as *content abusers* or *bad guys* interchangeably. The outcomes of this study could aid human moderators with automated tools in order to maintain the health of the community.

To understand the position and contribution of content abusers in Yahoo Answers, we raise four research questions:

[1] Much of the work in this chapter was first published in [150]. Permission is included in Appendix A.

R_1 : Are flags appropriate proxies for content abuse? If so, how can we use the flags to identify content abusers?

We discover that, although used correctly, flags do not tell accurately which users should be suspended: while 32% of the users active in our observation period have at least one flag, only 16% of them are suspended during this time. Even considering the top 1% users with the largest number of flags, only about 50% of them have their accounts suspended.

However, a high correlation between the number of content users post and flags they receive suggests that a portion of the flags a user receives might be simply the results of the high level of user activity. So, we use "deviance", a measure that indicates given the amount of activity a user has, how much she deviates from the norm of receiving flags, as a proxy of content abuse. We find 65% among the top 1% most deviant users are suspended. Given that deviance is a better metric than flags for identifying content abusers, we use the metric to understand the contribution of deviant users. We ask:

R_2 : Is deviance necessarily bad?

We find that, unlike in other environments where abusers are clearly the bad guys (e.g., cheaters in online games [56]), the situation is not black and white. That is, users flagged many times for rule violations contribute positively to the community by increasing user engagement and providing the best answers.

However, complicating an already complex problem, we find that 40% of the suspended users have not received any flags. To reduce this large gray area of questionable behavior, we employ social network analysis tools in an attempt to understand the position of content abusers in the *YA* community. We ask the following question:

R_3 : Does the follower-followee network impact behavior?

We learn that the follower-followee social network tunnels user attention not only in terms of generating answers to posted questions, but also in monitoring user behavior. More importantly, it turns out that this social network divulges information about the users who go under the community radar and never get flagged even if they seriously violate community rules. In the light of the insights from our analysis, we aim to classify fair and suspended users. We ask the following question:

R_4 : How can we build predictive models to automatically detect suspended users?

The network-based information, combined with user activity, leads to accurate detection of the bad guys: our classifier is able to distinguish between suspended and fair users with an accuracy as high as 83%.

The chapter is structured as follows. We introduce a deviance score in Section 4.1 that identifies the pool of abusive users more accurately than the number of flags alone. Section 4.2 demonstrates that deviant users are not all bad: despite their high deviance score, in aggregate their presence in the community is beneficial. Section 4.3 shows the effects of the social network on user contribution and behavior. Section 4.4 presents an analysis of the good guys, who voluntarily flag the abusive content. Section 4.5 shows the classification of suspended and fair users. We discuss the impact of these results in Section 4.6.

4.1 Flags in Yahoo Answers: a Proxy for Content Abuse?

In this section, we study whether flags (we use flags and abuse reports interchangeably) can be used as an appropriate proxy for content abuse. First, we investigate whether the flags reported from users are typically valid, i.e. if human inspectors remove the flagged content and further, how quickly this is done. Then, we explore how the flags can be used to detect content abusers.

4.1.1 Abuse Reports

YA is a self-moderating community; the health of the platform depends on community contributions in terms of reporting abuses. Besides participating by providing questions and answers, *YA* users also contribute to the platform by reporting abusive content. Reporters serve as an intermediate layer in the *YA* moderation process since these abuse reports are verified by human inspectors. If the report is valid, the content is promptly deleted.

To check if valid abuse reports are indeed an accurate sensor for the correct monitoring of the platform, we look at how soon a report is curated. Figure 4 shows the distributions of the time interval between the time when a content (question or answer) is posted and when it is deleted due to abuse reports. About 97% of questions and answers marked as abusive are deleted within the same day they are posted. All reported abusive questions and answers are deleted within three days of posting.

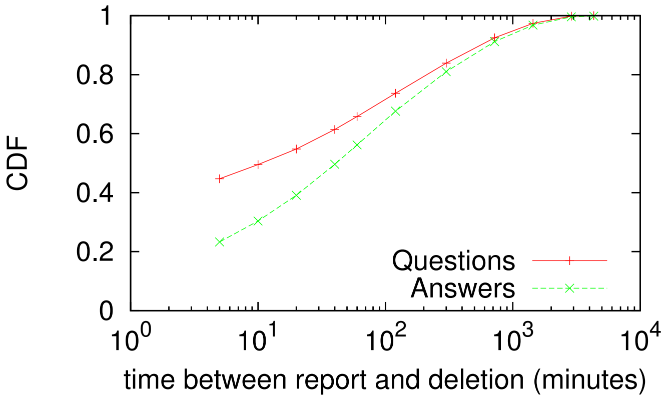


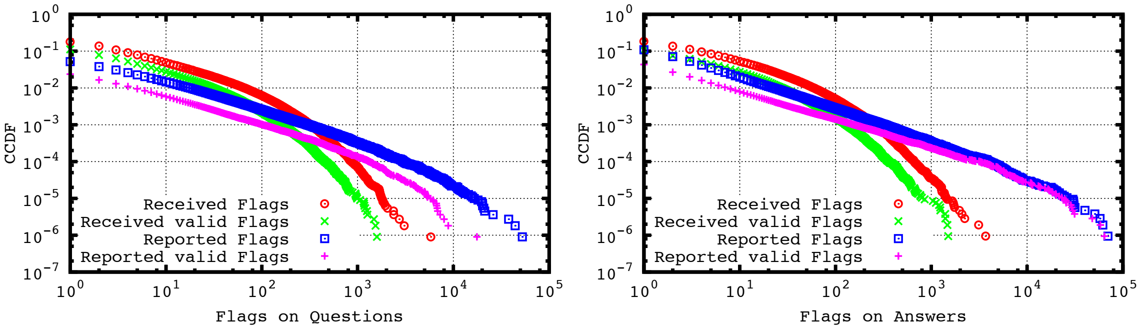
Figure 4 The CDF of the time delay between the posting of the content (questions or answers) and its deletion due to valid abuse reporting.

This result highlights two facts. First, that the users monitoring the platform act very quickly on content: within 10 minutes from being posted, 50% of the bad posts are reported. Second, that validation of abuse reports happens within 3 days (and in vast majority within a day). Hence, in our dataset, if there are abuse reports that did not have the chance of being curated yet and thus we do not consider them, those are too few to impact our analysis.

However, the abuse reporting functionality might be abused as well, due to several reasons. First, reporting is an easy and fast process, requiring only a few steps. Second, a user is not penalized for misreporting content abuse, perhaps in an attempt to not discourage users from exercising good citizenship. And third, independent of their level in the *YA* platform (that limits the number of questions and answers the user can post per day), users can report an unlimited number of abuses.

To check whether users abuse the abuse reporting functionality, we compare the number of flags received/reported with the number of validated flags received/reported per user.

Figure 5 shows the distributions of reported valid flags and received valid flags, along with reported flags and received flags for questions and answers. For both questions and answers, the distribution patterns are the same for flags reported and that are valid and also flags received and that are valid. Reported valid flags lies very closely below the reported flags. This is also the same for reported valid flags and reported flags. The distributions, thus indicate that there might be good correlations between flags received/reported and that are valid.

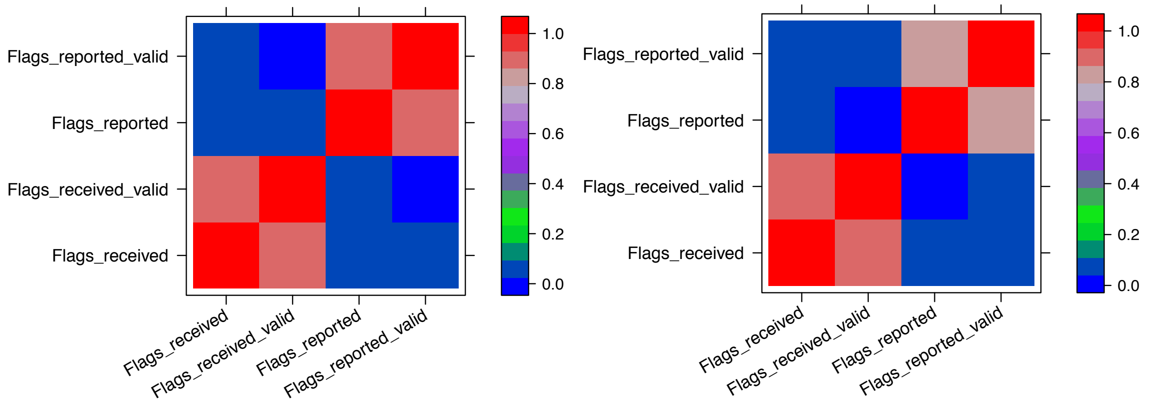


(a) (b)

Figure 5 Distributions of abuse reports (flags) on (a) questions; (b) answers.

Figure 6 shows correlation heat maps of the flags received, flags received valid, flags reported and flags reported valid on questions and answers for all contributors. For questions (answers), we have a very high correlation between flags received by users and flags that are valid ($r = 0.90$ (0.87), $p < 0.01$) and between flags reported by users and that are valid ($r = 0.80$ (0.92), $p < 0.01$).

These high correlations indicate that, in general, users are not exploiting the abuse reporting functionality. When a user reports an abuse, it is very likely that the content is violating community rules. Another interesting finding from the correlation heat maps is that for both questions and answers, users have almost negligible or very weak correlation between the number of flags they reported that are valid and the number of flags they received that are valid. This hints that the good guys of the community are not bad guys at the same time: the users who correctly report a lot of content abuses are not posting abusive content themselves.



(a)

(b)

Figure 6 The Pearson correlation coefficient heat map of flags received, flags received valid, flags reported and valid flags reported on (a) questions; (b) answers. All values are statistically significant (p-values <0.01).

4.1.2 Deviance Score

Given that flags are good proxies for identifying bad content, how should they be used to detect content abusers and thus determine which accounts to be suspended? Common wisdom might suggest that content abusers are those who receive a large number of flags. Of the top 1% flagged askers and answerers, we find 51.63% and 53.89%, respectively, are suspended. But finding a threshold on the number of flags received by a user is not likely to work accurately for content abuser detection: users with low activity who received flags for all their posts might go below this threshold. At the same time, highly active users may collect many flags even if for a small percentage of their posts, yet contribute significantly to the community.

This intuition motivated us to measure the correlation between a user's number of posts and the number of flags received. Indeed, we find that the correlation between the number of questions a user asks and the number of valid flags she receives from others is high ($r = 0.49$, $p < 0.05$). Similarly, the number of answers posted and the number of valid flags received per user are highly correlated ($r = 0.37$, $p < 0.05$). The distributions of the fraction of flagged questions and answers is shown in Figure 7. While about 27% users have more than 25% flagged questions, about 34% users have more than 25% flagged answers. Also, about 16% and 19% of users have more than 50% flagged questions and answers, respectively.

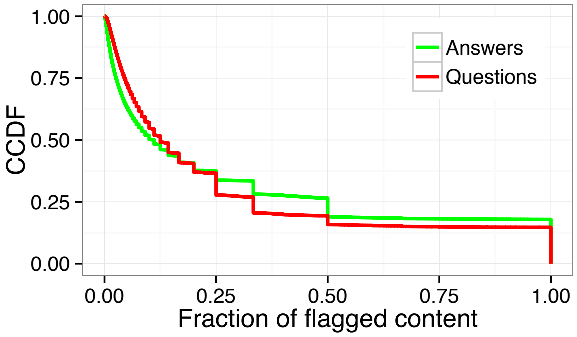


Figure 7 Distributions of fraction of flagged questions and answers.

So, instead of directly considering flags, we define a *deviance score* metric that indicates how much a user deviates from the norm in terms of received flags considering the amount of activity. Deviant behavior is defined by actions or behaviors that are contrary to the dominant norms of the society [78]. Although social norms differ from culture to culture, within a context, they remain the same and they are the rules by which the members of the community are conventionally guided.

We define the deviance score for a user u as the number of correct abuse reports (flags) she receives over the total content (question/answer) she posted, after eliminating the expected average number of correct abuse reports given the amount of content posted:

$$\text{Deviance}_{\text{Q/A}}(u) = Y_{Q/A,u} - \hat{Y}_{Q/A,u} \tag{4.1}$$

where $Y_{Q/A,u}$ is the number of correct abuse reports received by u for her questions/answers, and $\hat{Y}_{Q/A,u}$ is the expected number of correct abuse reports to be received by u for those questions/answers.

To capture the expected number of the correct abuse reports a user receives for questions/answers, we considered a number of linear and polynomial regression models between the response variable (number of correct abuse reports) and the predictor variable (number of questions/answers). Among them, the following linear model was the best in explaining the variability of the response variable.

$$Y = \alpha + \beta X + \epsilon \tag{4.2}$$

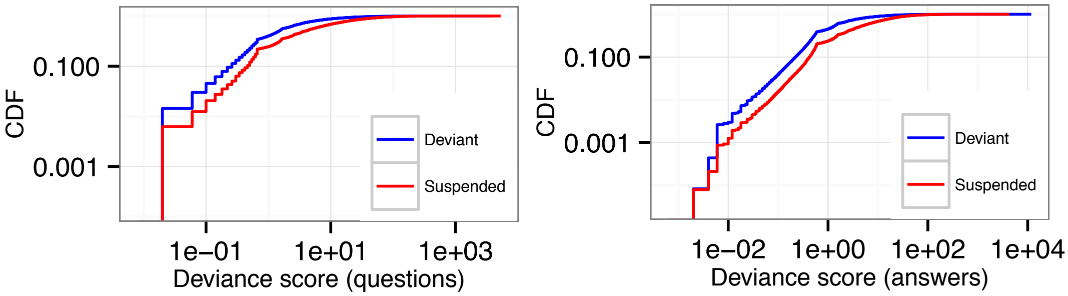
where Y is the number of correct abuse reports (flags) received for the content, X is the number of content posts and ϵ is the error term.

In eq. (4.1), a positive deviance score reflects deviant users, i.e., those whose deviance cannot be only explained by their activity levels.

4.1.3 Deviance Score vs. Suspension

We have found $105,340$ users with positive *question* deviance scores and $121,705$ users with positive *answer* deviance scores. Among the users with positive question deviance score, $31,891$ users (30.27%) have been suspended. Similarly, among the users with a positive answer deviance score, $37,633$ users (30.92%) have been suspended. The CDF of suspended and deviant (but not suspended) users' deviance scores for both questions and answers is shown in Figure 8. In both cases, suspended and deviant users are visibly characterized by different distributions:

suspended users tend to have higher deviance scores than deviant (not suspended) users. While this difference is visually apparent, we also ensure it is statistically significant using two methods: 1) the two-sample Kolmogorov-Smirnov (KS) test, and 2) a permutation test, to verify that the two samples are drawn from different probability distributions.

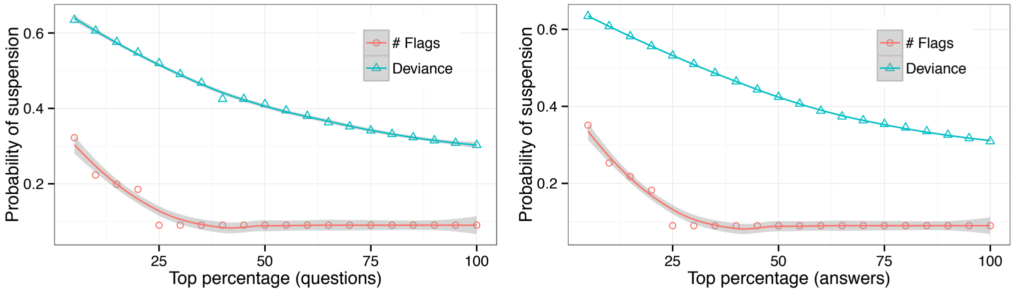


(a) (b)

Figure 8 The CDF of suspended and deviant users' deviance scores for (a) questions; (b) answers. Distributions are different with $p < 0.001$ for both KS and permutation tests. For questions: $D = 0.22$, $Z = 46.04$. For answers: $D = 0.28$, $Z = 50.53$.

We also find that 63.94% of top 1% deviant question askers' and 64.77% of top 1% deviant answerers' accounts have been suspended. This hints that the higher deviance score a user has, the more likely (s)he is to be removed from the community. Figure 9 shows the probability of a user being suspended as a function of its rank in the community as expressed by deviance score and number of flags. We observe that the more deviant a user is, the more probable is that she will be suspended. Also, in all cases, deviance score shows a higher probability of suspension compared to the number of flags.

These results show that the deviance score is a better metric for identifying the content abusers than the number of flags is by itself. However, both metrics fail to identify content abusers who go under the community radar. We found that about 40% of the suspended users had never been flagged for the abusive content they certainly posted, thus maintaining a negative deviance score. Thus, our investigation into user behavior in the *YA* community continues.



(a) (b)

Figure 9 Probability of being suspended, given a user is within top $x\%$ of (a) question or (b) answer deviance scores and flags. Local polynomial regression fitting with 95% confidence interval area is also shown.

4.2 Deviant vs. Suspended Users

Despite the fact that deviance score better identifies the pool of suspended users, it is clearly an imperfect metric. On one hand, there are high deviance score users who are not suspended, despite the fact that the platform seems to be fairly quick in responding to abuse reports. On the other hand, there are “ordinary” users, according to the deviance score (i.e., with a negative deviance score) who are never reported for abusive content, yet get suspended. To better understand these two groups of users—deviant but not suspended and suspended but not flagged—we analyze in more detail their activity. Note that the two groups are disjoint (i.e., deviant users have received at least one flag).

4.2.1 Deviance is Engaging

One of the success metrics of CQA platforms is *user engagement* [191], which can be measured by the number of contributions and by the number of users who respond to a particular content. Thus, we use the number of answers deviant users receive to their questions and the number of distinct users who respond to the deviant users’ questions as measures of deviants’ contribution to user engagement with the platform. To this end, for each category of users (typical, deviant but not suspended, and suspended) we randomly selected $500k$ questions they asked.

Table 1 Descriptive statistics of the number of answers received by typical, deviant but not suspended, and suspended users per question.

Type	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.
Typical	1.00	1.00	2.00	4.36	5.00	1296.00
Deviant	1.00	5.00	11.00	17.96	22.00	1205.00
Suspended	1.00	1.00	4.00	8.67	9.00	1144.00

For each question, we extracted all answers received and also the users who answered those questions. Table 1 presents the statistics of the number of answers received per category of users.

Deviant users' questions get significantly more answers than typical users's questions get: on average, a question posted by a deviant user gets about 5 times more answers than the average question posted by a typical user. This difference is also seen in the CCDF of the number of answers received by typical, deviant and suspended users in Figure 10(a). The distributions (pairwise) are different with $p_{ks} < 0.01$ and $p_{perm} < 0.01$.

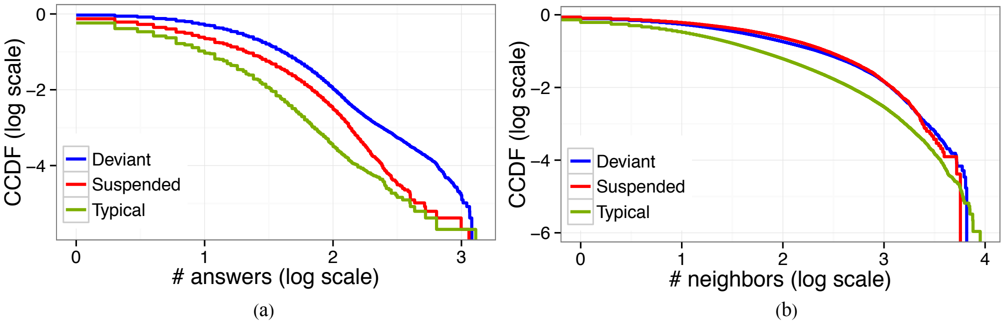


Figure 10 (a) CCDF of the number of answers received by the typical, deviant but not suspended, and suspended users on questions; (b) CCDF of the number of neighbors (distinct answerers) that typical, deviant but not suspended, and suspended users have.

Deviant users not only attract more answers, but also interact with more users than typical users do, as shown by Figure 10(b) and these two distributions are different ($p_{ks} < 0.01$, $p_{perm} < 0.01$).

This result from analyzing a random sample of $500k$ questions is confirmed when looking at the indegree of nodes in the activity network, which represents the number of users who answered that node's questions, as shown in Table 2 for typical and deviant users. Deviant askers

Table 2 Descriptive statistics of the number of neighbors askers have in the Activity Network.

Type	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.
Typical	0.00	1.00	5.00	28.16	19.00	13270.00
Deviant	0.00	3.00	20.00	103.40	90.00	5698.00
Suspended	0.00	2.00	13.00	88.62	60.00	6576.00

have a higher number of neighbors than typical askers. An explanation might be, as shown in [122], that users who ask conversational questions tend to have more neighbors (with whom the asker has interaction) than users who ask informational questions. This suggests that deviant users tend to ask more conversational questions, which engage a larger number of responders.

4.2.2 Deviance is Noisy

We observed that deviant users impact the *quantity* of content in the system. Do they impact *quality*, too? To address this question, we look at the percentage of the best answers with respect to the total number of answers submitted per user.

Figure 11 shows the CDF of the percentage of best answers for different classes of users: 1) typical, 2) deviant but not suspended, and 3) suspended. The results show that users who are moderately deviant but did not get suspended have higher percentage of best answers than suspended users (distributions are different $p_{ks} < 0.01$, $p_{perm} < 0.01$), but lower than that of typical users (distributions are different $p_{ks} < 0.01$, $p_{perm} < 0.01$).

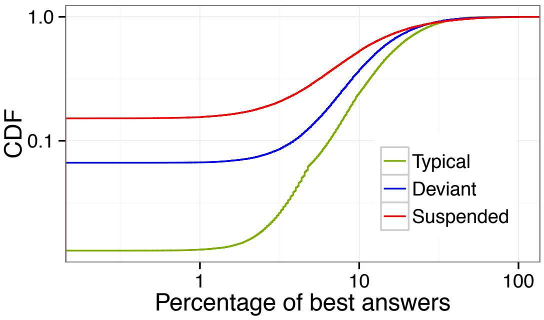


Figure 11 CDF of the ratio of best answers for typical, deviant but not suspended and suspended users.

To conclude, it turns out that while deviant users are beneficial in terms of platform success metrics, as they increase user engagement by attracting more answers and attracting more users who answer their questions, they do not contribute more than the norm-following users in terms of content quality.

4.2.3 Deviance is Everywhere

Deviance is culturally determined and the perception of deviance changes from culture to culture [80]. We investigate to what extent deviance is moderated across various languages and how diverse the abuse reporters are.

Based on users' question and answer deviance scores, we took 1% of the top deviant users and randomly selected 10% of their questions and answers in each category, respectively. We also took the same number of top fair users (negative deviance scores and not suspended from the community) in both categories and took the same number of randomly selected questions and answers. We used a language identification package in Python, LangID[2], to detect the languages in which these questions and answers are posted. LangID is a pre-trained language identification classifier and it is not sensitive to domain-specific features such as HTML/XML markup.

As shown in Figures 12(a) and (b), English is the most dominant language in *YA* and languages differ in terms of the percentage of questions and answers contributed by top deviant and fair users. Top fair users in English language contributed most content (49.27% questions, 65.31% answers) among all content contributed by top fair users in different languages. Also, top deviant users in English language contributed 16.10% questions and 49.24% answers among all content contributed by top deviant users in different languages.

Given the differences of fair users' contributions in different languages, we asked whether some languages (e.g., English) enjoy better moderation than others. To answer that, we used top deviant users who have been already suspended from the *YA* community as described in Sec-

[2] https://github.com/saffsd/langid.py

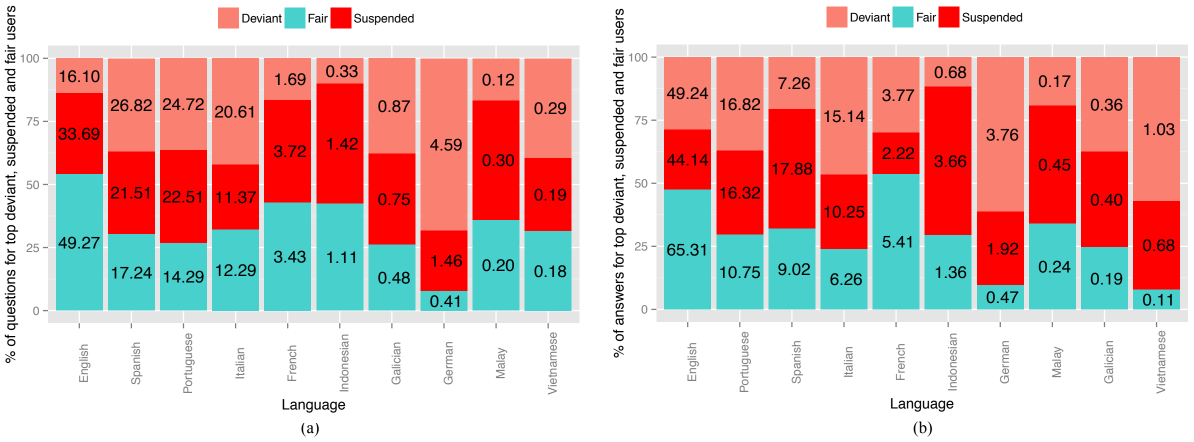


Figure 12 Percentage of (a) questions and (b) answers contributed by top deviant, fair and suspended users in different languages. Numbers inside a bar show the percentage of questions (answers) that the specific language's different types of users contributed among all languages. A bar represents the percentage of questions (answers) different types of users contributed in that language.

tion 4.1. Using the same procedure as before, we compared their languages in questions/answers with fair and deviant users' questions/answers (we refer to them as suspended users in Figure 12). Suspended users in English language contributed 33.69% questions and 44.14% answers among all contents contributed by suspended users in different languages. In English language, we see more contribution from top fair users than top deviant and suspended users. Although in our sample, deviant and suspended users' combined contribution is twice than fair users, in English we see that about 50% of questions and answers are from top fair users, where the rest are from deviant and suspended users. However, this is not a trend in all of languages. In some languages, top deviant and suspended users have higher contribution than top fair users, e.g., German and Vietnamese for questions and answers. Moreover, the presence of suspended users in various languages shows that deviance is moderated across languages.

To assess the diversity of abuse reporters in different languages, we map abuse reports to reporter location. We used unique IP addresses of the reporters to get the number of distinct reporters. Using an API from HostIP [131], we convert IP addresses to locations (cities and countries). Figure 13 shows the percentage of abuse reporters from different countries normalized by their Internet populations [19]. We find that users have reported from 197 countries and territories. The largest number of reporters (25.57%) is from the USA. The number of reporters is high from American countries (both north and south), Western Europe, Southeast Asia and Australia. These top countries represent all the languages in Figure 12, i.e., USA, Canada, Australia (English), Western Europe (Spanish, Portuguese, Italian, French, German, Galician), Southeast Asia (Malay, Vietnamese).

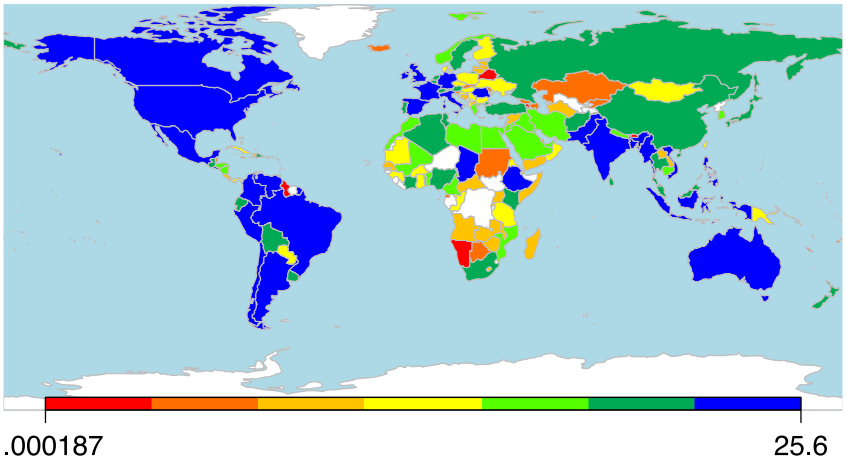


Figure 13 Locations of abuse reporters.

4.2.4 Centrality of Suspended Users

In order to understand structural positions of the suspended users in *YA*, we study their centralities in the *FF* network. Structural positions of the nodes are often correlated with their characteristics and behaviors [10, 11]. For example, most central users in a competition-based expertise network of *YA* are found to be an expert and knowledgeable answerers [13]. We study centrality measures belonging to two main families: *degree*-based centrality (*Indegree*, *Outdegree*) and *spectral* centrality (*PageRank*). We do not use *path*-based centrality, such as *betweenness* or *closeness*, due to their computational complexity in large datasets. Indegree and

Table 3 Percentage of the suspended users found in Top-N% of high centrality users in Yahoo Answers.

Top-N%	0.1	0.5	1.0	5.0
Indegree	0.30	1.43	2.71	10.61
Outdegree	0.28	1.31	2.34	8.13
PageRank	0.25	1.34	2.50	9.42

Outdegree centralities are based on the number of followers and followees respectively, i.e., most indegree central users are the nodes with the highest number of followers. However, the importance of the neighboring nodes is also crucial for a node to be central. PageRank centrality considers this impact of neighboring nodes and assigns numerical weights to each node of a network based on its relative importance within the network.

We compute the most central users in the network and analyze how many of them are suspended. The percentage of the suspended users found in Top-N% of high centrality users are shown in Table 3. Although suspended users are 9% of the population, they are only 0.30%, 0.28% and 0.25% within the top 0.1% Indegree, Outdegree and PageRank central users respectively. The under-representation of suspended users doesn't change if we increase top centrality users from 0.1% to 5.0% gradually. These results suggest that on average, suspended users are not central in *YA* and hence, their potential influence should be limited.

4.2.5 The Suspended but Not Flagged Users

While the results already presented show how the deviant users differ from the suspended and from the typical users, we do not have yet an understanding of the behavior of the users who get suspended without other users flagging their abusive content. An initial analysis of these users—suspended but not flagged—shows the following particularities when compared to the fair users (all users, independent of their deviance status, who are not suspended).

First, they are followed by and follow significantly fewer other users. Figures 14 (a) and (b) show the distributions of indegree and outdegree of never-flagged-suspended users compared to those of fair users. Not only these users have smaller social circles, but they also have lower activity levels, as shown in Figure 14 (c). Of course, these results could be correlated: low activity may mean low engagement in the social platform. These results may also suggest that (some of) these users join the platform for particular objectives that are orthogonal to the platform purpose, such as spamming. More importantly, however, these results suggest directions that we present in the following.

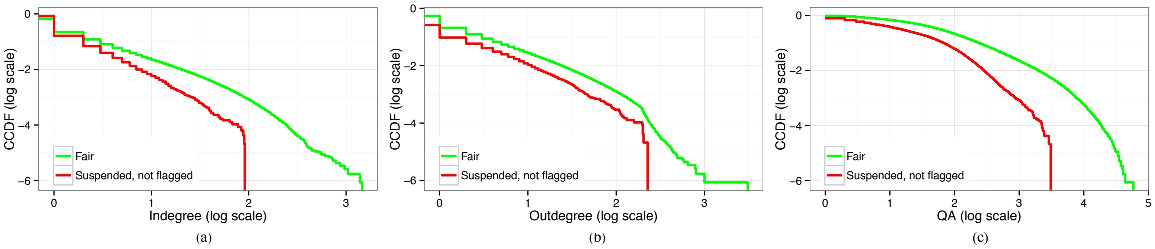


Figure 14 Distributions of (a) indegree; (b) outdegree and (c) number of questions and answers (QA) of never flagged suspended users and fair users. For outdegree: $D = 0.28$ and $Z = 27.40$, $p < 0.001$. For indegree: $D = 0.17$ and $Z = 15.86$, $p < 0.001$. For activity: $D = 0.30$ and $Z = 40.30$, $p < 0.001$.

4.3 Social Network Analysis

We investigate how the social network defined by the follower-followee relationships impacts user activities and behaviors in *YA*. Our final goal is to understand how to separate fair users from users who should be suspended even in the absence of flags. We learn that users close in the *FF* network not only help each others by answering questions, but also monitor each other's behavior by reporting flags (Section 4.3.1). Thus, the social network allows users to implicitly coordinate their behavior so much so that users who are socially close exhibit not only similar behavior, but also a similar deviation from the typical behavior (Section 4.3.2).

4.3.1 Out of Sight, Out of Mind

We expect that users receive more answers from users that are close in the social network. To verify this intuition, we randomly selected 7M answers such that both parties of the dialogue (the user who posted the question and the user who answered it) are in the social network, and measured the social distances between the two users. For a user u and a social distance h, the probability of receiving an answer from followers at distance h is the following:

$$p_h = \frac{\text{\# of } u\text{'s followers at distance } h \text{ who answered } u\text{'s questions}}{\text{\# of } u\text{'s followers at distance } h} \tag{4.3}$$

Figure 15 plots the geometric average of all these probabilities at a given distance as a function of social distance. The figure confirms that the probability of receiving answers from h-hop followers decreases with social distance.

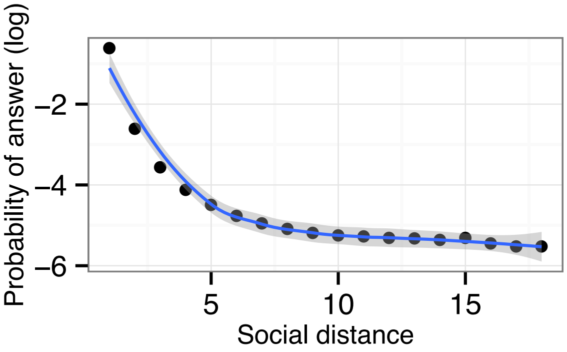


Figure 15 Probability of getting answers from h-hop followers. Local polynomial regression fitting with 95% confidence interval area is also shown.

Therefore, the *FF* network channels user attention, likely via its newsfeeds feature that sends updates to followers on the questions posted by the user. Does the same phenomenon hold true for abuse reports?

To answer this question we investigate both networks: along with the *FF* which is an explicit network, we also investigate the activity network (*AN*), which connects users based on their direct interactions question-answer. For each (reporter, reportee) pair in the editorially-curated abuse reports, we calculated the shortest path distance between them in the social network and the activity network. We compare our results with a null model that randomly assigns the abuse reports in our sample dataset to users in the two networks.

Figure 16 shows the percentage of abuse reports users receive from close distances (up to 8 hops) for both (social and random) cases. About 75% of the reports that users receive are from reporters located within 5 social hops in the *FF* network. However, when reports are distributed randomly, about 9% are from within 5 social hops and very few from within 3 social hops.

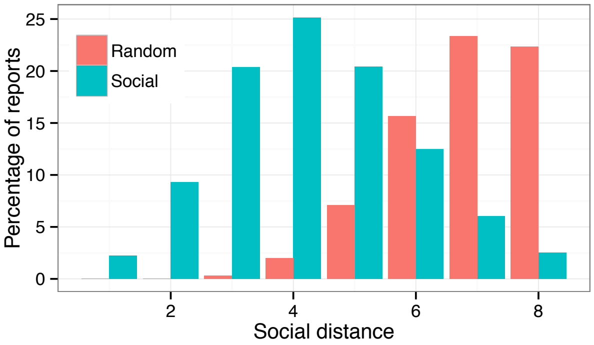


Figure 16 Percentage of the abuse reports received by users from different distances in the social network, for the observed case and a random case.

When comparing the percentage of abuse reports users receive with respect to distance in the *AN* (Figure 17), we notice that 94% of reports come from users within the first 3 hops, which is significantly higher than the social network (about 32%). We believe this is due to the high density of *AN*: most of the nodes are reachable from others within a few hops. However, even in this denser network, the null model has only about 10% of reports applied from within 3 hops.

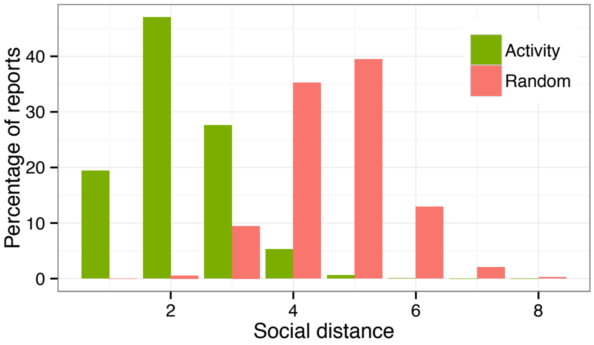
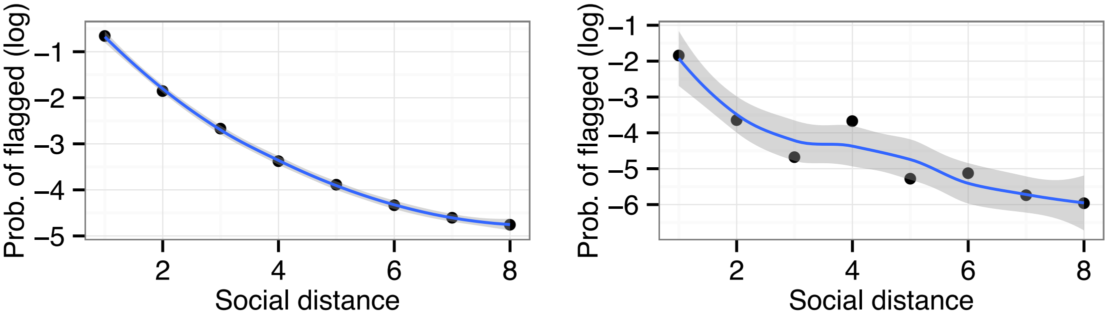


Figure 17 Percentage of the abuse reports received by users from different distances in the activity network, for the observed case and a random case.

To further quantify this phenomenon, we calculate the probability of being correctly flagged by users located at different network distances in the social and the activity network. For a user u and a social distance h, the probability of being flagged by followers at distance h is the following:

$$p_h = \frac{\text{\# of } u\text{'s followers at distance } h \text{ who flagged } u}{\text{\# of } u\text{'s followers at distance } h} \tag{4.4}$$

Figure 18 plots the geometric average of all probabilities at a given distance against the social distance for both networks. As expected, the probability decreases with social distances in both the social and the activity network. The plot shows that users are likely to receive flags from others close to them in terms of social relationships and interactions.



(a)

(b)

Figure 18 Probability of being flagged by h-hop followers (a) social network; (b) activity network. Local polynomial regression fitting with 95% confidence interval area is also shown.

These results confirm that the abuse reporting behavior is dominated by social relationships and interactions: users are reported for content abuse more from their close social or activity neighborhoods than from distant users. The underlying reason is likely content exposure: a user's contents (questions/answers) are disseminated to nearby followers, thus they get higher exposure to that content compared to more distant users in the social graph. Similarly, users who interact frequently with a user are more probable to view her contents and to report the inappropriate ones.

4.3.2 Birds of a Feather Flock Together

Similarity fosters connection– a principle commonly known as homophily, coined by sociologists in the 1950s. Homophily is our inexorable tendency to link up with other individuals similar to us [196]. We verify in this section whether homophily is also present in terms of deviance–that is, if deviant users tend to be close to each other in the social network.

One way to conclude about the homophily of a network is to compute the attribute assortativity of the network [213]. The assortativity coefficient is a measure of the likelihood for nodes with similar attributes to connect to each others. The assortativity coefficient ranges between -1 and 1; a positive assortativity means that nodes tend to connect to nodes of similar attribute value, while a negative assortativity means that nodes are likely to connect to nodes with very different attribute value from their own. If a network has positive assortativity coefficient, then it is often called assortative mixed by the attribute, otherwise called disassortative mixed.

In this work, we used question and answer-based deviant scores. We considered each of the scores as an attribute and calculated the assortativity coefficient r based on [214] for each type of deviance. The assortativity coefficients r are shown in Table 4 and are positive.

In [214], Newman studied a wide variety of networks and concluded that social networks are often assortatively mixed (Table 4), but that technological and biological networks (e.g., World Wide Web $r = -0.067$, software dependencies $r = -0.016$, protein interactions $r = -0.156$) tend to be disassortative. Comparing them quantitatively with the assortativity coefficients of the *YA* network, we conclude that the *YA* network is assortatively mixed in terms of deviance. So, users having contacts with (low)high deviance scores will also have (lower)higher deviance scores.

We next measure how similar the deviance scores of a user's contacts are with the user's, and how this similarity varies over longer social distances. For this, we randomly sampled $100,000$ users from the social network for each social distance ranking from 1 hop to 4 hops.

Table 4 Assortativity coefficient r for deviance scores in the *YA* network. r values are also shown for other social networks from [214].

Yahoo! Answers	Other Social Networks
Question deviance $r = +0.11$	Mathematics coauthorship $r = +0.120$
Answer deviance $r = +0.13$	Biology coauthorship $r = +0.127$

Let U_h be the set of all the users (100,000) selected for the social distance h. We calculated the probability that user u's h-hop contacts (with $u \in U_h$) will have the same deviance score as:

$$p_u = \frac{\text{\# of } u\text{'s followers at distance } h \text{ with same deviance score}}{\text{\# of } u\text{'s followers at distance } h} \tag{4.5}$$

Rather than computing the exact similarity between a user and her follower's deviance scores, we focused on whether their difference is small enough to be dubbed as the same. We considered two users' deviance scores are the same if their corresponding deviance score difference is less than a "similarity delta". More specifically, u will have *about the same* deviance score with user s located at distance h if:

$$|deviance_u - deviance_s| < \delta \tag{4.6}$$

The same technique was used for both types of deviance scores.We experimented with two values for δ equal to one or two standard deviations of the distribution of deviance scores in the network. We report the geometric average of all p_u probabilities computed in each hop h.

Figure 19 shows the probability plots for both types of deviance, keeping similarity δ equal to one or two standard deviations. Although different values of the δ, the shapes of the figures are almost the same: up to 3-hops, the probability decreases gradually with the social distance.

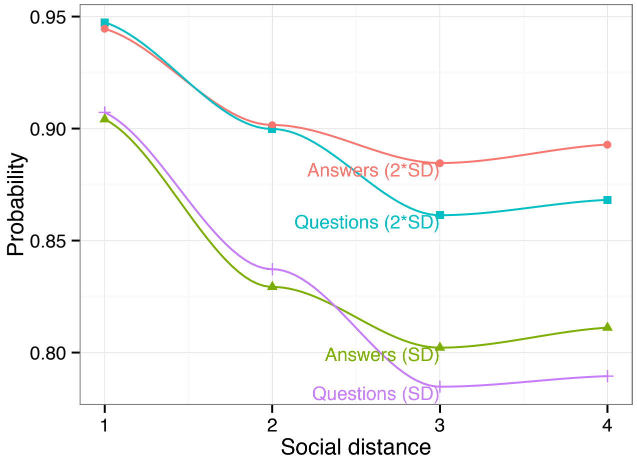


Figure 19 Probability that a h-hop follower has the same deviant score to the user for $\delta = \sigma$ and $\delta = 2\sigma$. SD: standard deviation.

4.4 The Good Guys

Crowd-sourced monitoring of the CQA platforms is facilitated by the *Good guys*. These users voluntarily flag the abusive content, even without getting any incentive from the platform. The flags are overwhelmingly correct, as seen by our previous results from Section 4.1. In this section, we ask several questions about good guys to better understand their behavior. More specifically, we ask (1) Are good guys few in numbers, or many? (2) Do good guys report consistently? (3) Do good guys focus on specific categories or do they report on diverse categories? (4) Are the good guys also good content contributors?

To answer the questions above, we first define a goodness score of the good guys by considering the correct and incorrect flags that they reported. Intuitively, the good guys are likely to report a high number of valid reports, i.e., reports that lead to content deletion after human inspection. As such, we define the goodness score of user u as follows.

$$Goodness_u = \#Valid_u - \#Invalid_u \tag{4.7}$$

where $\#Valid_u$ and $\#Invalid_u$ are the number of valid and invalid abuse reports, respectively, submitted by u.

We first look at the distribution of abuse reports. We find that although a large number of users interact with the platform by asking, answering questions or creating social connections (our dataset has 1.5M active users), only about 20% of users report abuses. This might be due to the fact that reporting has no incentives, e.g., reporters do not get any points in *YA*. The distribution of the number of abuse reports is shown in Figure 20. This distribution is highly skewed: most of the users have reported few abuses. About 46% of the users among the abuse reporters reported only one abuse, and 10% of users reported more than 13. Overall, 60% of abuse reports are contributed by only 932 users and 90% of abuse reports are contributed by 21,992 of such users (7.96% of users).

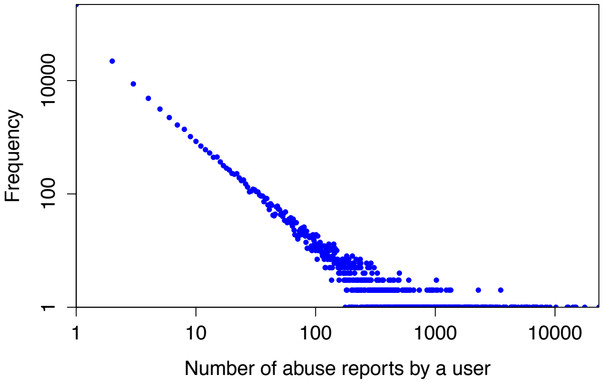


Figure 20 Distribution of the reports.

To answer whether good guys report on a regular basis, we calculate their temporal predictability of reporting. We use *entropy* [58] to calculate the temporal predictability. In classical thermodynamics, entropy is a measure of disorder. For a given report and T intervals (days), we can compute $p(r_t)$, the probability that the report belongs to an interval $t \in T$. We measure the normalized entropy for a user for all reports as:

$$\text{Report Entropy} = -\frac{\sum_{t \in T} p(r_t) log(p(r_t))}{|logT|} \tag{4.8}$$

This normalized report entropy ranges from 0 to 1 and it defines how consistent users are in reporting daily over the observation period. A normalized report entropy close to 1 indicates that the user has reported in each day of the observation period. Figure 21 shows CCDF of report entropies. We find that, in general, the good guys are regular in reporting. About 22% of the good guys have entropies at least 0.35. It appears that few good guys, about 0.3%, have taken reporting very seriously. Their entropy is at least 0.70, which means that they have reported almost 70% of the days of a year. The correlation between goodness scores and entropies is strong (r=0.67, p<0.001), which suggests that the better the contribution in reporting, the higher the consistency of reporting over time.

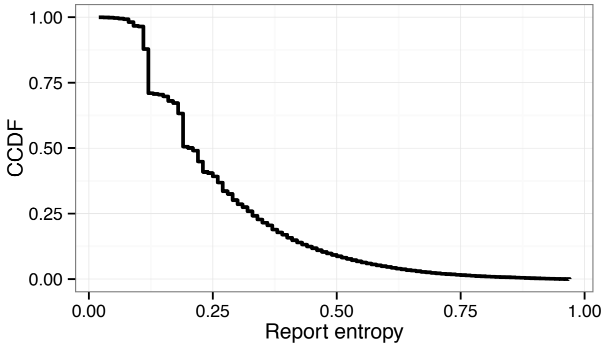


Figure 21 Distributions of report entropy.

Now we turn to answering how focused the good guys are at reporting, i.e., do they report on different topics (e.g., different *YA* categories)? Similar to equation 4.8, we calculate entropies of their reports on different categories over the observation period. This category entropy defines how consistent users are in reporting on different categories. For a given report and C categories, we can compute $p(r_c)$, the probability that the user has posted a report on a question or answer from the category $c \in C$. We measure the normalized entropy for a user for all reports as:

$$\text{Category Entropy} = -\frac{\sum_{c \in C} p(r_c) log(p(r_c))}{|logC|} \tag{4.9}$$

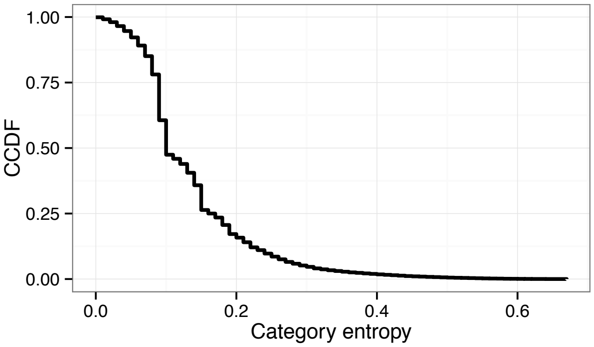


Figure 22 Distributions of category entropy.

Figure 22 shows the CCDF of category entropies. Given the high number (more than 1300) of categories in *YA*, in which users ask and answers questions, it will be unrealistic to assume that some guys will report across all categories (in this case their entropy will be 1.00). However, the distribution shows that about 8% of the good guys have entropies at least 0.25, meaning that they have reported on at least 25% of the categories. Some users, although very few (0.02%), have entropies at least 50%: these guys reported on half of the categories. The highest entropy that we have found is 0.67. In general, users report on different categories; only 7.77% users have entropy less than 0.05. The correlation between goodness scores and entropies (r=0.38, p<0.001) shows that the higher goodness score a user as, the more category entropy she has.

We finally want to understand the content contribution of good guys compared to typical users (who do not report) in the platform. We plot CCDF of the total number of questions and answers (QA) of good guys vs. typical users in Figure 23. About 96% good guys have more than 10 QA, where 66% of typical users have more than 10 QA. While 59% good guys have more than 100 QA, only 21% typical users have more than 100 QA.

While the differentiation is visually apparent from the distributions in Figure 23, we ensure it is statistically significant verifying that the two samples are drawn from different probability distributions with $p_{ks} < 0.001$, D = 0.418, $p_{permute} < 0.001, Z = 129.17$. Also, goodness scores of good guys are correlated with their QA (r=0.41, p<0.001). These results show good guys are also good contributors of content.

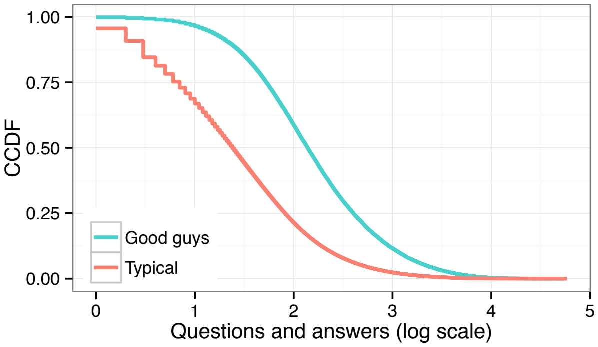


Figure 23 Distributions of the QA from good guys and other users.

4.5 Suspended User Prediction

Based on our previous analysis, we extract various types of features that we use to build predictive models. We formulate the prediction task as a classification problem with two classes of users: fair and suspended. Next, we describe the features used and the classifiers tested, and demonstrate that we are able to automatically detect fair from suspended users on Yahoo Answers with an overall high accuracy.

4.5.1 Features for Classification

Our predictive model has 29 features that are based on users' activities and engagements e.g., *social*, *activity*, *accomplishment*, *flag* and *deviance*. Table 5 shows the different categories of features used for the classification. *Social* features are based on the social network of the users, where *Activity* features are based on community contributions in the form of questions and answers. *Accomplishment* features acknowledge the quality of user contribution (e.g., points, best answers). *Flag* summarizes the flags of a user (both received and reported). *Deviance Score* features are the scores that we have got based on users' flags and activities. Finally, *Deviance*

Homophily represents the homophilous behavior with respect to deviance. Although most of the features are self-explanatory, below we clarify the ones which may not be.

Reciprocity measures the tendency of a pair of nodes to form mutual connections between each other [106]. Reciprocity is defined as follows:

$$r = \frac{L}{L^*}$$

where L is number of edges pointing in both directions and L^* is the total number of edges. $r = 1$ holds for a network in which all links are bidirectional (purely bidirectional network), while a purely unidirectional network has $r = 0$.

Status is defined as follows:

$$Status = \frac{\#followers}{\#followees}$$

Thumbs is the difference between the number of up-votes and the number of down-votes a user receives for all her answers. *Award Ratings* is the sum of the ratings a user receives for her best answers.

Altruistic scores is the difference between a user's contribution and his takeaway from the community. For altruistic scores, we consider *YA*'s point system, which awards two points for an answer, 10 points for a best answer, and penalizes five points for a question:

$$\begin{aligned} \text{Altruistic scores}_u &= f(contribution) - f(takeaway) \\ &= 2.0 * A_u + 10.0 * BA_u - 5.0 * Q_u \end{aligned} \tag{4.10}$$

where Q_u is the number of questions posted by u, A_u is the number of answers posted by u, and BA_u is the number of best answers posted by u.

4.5.2 Experimental Setup and Classification

In our dataset, the percentage of fair users (about 91%) are high compared to the suspended users (about 9%). This leads to an unbalanced dataset. Various approaches have been proposed in the machine learning literature to fix the unbalanced dataset. We use ROSE [198] algorithm to create a balanced dataset from the unbalanced one. ROSE creates balanced samples by random over-sampling minority examples, under-sampling majority examples or by combining over and under-sampling. Our prediction dataset has 250K users with 60-40% training–testing split. Using the under and over sampling technique of ROSE, we sample 150K users (fair and suspended each class has 75K users) to train the classifier. The testing set has 100K users, who are not present in the training dataset. They are drawn randomly and fair vs. suspended ratio in the testing dataset is the same as the original YA dataset.

We have used various classification algorithms, including Naive Bayes, K-Nearest Neighbors (KNN), Boosted Logistic Regression, and Stochastic Gradient Boosted Trees (SGBT) and found that the SGBT shows the best performance. SGBT offers a prediction model in the form of an ensemble of weak prediction models [100]. Table 6 shows a summary of our experimental setup. First, we use individual feature sets to investigate how successful one feature set is by itself only, and finally used all features for prediction. For evaluation, we measure widely used metrics in classification problems: Accuracy, Precision, Recall and F1-score.

4.5.3 Classification Results and Evaluation

Figure 24 shows the performance (accuracy, precision, recall and F1 score) of the models trained with different subsets of features using the Stochastic Gradient Boosted Trees (SGBT) classifier. We observe that each feature set has a positive effect on the performance of the classifier across all performance metrics. This suggests that all our feature sets are important for prediction. Particularly, accomplishment, deviance, flags and activity features individually can predict more than 70% accuracy with good precision, recall and F1 score. However, when all the features are used for classification, the performance metrics yield the best results, i.e., accuracy is improved by 4.11% compared to activity features.

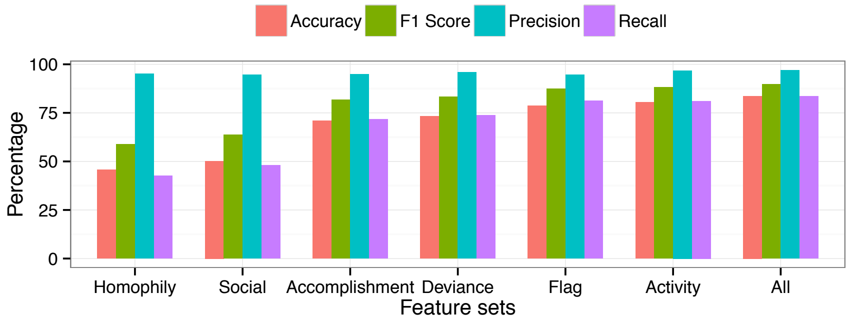


Figure 24 Performance of the SGBT while classifying fair and suspended users. Four performance measures are shown: Accuracy, Precision, Recall and F1 score.

The performance results of various classifiers while using all features are shown in Table 7. The SGBT classifier outperforms other classifiers in all performance metrics. It achieves 82.61% accuracy in classifying fair vs. suspended users with a high precision (96.94) and recall (83.52). The confusion matrix of the classifier is shown in Table 8. The matrix shows that the SGBT classifier is able to correctly classify 83.52% of fair users and 73.39% of suspended users.

Figure 25 shows the most important features (top 15) in classification of fair vs. suspended users. The model uses a backwards elimination feature selection method for feature importance. For each feature, the model tracks the changes in the generalized cross-validation error and uses it as the variable importance measure.

We observe that the number of flagged content and deviance scores are the best predictors of fair and suspended users. Also, at least one feature from all feature sets is within the top 15 features.

4.6 Summary and Discussions

This chapter investigates the flagging of inappropriate content in a popular and mature community Q&A, Yahoo Answers. Based on a year's worth of activity records that included

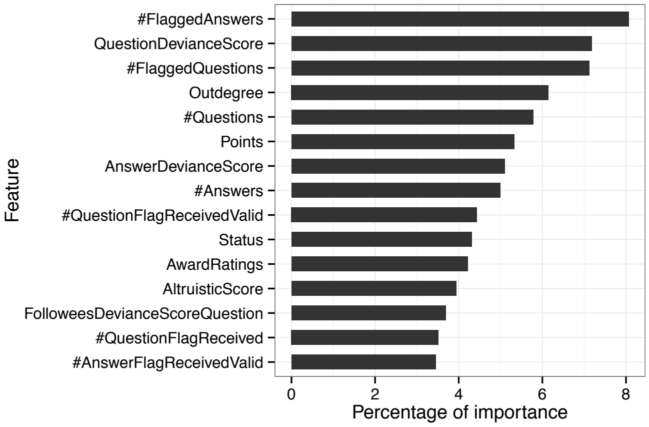


Figure 25 Relative importance of top 15 features in classifying fair and suspended users.

about 10 million flags in a population of about 1.5 million active users, our analysis revealed the following:

The use of flags is overwhelmingly correct, as shown by the large percentage of flags validated by human monitors. This is an important learning for crowd sourcing, as it shows for the first time (to the best of our knowledge) that crowd sourced monitoring of content functions well in CQA platforms. Moreover, although there are no explicit incentives (e.g., points) for flagging inappropriate content, users take the time to curate their environment. In fact, 46% of the users reported at least one abuse report, with the top abuse reporters flagging tens of thousands of posts.

Second, we discover that many users have collected a large number of flags, yet their presence is not necessarily bad for the community. Even more, their contributions are engaging, which is certainly a benefit to the platform: the questions asked by the users who deviate from the norm in terms of number of flags received for their postings receive many more answers and from many more users than the questions posted by ordinary users or by users who later had their accounts suspended. More content-based analysis is needed to understand how the deviant users engage the community. We posit that they might ask conversational questions rather than informative questions, as this behavior is shown to increase community engagement.

Third, we showed the importance of the follower-followee social network for channeling attention and producing answers to question. This network also channels the attention of flaggers: we showed that users in close social proximity are more likely to flag inappropriate content than distant users. Social neighborhoods, thus, tend to maintain their environment clean.

Fourth, a significant problem in *YA* is posed by the users who manage to avoid flagging, possibly by remaining at the outskirts of the social network. This relative isolation in terms of followers and in terms of interactions probably allows such users to remain invisible. They are likely caught by automatic spam-detection-like mechanisms and by paid human operators. Our empirical investigations show that classifiers that use activity-based features and social network-based features can successfully identify fair and suspended (40% of them are not flagged) users with an accuracy as high as 83%.

Our insights might inform the design of new applications, which include:

- *Quantified 'Self':* One could imagine user interfaces that show to what extent one's question-answering behavior deviates from the typical community behavior. By being aware of how their actions impact the community, users could accordingly modify or reinforce their behavior.

- *Collaborative Moderation Leadership:* Being able to identify passionate abuse reporters might well translate into promoting some of them to be community moderators and, as such, reducing CQA human inspectors' workload.

Our work has two main limitations. The first is that our analysis is not exhaustive as there might be unethical/deviant behaviors we have not considered. Users could game the incentive system for their own advantage. A user could, for example, post a question from a profile, answer that question from another profile, upvote the answer, and consequently accumulate more reputation points. The second limitation is that capturing deviance with a single metric (even after controlling for activity) might be oversimplified. For example, we have assumed that a user's behavior does not change over time. However, unstable behavior might be quite likely (especially for active users) and cannot be fully captured by a single measure of deviance.

Table 5 Different categories of features used for fair vs. suspended user prediction.

Category	Number	Features
Social	6	Indegree Outdegree Status Reciprocity Reciprocated networks degree Reciprocated networks CC
Activity	4	#Questions #Answers #Flagged Questions #Flagged Answers
Accomplishment	5	Points #Best Answers Award Ratings Thumbs Altruistic scores
Flag	8	#Question Flag Received #Question Flag Received Valid #Question Flag Reported #Question Flag Reported Valid #Answer Flag Received #Answer Flag Received Valid #Answer Flag Reported #Answer Flag Reported Valid
Deviance Score	2	Question deviance score Answer deviance score
Deviance Homophily	4	Followers' question deviance score Followers' answer deviance score Followees' question deviance score Followees' answer deviance score

Table 6 Details of experimental setup.

Dataset	250,000 users
Class Balancing Alg.	Random Over-Sampling Examples (ROSE)
Classifiers	Stochastic Gradient Boosted Trees (SGBT) Naive Bayes, Boosted Logistic Regression K-Nearest Neighbors (KNN) Support Vector Machines RDF
Feature Sets	Social, Activity, Accomplishment Flag, Deviance Homophily, All features
Train-Test Split	150K users training, 100K users testing
Cross Validation	10-folds, repeated 10 times
Performance	Accuracy, precision, recall, F1 score

Table 7 Performance of various classifiers while using all features.

Classifier Name	Accuracy	Precision	Recall	F1 Score
Naive Bayes	47.21	96.93	43.34	59.89
Boosted Logistic Regression	71.61	96.62	71.28	82.03
KNN	73.81	96.41	73.97	83.71
SVM-RDF	75.92	95.62	77.06	85.34
SGBT	82.61	96.94	83.52	89.73

Table 8 Confusion matrix for SGBT classifier.

		Actual	
		Fair	Suspended
Predicted	Fair	83.52%	26.60%
	Suspended	16.47%	73.39%

CHAPTER 5: PRIVACY CONCERNS VS. USER BEHAVIOR IN COMMUNITY QUESTION ANSWERING SOCIAL NETWORKS[1]

Given the social privacy risks and abuse users face in OSNs, it is realistic to assume that they might feel intimidated to actively participate in the community. However, one the other hand, privacy-concerned users might feel more control of privacy settings and could engage more with the community. It is unclear, thus, what is the association between users' privacy concerns and their behavior in an online social network.

Various studies [63, 235, 259] have shown the correlation between users' self-reported privacy concerns and their self-reported behavior in online social networks. For example, users who express concerns on Facebook privacy controls and find it difficult to comprehend sharing practices also report less engagement such as visiting, commenting, and liking content [259]. At the same time, users who exercise their privacy rights (specifically, by restricting the visibility of their content) are more engaged and thus contribute more to the community. Similarly, the frequency of visits, type of use, and general Internet skills are shown to be related to the personalization of the default privacy settings [63]. However, measuring privacy concerns and behavior through self-reporting is subject to bias [113, 215].

In this chapter, we focus on CQA social networks and empirically measure the relationship between users' privacy concerns and contribution behavior from their recorded activity logs. Privacy settings are typically available for CQA platform users to personalize. We use modifications of the privacy settings as a proxy of privacy concern, and users' recorded activity logs to infer their behavior. We analyzed more than a year of activity traces from 1.5 million users

[1] Much of the work in this chapter was first published in [148]. Permission is included in Appendix A.

from *YA* (the dataset is discussed in Chapter 3) to answer the following questions related to users' contribution behavior:

R_5 : Are there quantitative and qualitative differences in user contributions between user groups with private vs. public settings?

R_6 : Is user engagement (measured by frequency of contributing content and number of social contacts) correlated with user privacy settings?

R_7 : Do users with privacy settings enabled tend to violate community norms more than users with public content?

Our study finds that privacy-concerned users contribute more to the community. They are more engaged, having higher retention and larger social circles, and have higher perception on answer quality. However, they also exhibit more violations of platform rules in asking and answering questions than the users with public profiles.

The chapter is structured as follows. Section 5.1 overviews privacy settings in *YA* and presents more granular level questions on users' contribution behavior. Section 5.2 presents the results. We conclude with a discussion of results in Section 5.3.

5.1 Privacy Settings in *YA*

Our goal is to study the association between privacy concerns and behavior in *YA*. Previous works [63, 259] on Facebook have inferred users' privacy concerns using their self-reported feedback on privacy. Rather than self-reporting, which is subject to bias [113], we use modifications on privacy settings as a proxy for privacy concern.

The available privacy configurations in *YA* allow 4 user groups:

1. Public: all information is publicly visible (87.20% of users). This is the default setting.

2. QA-private: only Q/A information is private (2.23% of users), i.e., their questions and answers are visible only to their followers.

3. Network-private: only network information is private (0.81% of users), i.e., only their followers see the network.

4. Private: Q/A and network information is private (9.74% of users), thus only visible to the user's followers.

In the rest of the study we collectively refer to the QA-private and network-private users as semi-private. The default privacy in *YA* is public. It might be possible that many of the users in the pubic group are dormant: users who signed up, asked and answered some questions, and disappeared quickly. These users might skew the results of our study, thus, we only consider active users, who have asked and answered more than 10 questions. The active users are about 68% of the population and out of them 84.43% are public, 2.50% are QA-private, 0.89% are network-private, and 12.16% are private. We note that our observations remain the same even if we consider more active users by filtering-in users who have asked and answered more than 20 questions.

We measure several characteristics of user behavior that are related to CQA such as engagement, retention, accomplishments, abuse reporting, and deviance. We ask the following questions:

1. Is privacy preference associated with user engagement?
   We consider two metrics of user engagement: retention, which measures the average interval time between consecutive user contributions (addressed in Section 5.2.1), and social engagement, given by the number of followers and followees (Section 5.2.2). This question aims to investigate the pattern identified in survey-based Facebook studies, but using CQA-specific and more nuanced engagement metrics on longitudinal activity traces.

2. Do privacy-concerned users contribute differently to the community than public users?
   Users contribute by posting questions and answering others' questions. The quality of user-generated content is measured in the number of best answers and the askers' satisfaction with the answer received. The overall activity is measured in points. We characterize user contributions quantitatively and qualitatively in Section 5.2.3.

3. Do privacy-concerned users have different perception on answer quality than public users?
   Users can themselves select best answers for their posted questions or they can rely on community voting to mark the best answers. In Section 5.2.4, we look at how the community sees the best answers selected by the users who received them. Specifically, we compare the quality of best answers selected by privacy-concerned users with those selected by public users in terms of the number of thumbs-up and thumbs-down given by the community.

4. Are privacy-concerned users also more abuse-conscious?
   Intuitively, engagement is also correlated with the desire to keep the community free of unethical users (who, for example, may post spam in violation of the community rules). The related analysis is presented in Section 5.2.5.

5. Are privacy-concerned users more likely to violate community rules?
   Intuitively, reduced visibility can give a false sense of confidence that might lead to violations of community rules. One study [30] in online gaming social networks shows that newly found and banned cheaters are more likely to change their profile to a more restrictive privacy settings than non-cheaters. In *YA*, we ask, is this observed more with privacy-concerned users than public users? This question is studied in Section 5.2.6.

5.2 Results

This section presents the results of the research questions we asked in the previous section.

5.2.1 Privacy and Retention

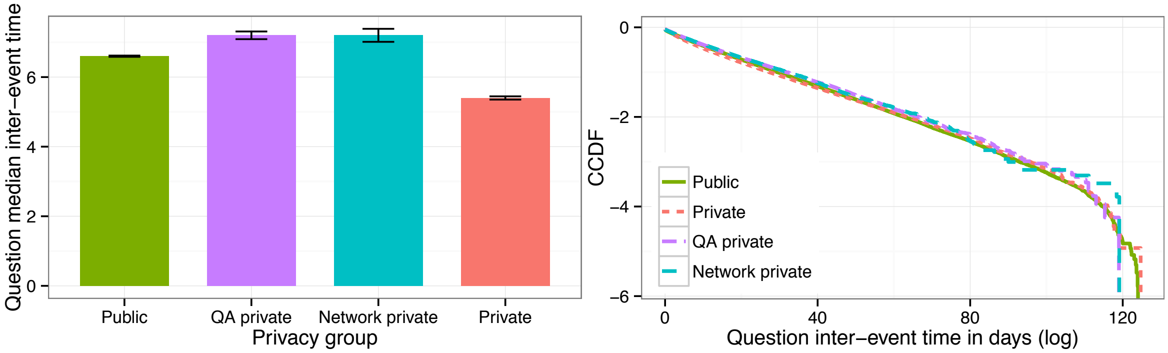
We define retention as the inverse of the average time difference between two actions not marked as abusive (i.e., fair). We consider two types of retention, based on questions and answers. For both types, if a user has a high average time difference between two fair actions, her retention is low.

Figures 26(a) and (b) show the medians and CCDF of the question inter-event time for the different groups, respectively. On average, private users have lower question inter-event time (thus higher retention) than public users. The answer inter-event time in Figures 27(a) and (b) show similar patterns. It seems semi-private (QA-private and network-private) users have higher average inter-event time, compared to private users, but similar to public users.

We performed a Kruskal-Wallis test to assess the difference among privacy groups in terms of retention. The test shows that at least one of the groups is different from at least one of the others for question ($\chi^2 = 458.83, df = 3, p < 2.2e - 16$) and answer retention ($\chi^2 = 119.32, df = 3, p < 2.2e - 16$). All-pairwise comparison tests after the Kruskal-Wallis test show that besides the QA-private and network-private for question retention and network-private and private for answer retention, all others are different for questions and answers retention ($p<0.05$). These results show that privacy-concerned users are more retained than others.

5.2.2 Privacy and Social Circles

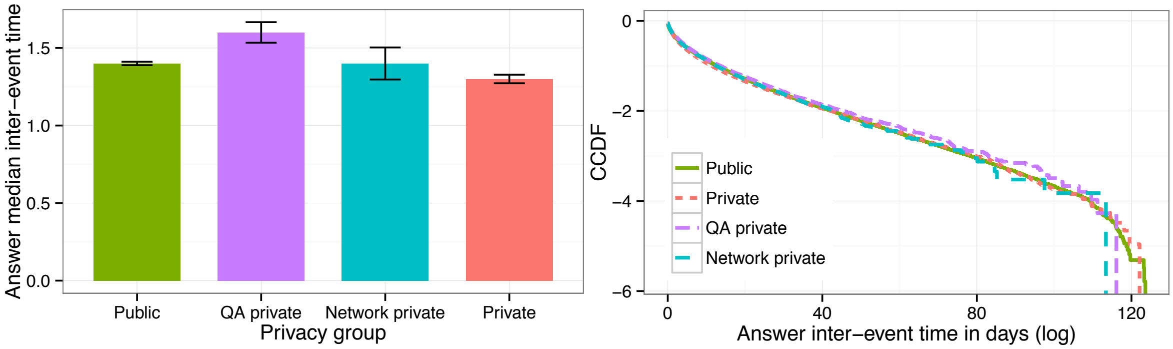
YA users can follow each other, thus, we compute the indegree (total number of followers) and outdegree (total number of followees) of different privacy group users. Figures 28(a) and 29(a) show the median of indegree and outdegree, respectively, for the four privacy groups.



(a)

(b)

Figure 26 (a) Median of question inter-event time in days with standard error bars; (b) CCDF of question inter-event time in days.



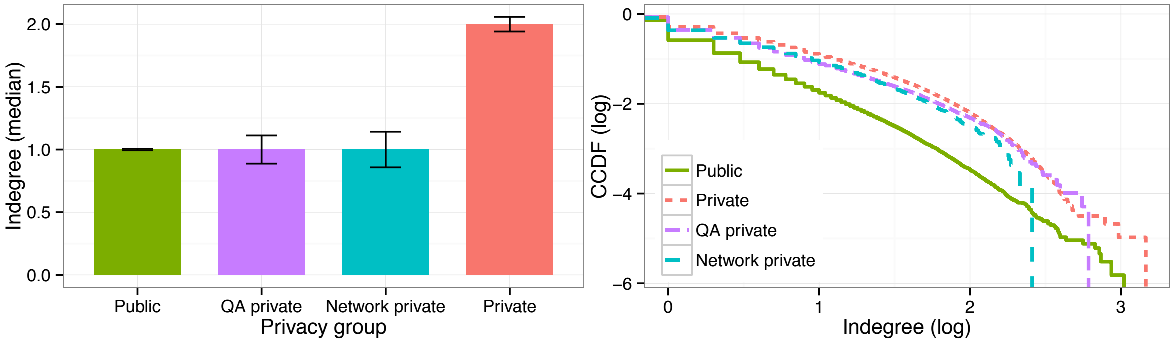
(a)

(b)

Figure 27 (a) Median of answer inter-event time in days with standard error bars; (b) CCDF of answer inter-event time in days.

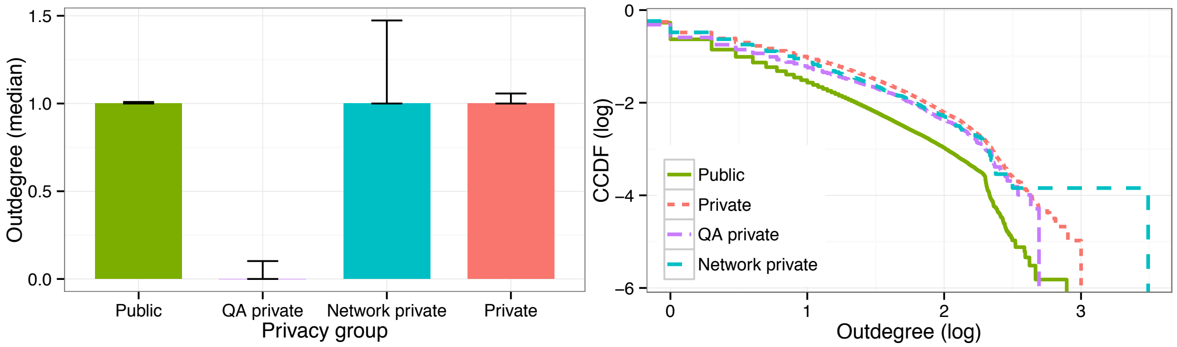
The CCDF of indegree and outdegree of them are shown in Figures 28(b) and 29(b), respectively. While 20.56% of private users have more than 5 followers, only 4.42% of public users do. However, 15.33% of network-private and 14.48% of QA-private users have more than 5 followers. Alternatively, while 14.79% of private users follow more than 5 users, only 5.85% of public users do. For network-private and QA-private users, these numbers are 12.92% and 9.79%, respectively.

The results indicate that more restrictive private settings users have richer social circles. Indeed, Kruskal-Wallis tests show that at least one of the privacy groups is different from at least one of the other groups, for both the indegree ($\chi^2 = 29383.67, df = 3, p < 2.2e - 16$) and outdegree ($\chi^2 = 2913.63, df = 3, p < 2.2e - 16$). All-pairs comparison tests between the privacy groups show that all pairwise privacy groups are different ($p < 0.05$) for indegree, and only network-private and private users are same for outdegree ($p < 0.05$).



(a) (b)

Figure 28 (a) Median of indegree with standard error bars; (b) CCDF of indegree.



(a) (b)

Figure 29 (a) Median of outdegree with standard error bars; (b) CCDF of outdegree.

5.2.3 Privacy and Accomplishments

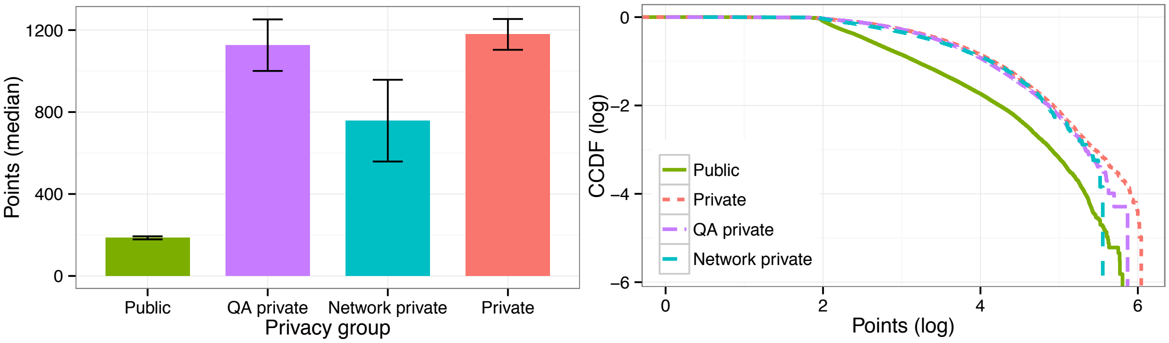
We consider two accomplishments that measure the quantity and quality of user contribution, through the point system described in Section 3.1. Quantity of contribution is measured by the points users earn for their activities. To measure quality of contribution we use two metrics:

Best Answer Percentage (BAP) and Award Rating Percentage (ARP). BAP is the percentage of a user's answers that are selected as best. ARP measures how satisfactory a user's best answers are. A *YA* asker can rate a best answer from 1 to 5 to declare how satisfied she is with the answer. ARP_j is the average rating a user j receives for her best answers:

$$ARP_j = \frac{\sum_{i=1}^{\#\text{best answers of j}} \text{Award rating for best answer i}}{\#\text{Total answers of j * 5}} * 100$$

Figure 30(a) shows median points with standard error for different privacy group users. It appears that median points of private and semi-private users are higher than public users. In fact, the CCDF of points in 30(b) shows that while 53.28% of private, 52.35% of QA-private and 45.51% of network-private users have more than 1000 points, only 14.14% of public users have more than 1000 points.

A Kruskal-Wallis test shows at least one of the privacy groups is different from at least one of the other groups for award points ($\chi^2 = 75884.12, df = 3, p < 2.2e - 16$). Moreover, all-pairs comparison tests between the four privacy groups show that besides private and QA-private, all others are different ($p < 0.05$). These results indicate that privacy-concerned users contribute more in YA from a quantitative point of view.



(a)

(b)

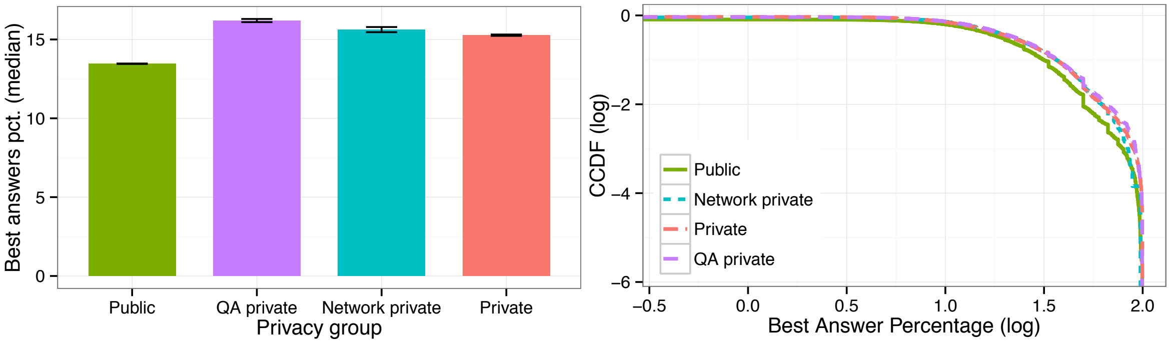
Figure 30 (a) Median of points with standard error bars; (b) CCDF of points.

However, unlike quantitative contributions where public users are far behind the private ones, we found smaller, albeit significant, difference in the qualitative contributions among the four privacy groups. Figures 31(a) and 32(a) show the medians of best answer percentage (BAP) and award rating percentage (ARP) of different privacy group users, respectively. Although in both cases, private and semi-private group users have higher percentage than public users, the difference is less compared to points (even by a visible inspection on CCDF of BAP (Figure 31(b)) and ARP (Figure 32(b)) shows no difference across all privacy groups). Analyzing the CCDFs we get 27.96% of public users have best answers percentage more than 20, and 34.91% of private, 35.42% of network-private and 37.17% of QA-private users have best answers percentage more than 20. On the other hand, 27.10% of public, 33.56% of private, 34.03% of network-private and 36.01% of QA-private users have award rating percentage more than 20.

For both BAP and ARP, we notice that all privacy groups' numbers (median or CCDF) are close, especially private and network-private. So, one important question is how different privacy groups are in terms of users' qualitative contribution. We conducted a Kruskal-Wallis test on both BAP and ARP. The test results show that at least one of the privacy groups is different from at least one of the other groups for BAP ($\chi^2 = 5832.93, df = 3, p < 2.2e - 16$) and also for ARP ($\chi^2 = 5604.056, df = 3, p < 2.2e - 16$). Moreover, all-pairs comparison tests between the four privacy groups show that only private and network-private groups are the same ($p < 0.05$), and all other pairwise privacy groups are different. Thus, we confirm that privacy-concerned users have higher quantitative and qualitative contributions than others.

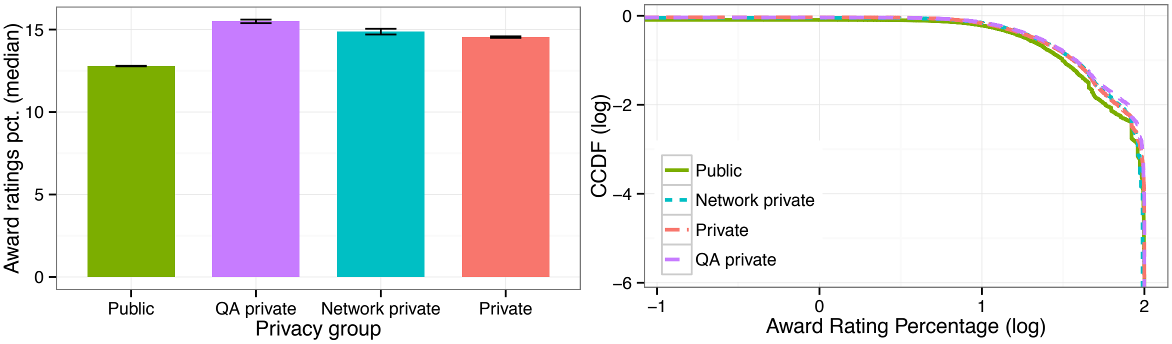
5.2.4 Privacy and Best Answer Quality

In *YA*, the best answer of a question is selected either by the asker of the question or by the community. If an asker does not select the best answer, the community members do that by voting. We first look at how different privacy groups are in selecting the best answers by them-



(a) (b)

Figure 31 (a) Median of best answers percentage with standard error bars; (b) CCDF of best answers percentage.

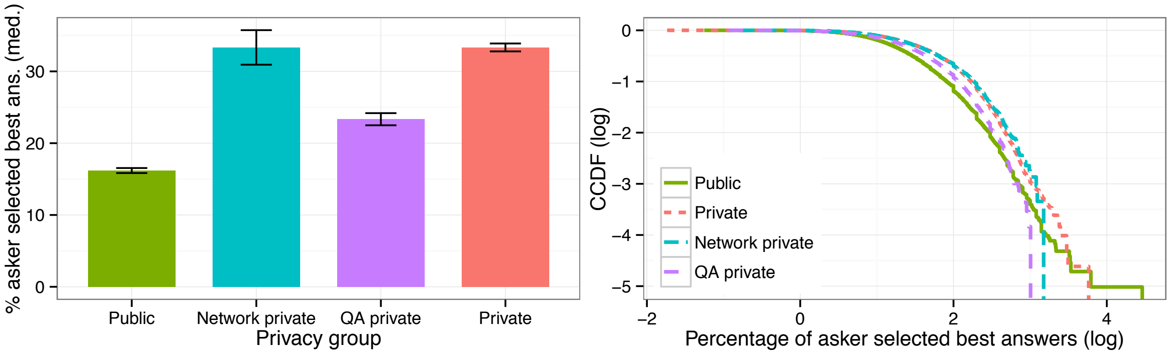


(a) (b)

Figure 32 (a) Median of award rating percentage with standard error bars; (b) CCDF of award rating percentage.

selves. We calculate the percentage of the best answers selected out of the total number of questions asked per user.

Figures 33(a) and (b) show the median and CCDF of the percentage of asker-selected best answers for different privacy group users, respectively. Analyzing the distribution, we observe that while 61.35% of private, 57.85% of network-private, 51.61% of QA-private users selected more than 20% of their best answers by themselves, only 38.35% of public users have done the selection by themselves. A Kruskal-Wallis test shows that at least one of the privacy groups is different from at least one of the other groups in terms of asker-selected best answers ($\chi^2 =$



(a)

(b)

Figure 33 (a) Median of percentage of asker selected best answers with standard error bars; (b) CCDF of percentage of asker selected best answers.

$9522.60, df = 3, p < 2.2e - 16$). All-pairs comparison tests between the four privacy groups show that besides the network-private and private groups, all other pairwise privacy groups are different $(p < 0.05)$.

Next, we focus on the quality of the best answers that users selected by themselves. We measure this quality based on community members' feedback on those answers. Community members can provide feedback on answers by giving either a *thumbs up* or a *thumbs down* (at most one such feedback per answer). For each user j who selected best answers to his own questions, we calculate the average number of thumbs as the ratio between the positive community feedback and the number of asker-selected best answers.

$$\text{AvgThumbs}_j = \frac{\text{\# Thumbs up – \#Thumbs down}}{\text{\# Best answers selected by j}}$$

Figures 34(a) and (b) show the median and CCDF of the average thumbs on best answers selected by the askers, respectively. The distribution shows that all private group users have more average thumbs on best answers than public users. We observe that while 21.40% of private, 24.62% of network-private, 16.51% of QA-private users have got 5 average thumbs on their best answers, only 11.45% of public users have got 5 average thumbs on the best answers they selected. A Kruskal-Wallis test shows that at least one of the privacy groups is different from

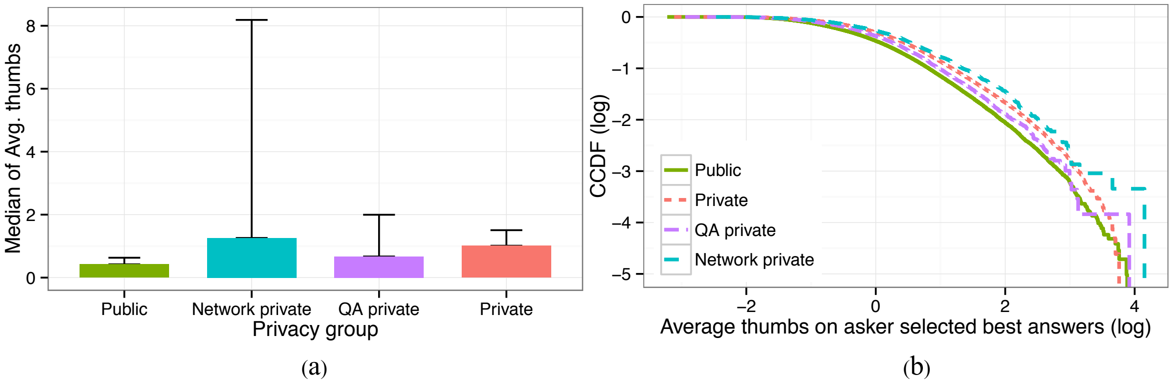


Figure 34 (a) Median of average thumbs on asker selected best answers with standard error bars; (b) CCDF of average thumbs on asker selected best answers.

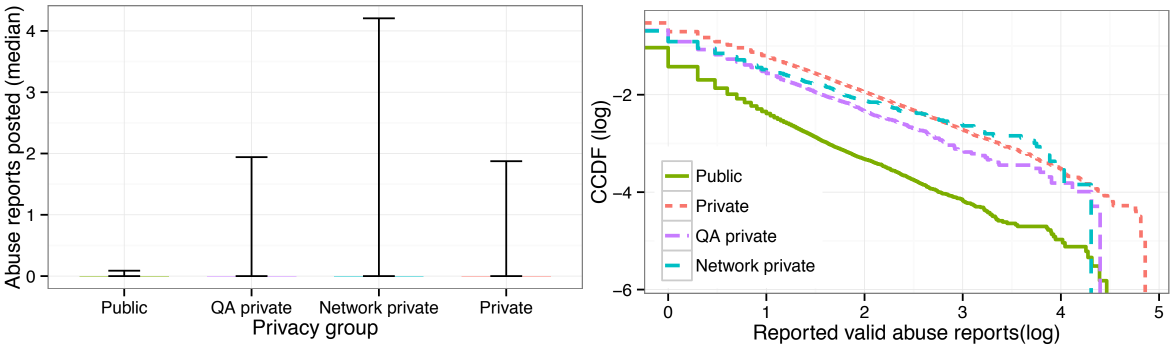
at least one of the other groups in terms of average thumbs with $\chi^2 = 5680.47, df = 3, p < 2.2e - 16$. All-pairs comparison tests between the four privacy groups show all pairwise privacy groups are different $(p < 0.05)$.

5.2.5 Privacy and Abuse Reporting

As a crowd-sourced community, *YA* relies on its users for self moderation. Thus, users not only provide questions and answers, but also report inappropriate content using the abuse report functionality. If the report is valid, the content is deleted from the community. In this way, users serve as an intermediate layer in the *YA* moderation process since these abuse reports are verified by human inspectors. We have already seen that privacy preferences of users have significant association with a number of different dimensions including retention and accomplishments, thus we suspect that privacy is also associated with abuse reporting.

The median and CCDF of the valid abuse reports posted by users are shown in Figures 35(a) and (b), respectively. Although, abuse reports are highly appreciated for maintaining a clean CQA environment, very few people tend to report abuses. We find that 46% of the users reported only one abuse and 90% of abuse reports are contributed by only 7.96% of users. So, it's not surprising that all median values are zero in Figures 35(a). However, the private users have very high variability in abuse reporting compared to the public users.

The distributions in Figure 35(b) show that, on average, private users have posted more abuse reports than semi-private and public users. Indeed, all three private groups of users have posted a very large number of valid abuse reports compared to public users. Analyzing the distribution, we observe that 5.93% of private, 3.15% of network-private, 2.73% of QA-private and only 0.20% of public users have posted more than 10 valid abuse reports. A Kruskal-Wallis test shows that at least one of the privacy groups is different from at least one of the other groups in terms of abuse reporting behavior ($\chi^2 = 37647.77, df = 3, p < 2.2e-16$). All-pairs comparison tests between the four privacy groups show that besides the QA-private and network-private groups, all other pairwise privacy groups are different $(p < 0.05)$.



(a)

(b)

Figure 35 (a) Median of average valid abuse reports with standard error bars; (b) CCDF of valid abuse reports.

5.2.6 Privacy and Deviance

Deviant behavior is defined by actions or behaviors that are contrary to the dominant norms of the society [78]. Although social norms differ from culture to culture, within a context, they remain the same and they are the rules by which the members of the community are conventionally guided. *YA* has established norms as reflected by its community guidelines and terms of service [8]. We define user behaviors as deviant if they depart from these norms. In Section 4.1, we define a *deviance score* metric that indicates how much a user deviates from the norm in terms of received flags considering the amount of the user's activity. In short, we define the deviance score for a user u as the number of correct abuse reports (flags) she receives over the total content (question/answer) she posted, after eliminating the expected average number of correct abuse reports given the amount of content posted:

$$\text{Deviance}_{\text{Q/A}}(u) = Y_{Q/A,u} - \hat{Y}_{Q/A,u} \tag{5.1}$$

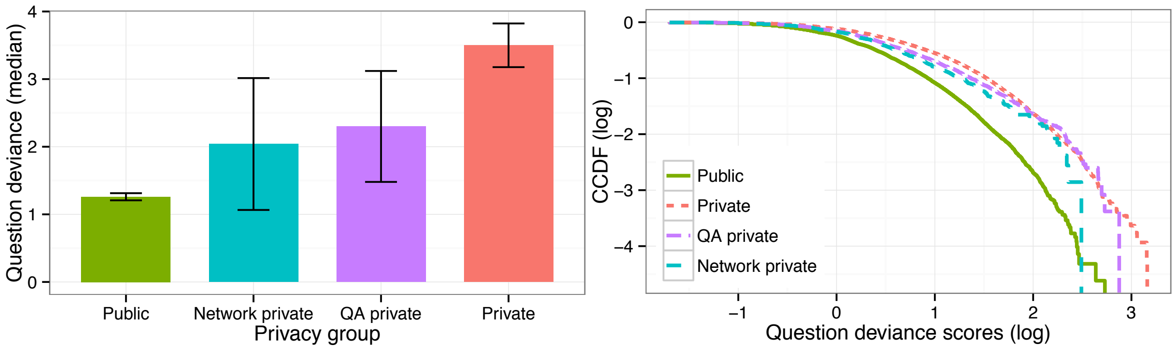
where $Y_{Q/A,u}$ is the number of correct abuse reports received by u for her questions/answers, and $\hat{Y}_{Q/A,u}$ is the expected number of correct abuse reports to be received by u for those questions/answers.

To capture the expected number of the correct abuse reports a user receives for questions/answers, we considered a number of linear and polynomial regression models between the response variable (number of correct abuse reports) and the predictor variable (number of questions/answers). Among them, the following linear model was the best in explaining the variability of the response variable.

$$Y = \alpha + \beta X + \epsilon \tag{5.2}$$

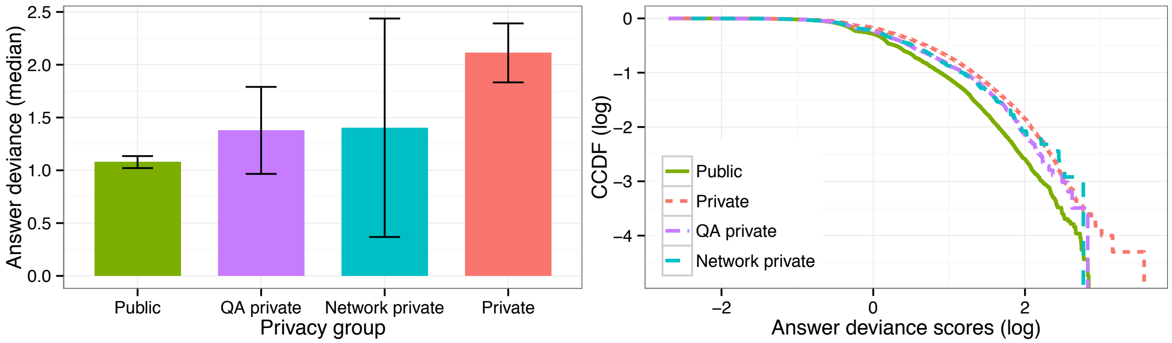
where Y is the number of correct abuse reports (flags) received for the content, X is the number of content posts and ϵ is the error term. In eq. (5.1), a positive deviance score reflects deviant users, i.e., those whose deviance cannot be only explained by their activity levels.

Figures 36(a) and (b) show the median and CCDF of the question deviance scores, respectively. In both cases, private and semi-private users' question deviance scores are higher than the public users. Also private users' question deviance scores are higher than semi-private users. We reach to the same conclusion for the answer deviance scores from the median and CCDF



(a) (b)

Figure 36 (a) Median question deviance scores with standard error bars; (b) CCDF of question deviance scores.



(a) (b)

Figure 37 (a) Median answer deviance scores with standard error bars; (b) CCDF of answer deviance scores.

of the answer deviance scores for all users in Figures 37(a) and (b), respectively. The Kruskal-Wallis test shows that at least one of the privacy groups is different from at least one of the other groups for question ($\chi^2 = 4432.72, df = 3, p < 2.2e - 16$) and answer deviance scores ($\chi^2 = 2662.416, df = 3, p < 2.2e - 16$). All-pairs comparison tests between the four privacy groups show that besides the network-private and QA-private groups, all other pairwise privacy groups are different ($p < 0.05$).

5.3 Summary and Discussions

By performing a large-scale quantitative study, we have shown how users' privacy concerns relate to their behavior in Yahoo Answers.We used users' modifications on their privacy settings as a proxy of privacy concerns and grouped users into three main categories: private, semi-private (consisting of two groups, QA-private and network-private), and public.

Our study highlighted a number of results. First, we found that 87.20% of user accounts on *YA* are public, the default privacy setting. This result is similar with Gross and Acquisti's study [116] on Facebook, where they found that about 90% of user profiles maintained the default, public setting. While expected, this confirmation warns again about the importance of correct default settings in online applications.

Second, we discovered that users with enabled privacy settings are more engaged with the community: they have higher retention, more social network contacts, they are better citizens in terms of reporting abuses, overall they contribute more and better content, and have higher perception on answer quality. This is in line with Staddon et al.'s study [259] on Facebook, who found that users reporting more control and comprehension over privacy are more engaged with the platform. Therefore, this result is important for two reasons: it applies to a type of online community not previously studied, and it is based on user logs instead of user surveys, prone to self-reporting bias.

Third, we found that, on average, privacy-concerned users show more behavioral deviation in asking and answering questions than users with public accounts. At a first look, this result seems counterintuitive, given that privacy-concerned users keep the environment clean by reporting more abuses. However, this result is consistent with our previous analysis in Section 4.2, which finds that deviance in CQA platforms is not necessarily bad. Deviant users in *YA* are found to promote user engagement by attracting more users to answer more of their questions.

In addition to characterizing the association between privacy concerns and user behavior, our results may lead to improvements in CQA platforms operation. Whether an expression of privacy awareness or Internet savviness, users who modify their default privacy settings can be expected to be better citizens. If they change their account settings early on in their interaction with the platform, they send a clear signal to platform operators of likely commitment.

CQA platforms could benefit by targeting these users in a number of ways. For example, the indication of changing privacy settings can be used in *question recommendation*, where questions are routed to the most appropriate users who are more likely to answer. To find such answerers, typical factors considered are followers, interests, question category, diversity and freshness; privacy settings can also serve as a complementary factor. Also, some of these users could be assigned community moderating duties to monitor community health, as our results show that they report more abuses. However, users who do not change their privacy settings are found to be less engaged. For these users, CQA platforms could provide extra incentives for participation and increased retention.

Our work also shows the importance of user-friendly and more practical design of privacy controls, as we find that increased engagement is associated with the use of privacy controls. For example, the lack of appropriate visual feedback has been identified as one of the reasons of the under-utilization of privacy settings [263]. A better interface for setting privacy controls in the CQA platforms can impact users' understanding of privacy settings and thus their success in exercising privacy controls.

We acknowledge that our study is observational, hence we can only associate privacy concerns with user behavior. In the absence of controlled experimental ground truth data, we cannot draw causal conclusions regarding whether users' privacy concerns lead to different behavioral pattens in contribution.

CHAPTER 6: CULTURES IN COMMUNITY QUESTION ANSWERING SOCIAL NETWORKS[1]

We have already seen a study on *YA*'s content abusers in Chapter 4. Moreover, we have also discussed the relationship between users' engagement and privacy concerns in Chapter 5. This chapter views users' engagement, privacy concerns and abusive behavior through the lenses of national cultures in *YA* CQA social networks.

Cultural differences exist in almost all aspects of social interactions. For example, in some cultures in Asia it may be considered disrespectful for people to express their opinions or ask questions to authority figures (e.g., teachers, elders). In other cultures (such as USA or Canada) asking questions is expected or even encouraged.

Cross-country cultural variations have been studied in the real world via small-scale experiments and opinion surveys. Geert Hofstede [130] administered opinion surveys to a large number of IBM employees from different countries in the 1960s and 1970s. He discovered five cultural dimensions (individualism, power distance, uncertainty avoidance, masculinity, and long term orientation), that can be attributed to the existence of cultural variations. Three of these dimensions, individualism, power distance, and uncertainty avoidance, have been used to assess cultural differences in online contexts such as Twitter communication [104], emoticon usage [224] and online scheduling [239]. In brief, individualism reflects the extent to which an individual is integrated into a group (e.g., individualistic cultures like USA emphasizes mostly on their individual goals, as opposed to collectivist cultures like China that emphasizes on group harmony and loyalty). Power distance is the extent to which the less powerful members of an organization or society expect and accept that power is distributed unequally (e.g., in high power

[1] Much of the work in this chapter was first published in [149]. Permission is included in Appendix A.

distance countries subordinates simply comply with their leaders). Uncertainty avoidance defines the extent to which society members feel uncomfortable with uncertainty and ambiguity (e.g., the stereotypical Swiss plans everything ahead supposedly to avoid uncertainty).

Psychologist Robert V. Levine [174] proposed the *Pace of Life* metric based on the walking speed of city people over a distance of 60 feet, the service time for standard requests for stamps, and the clock accuracy of city banks. During the 1990s, Levine employed 19 experimenters in large cities from 31 countries and computed country-specific Pace of Life ranks. He found significant differences in Pace of Life across cultures and ranked the cultures based on that.

Such cross-cultural variations that sociologists and psychologists already found in the offline world lead to our main research question:

R_8 : Does national culture determine how we participate in online Community Question Answering (CQA) social networks?

National cross-cultural variations have been studied in a number of online contexts, including social networks (e.g., Twitter [111], Facebook [234]), location search and discovery (e.g., Foursquare [250]) and online scheduling (e.g., Doodle [239]). If cultural variations exist in CQA platforms, they could be used for more informed system design, including question recommendation, follow recommendation, and targeted ads.

In this chapter, we analyzed about 200 thousand sampled *YA* users from 67 countries who were active between 2012 and 2013. We tested a number of hypotheses associated with Hofstede's cultural dimensions and Levine's Pace of Life. Our results indicate that *YA* is not a homogeneous subcultural community: considerable behavioral differences exist between the users from different countries.

The rest of the chapter is structured as follows. Section 6.1 presents how we selected a representative sample of countries for the study. We introduce the hypotheses and present the results relating to Levine's Pace of Life and Hofstede's cultural dimensions in *YA* in Section 6.2 and Section 6.3, respectively. We discuss the impact of these results in Section 6.4.

6.1 Data Preparation

We studied a random sample of about 200K users from *YA* who were active between 2012 and 2013. These users posted about 9 million questions (about 45 questions/user), 43 million answers (about 215 answers/user), and 4.5 million abuse reports (about 23 reports/user). They are connected via 490K follower-followee relationships in a social network. The indegree and outdegree distributions of the social network follow power-law distributions, with an exponential fitting parameter α of 1.83 and 1.85, respectively.

In our dataset, we have users from 67 countries. Figure 38 shows the number of users in our dataset as a function of the number of Internet users taken from the World Bank.[2] We find a linear relationship between the number of users per country in our *YA* dataset and the number of Internet users in the World Bank dataset for each country. It means that the *YA* users from our dataset are not skewed by country. Instead, they represent a sample of global Internet users.

To investigate how sensitive this correlation is to the number of users per country, we computed the Pearson correlation between the number of *YA* users in x countries and their respective Internet population. The x countries were ranked based on the number of *YA* users found in the dataset, and x was varied from top 20 to all 67 countries. Figure 39 shows that there are several peaks in the correlation, but the values are high and between 0.5 and 0.7. We select as a threshold the second highest correlation peak and thus included in the study 41 countries which have at least 150 users per country.

[2] http://www.data.worldbank.org/indicator/IT.NET.USER.P2

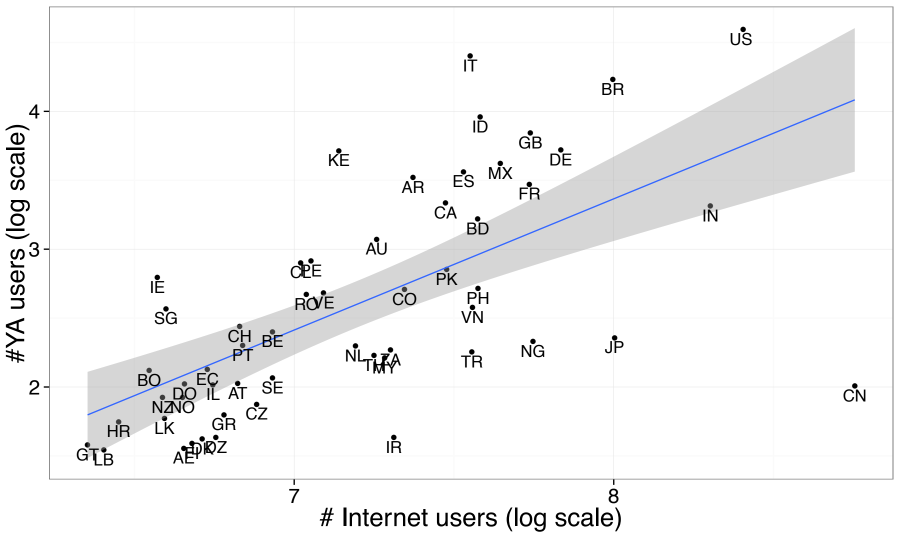


Figure 38 The number of Internet users and YA users for 67 countries. The regression line and 95% confidence interval area are also shown. The countries are represented by a 2-letter country code based on ISO 3166.

6.2 Levine's Pace of Life

In this section, we analyze Levine's Pace of Life cultural dimension in the context of *YA* and show how it relates to user activities such as questioning, answering and reporting. In his book [173], psychologist Robert Levine defines Pace of Life as "the flow or movement of time that people experience". With the help of 19 experimenters, he collected and compared three indicators of the Pace of Life in 36 large cities from 31 countries around the world during a warm summer month between 1992 and 1995 [174]. The indicators are:

- Walking speed: They measured walking speed of 35 men and 35 women over a distance of 60 feet in main downtown areas in each city. Measurements were done during prime business hours after controlling a number of variables such as sidewalks, crowd, effects of socialization. They found significant differences in pedestrians walking speed—for example, pedestrians in Rio de Janeiro, Brazil walked only two-thirds as fast as pedestrians in Zurich, Switzerland.

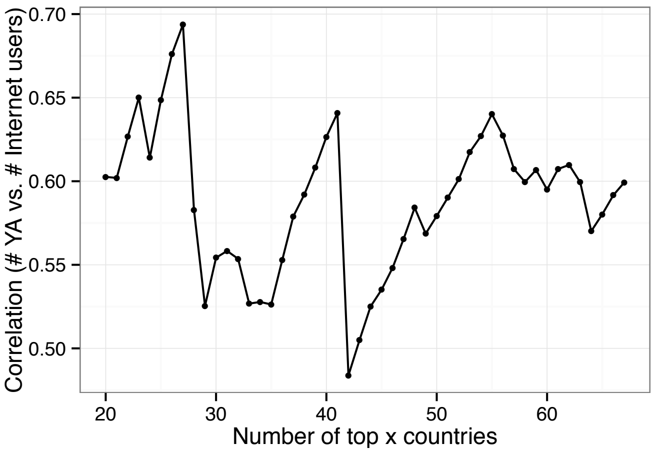


Figure 39 Number of top countries based on the number of YA users and correlation with their number of Internet users. All correlations are statistically significant with p-value<0.05.

- Postal speed: In each city, they measured the time it took postal workers to serve a standard request for stamps and considered this time as a proxy for work speed. They handed each clerk money and a note written in the local language requesting a common stamp. For example, in the United States, the clerk was handed a 5 dollar bill with a request for one 32-cent stamp. They found that overall Western Europe was the fastest to serve a standard request.

- Clock accuracy: To quantitatively measure time concerns, the researchers checked the clock accuracy of randomly selected 15 downtown banks in each city. The reported times were then compared to those reported by the telephone company, which was considered accurate.

Levine combined these three scores into a country-specific Pace of Life score and concluded that “the Pace of Life was fastest in Japan and the countries of Western Europe and was slowest in economically undeveloped countries. The pace was significantly faster in colder climates, economically productive countries, and in individualistic cultures” [174].

Intuitively, to cope with the rigid perception of time, people from the higher Pace of Life countries have to be planned and organized in their daily activities. On the other hand, people from lower Pace of Life countries might allow some unstructured activities, as in those countries the expectation of following the ‘right’ time is more relaxed.

Applying these findings to online communities such as CQA platforms, we expect that people from higher Pace of Life countries, such as the USA, will be less likely to ask or answer questions during busy hours of the day, e.g., office hours. From these ideas, we hypothesize the following in *YA*:

$H1$: Users from countries with a higher Pace of Life score show more temporally predictable activities.

To test this hypothesis, we calculate how probable a country’s users are in asking, answering and reporting at different times of day and correlate that with that country’s Pace of Life rank. For example, if a user only asks or answers questions in the evening, he is temporally more predictable than a user who asks or answers in the morning, afternoon and night. In a Twitter study [111], Golder and Macy also find diurnal mood rhythms in different cultures.

In order to calculate temporal predictability, we only consider working days, as weekends are less predictable. More specifically, similar to [105], we divide the working day in five time intervals: morning (6:00 - 8:59), office time (9:00-17:59), evening (18:00-20:59), late night (21:00- 23:59), sleeping time (00:00 - 05:59). All the reported times are users’ local time. We use *information entropy* [58], a measure of disorder, to calculate the temporal predictability.

For a given activity (asking, answering, or reporting) and C intervals, we can compute $p(c)$, the probability of an activity belonging to interval c. We measure the normalized entropy for user u for all activities as:

$$Entropy_u = \frac{-\sum_{c \in C} p(c)log(p(c))}{|logC|} \tag{6.1}$$

We calculate users’ normalized entropies for all their questions, answers and abuse reports and refer to them as *question*, *answer* and *report entropy*, respectively. In our dataset, each country has on average 134K questions, 642K answers and 67K abuse reports. Normalized entropy ranges from 0 to 1. A normalized question entropy close to 0 indicates that most of the questions the user asked are within one time interval of the day, whereas the closer to 1, the more likely is that the user asked questions during all intervals. Finally, the question/answer/report entropy for a country c, $Entropy_{q/a/r,c}$, is defined as the geometric mean of all $Entropy_{q/a/r,u}$ computed for the users of that country:

$$Entropy_{q/a/r,c} = \Big(\prod_{u \in U_c} Entropy_{q/a/r,u}\Big)^{\frac{1}{|U_c|}} \tag{6.2}$$

where U_c is the set of users in country c. We use geometric mean to account for the skewed distribution of the entropy scores, something that the regular arithmetic mean cannot handle.

Table 9 shows Pearson correlations between question, answer, report entropy and Pace of Life ranks, where lower ranks mean higher Pace of Life. For both questions and answers, the overall Pace of Life ranks have positive correlations with question and answer entropy with $r = 0.67$ and $r = 0.37$, respectively. These positive relationships are seen in the Figures 40 and 41. We find positive correlations between walking speed rank, post office service time rank, and clock accuracy time rank with question entropy with $r = 0.48$, $r = 0.60$ and $r = 0.48$, respectively. For answers, we find positive correlations between post office service time rank, and clock accuracy time rank with entropy with $r = 0.38$ and $r = 0.29$, respectively. However, we do not find any statistically significant relationships between report entropy and Pace of Life ranks.

These results confirm that users from countries with a higher Pace of Life score show more temporally predictable asking and answering behavior in *YA*.

Table 9 Pearson correlations between question, answer, report entropy and Pace of Life rank. Lower ranks mean higher Pace of Life. p-values are indicated as: $p < 0.005(***)$, $p < 0.05(**)$, $p < 0.1(*)$.

	Entropy		
Pace of Life	Question	Answer	Report
Overall	0.67***	0.37*	0.18
Walking speed	0.48**	0.18	0.06
Post office	0.60**	0.38*	0.19
Clock accuracy	0.48**	0.29*	0.21

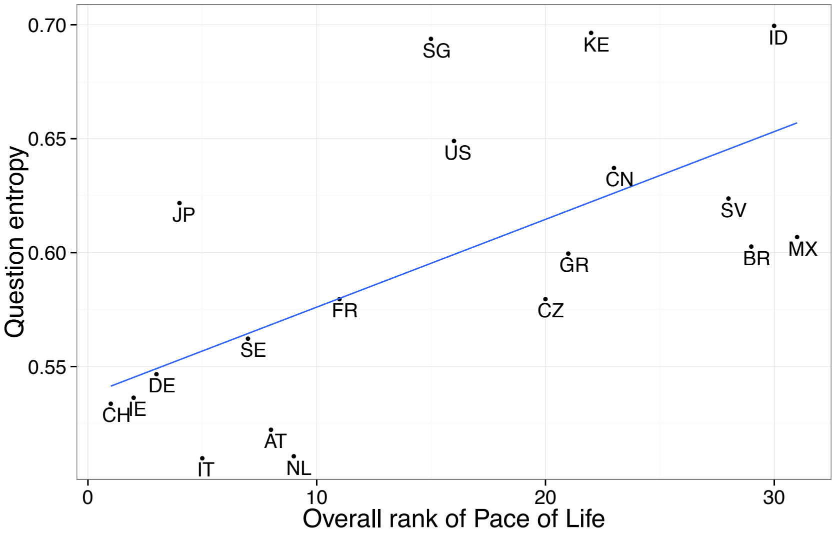


Figure 40 Pace of Life overall rank vs. average question entropy per country. Countries shown are the ones in our dataset for which a Pace of Life rank has been published. Countries are ranked in decreasing order of their Pace of Life value. A regression line is also shown.

6.3 Hofstede's Cultural Dimensions

In this section, we analyze a number of cultural dimensions in *YA* proposed by Geert Hofstede. We show how three cultural dimensions defined by Hofstede—individualism, power distance and uncertainty avoidance are manifested in the ecosystem of *YA*.

Hofstede's cultural dimensions theory is a framework for analyzing cultural variability. In his original model [129], Hofstede proposed four primary dimensions by surveying in the 1960s and 1970s a large number of IBM employees from 40 countries: power distance (PDI),

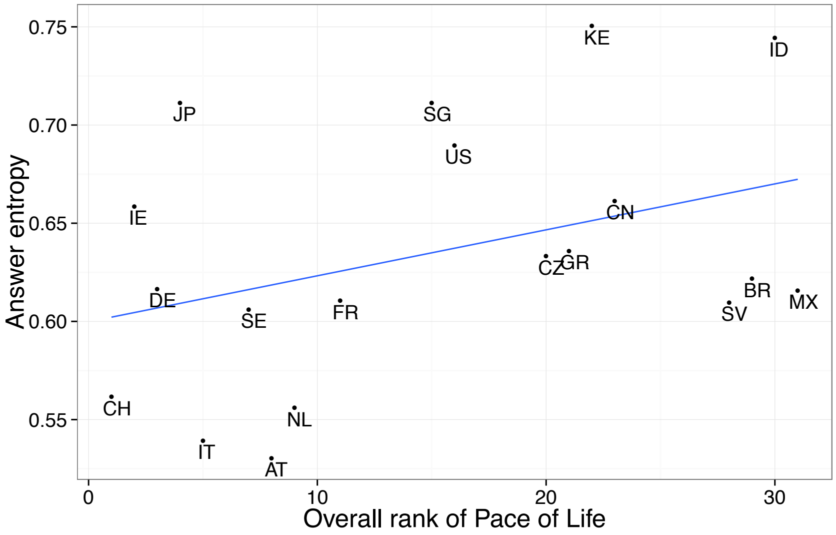


Figure 41 Pace of Life overall rank vs. average answer entropy per country. Countries shown are the ones in our dataset for which a Pace of Life rank has been published. Countries are ranked in decreasing order of their Pace of Life value. A regression line is also shown.

individualism (IDV), uncertainty avoidance (UAI) and masculinity (MAS). Later [130], he added two more dimensions: long-term orientation (LTO) and indulgence versus restraint (IVR). Three of the dimensions, individualism, power distance, and uncertainty avoidance, have been used in a number of recent studies of online behavior [104, 224, 239]. We also use these three Hofstede's cultural dimensions and relate them to a number of hypotheses in the context of *YA*.

6.3.1 Individualism (IDV)

Individualism is the extent to which an individual is integrated into a group. In individualistic societies (high IDV) such as the USA and England, personal achievements and individual rights are emphasized; an individual is expected to take care of only himself and his immediate family. In collectivist countries such as those of India, China, and Japan, individuals are expected to place the family and group goals above those of self. In this work, we investigate how individualism is related to users' contribution, (un)ethical behavior and privacy settings in *YA*.

The usage of the Internet takes time from a number of daily activities including face-to-face socialization. In collectivist countries, people are expected to give a fair amount of time on sociability, hence traditionally they seem to spend less time on the Internet compared to the people from the individualistic cultures [71]. In *YA* we expect that users from individualistic countries spend more time online, hence they can provide more answers and eventually they can contribute more to the community than their direct benefits from the community. We hypothesize the following:

$H2$: Users from countries with higher individualism index provide more answers.

$H3$: Users from countries with higher individualism index contribute more to the community than what they take away from the community.

We correlate the geometric mean of the number of answers posted by the users from a country with that country's individualism index (a higher score means higher individuality). We use geometric mean as an average because of the skewed distributions of the number of answers. In the calculation of the geometric mean, we exclude the users who have not provided any answers. We observe a positive correlation, shown in Figure 42, with $r = 0.46, p < 0.005$. This means that, on average, users from individualistic countries provide more answers.

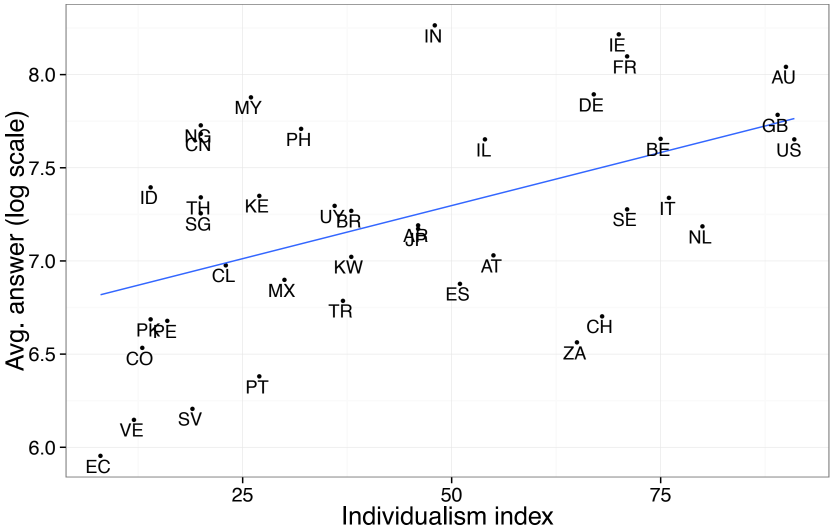


Figure 42 Individualism index vs. the average number of answers posted by users per country. A regression line is also shown.

To quantify users' contribution compared to their take away, we compute *yielding scores* of the users. The *yielding score* of a user is simply a difference between his contribution and his take away. For yielding scores, we consider *YA*'s point system, which awards two points for an answer, ten points for a best answer, and penalizes five points for a question:

$$\begin{aligned} \text{Yielding}_u &= f(contribution) - f(takeaway) \\ &= 2.0 * A_u + 10.0 * BA_u - 5.0 * Q_u \end{aligned} \tag{6.3}$$

where Q_u is the number of questions posted by user u, A_u is the number of answers posted by u, and BA_u is the number of best answers posted by u.

Finally, a country's yielding score $Yielding_c$ is defined as the geometric mean of all $Yielding_u$ computed for the users of each country c:

$$Yielding_c = \Big(\prod_{u \in U_c} Yielding_u\Big)^{\frac{1}{|U_c|}} \tag{6.4}$$

where U_c is the set of users in country c and we take only those users having yielding scores more than zero. We correlate a country's geometric mean of the yielding score with the country's individualism index and we obtain a positive correlation (Figure 43) with $r = 0.37, p < 0.05$. This result suggests that the more individualistic a country is, the more its users contribute to *YA* than what they take away from the community.

There might be multiple explanations about why users from individualistic countries contribute more to the community as reflected by hypotheses H2 and H3. One explanation is that individualistic cultures have a more favorable collaborative environment [254], so individuals feel the urge to contribute to the community. Another explanation could be that users from individualistic cultures simply want more points than collectivist cultures. As points are awarded for contribution (e.g., an answer earns two points) and participation (e.g., each login earns one point), users might be tempted to contribute more. In fact, we obtain a positive and significant

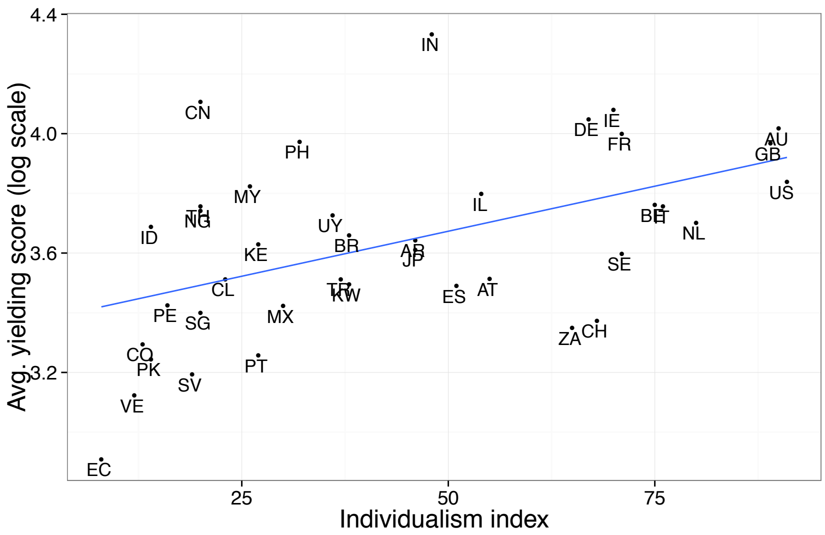


Figure 43 Individualism index vs. yielding score per country. A regression line is also shown.

correlation ($r = 0.42, p < 0.01$) between a country's points (calculated as geometric mean of the country's user points) and its individualism index. Finally, there might be other confounding factors (e.g., internet penetration) that affect the contribution of a country's users on the platform. Thus, it is difficult to confirm whether the users' behavioral differences on contribution are due to their cultural differences.

The degree to which a culture is collectivist or individualistic has an implication on its users' online (un)ethical behavior. For example, the more individualistic (less collectivistic) a culture, the lower the rate of software piracy [133] and online music piracy [155]. Personal rights are paramount in individualistic cultures, where people do not feel obligated to engage in group cooperation that involves conspiracy. Group cooperation and conspiracy are two key elements for the real world unethical behaviors such as corruption [223]. Triandis et al. [276] used Hofstede's individualism index and found that the countries with higher collectivist scores show the most corruption.

Based on this online and offline user unethical behavior that is influenced by culture, our intuition is that we could observe a similar trend in *YA*. In CQA platforms, the expectation is that users would provide helpful answers to posted questions. As such, users are required to follow the Community Guidelines and the Yahoo Terms of Service while answering. When users post bad answers, community members flag them. Later, human moderators check whether these flags are applied correctly or not. We expect that the more collective a culture is, the more probable the answers from its users will be flagged as abusive. Formally, we hypothesize that:

$H4$: Users from more collective (less individualistic) cultures have higher probability to violate CQA norms.

To this end, for each user u, we first calculate p_u, the probability that his answers violate community norms (and thus are correctly flagged by other users):

$$\mathrm{p}_u = \frac{\text{\# correctly flagged answers from u}}{\text{\# total answers from u}} \tag{6.5}$$

Finally, P_c, the geometric average of all p_u probabilities computed for each country c:

$$P_c = \Big(\prod_{u \in U_c} p_u\Big)^{\frac{1}{|U_c|}} \tag{6.6}$$

where U_c is the set of users in country c.

The Pearson correlation $r = -0.48, p < 0.05$ shows that the probability of abuses in answers provided by the users from a particular country is negatively correlated with that country's individualism index. Figure 44 indeed shows that the probability decreases with an increasing individualism index, meaning that if an answer comes from an individualistic country, it is less probable to violate community rules.

Next, we will discuss individualism and privacy concerns. Although online privacy concerns are global, the extent to which people perceive these concerns as real varies across cultures. For example, in the United States, privacy is a basic human right, endorsed by the American Bill of Rights, while Asian countries show little or no recognition on privacy in their legal systems [71]. A survey of 1261 Internet users from five big cities—Bangalore, Seoul, Singapore,

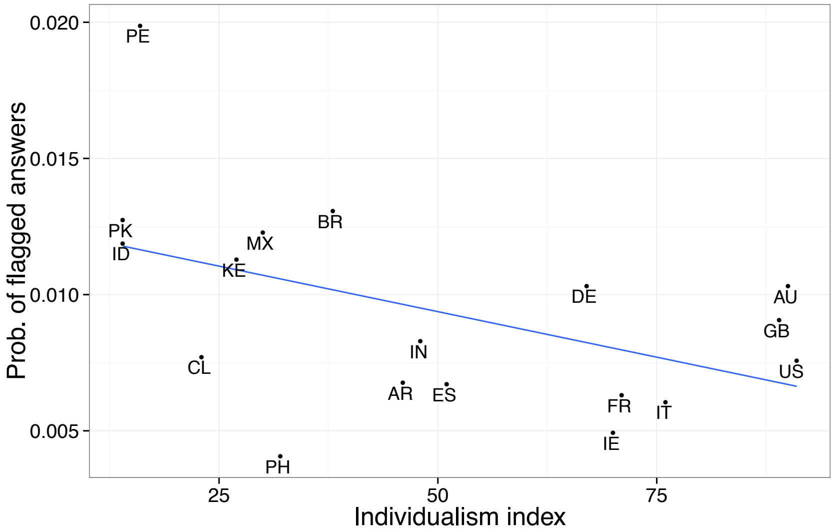


Figure 44 Individualism index vs. the probability that an answer from a country is correctly flagged. A regression line is also shown.

Sydney and New York—shows that Internet users from individualistic cultures are more concerned about privacy than those in collective cultures [52]. We expect that a similar trend also exists in CQA platforms. We hypothesize that:

$H5$: Users from higher individualism index countries exhibit higher level of concern about their privacy.

We use the modifications of the privacy settings on users' *YA* accounts as a proxy of privacy concern. In *YA*, privacy settings are typically available for users to personalize for content (questions or answers) and follower-followee network. Intuitively, privacy-concerned users would take the opportunity to change the default privacy settings. So, we consider the fraction of public privacy profiles in a country to draw a conclusion on how concerned its users are about their privacy. However, the default privacy in YA is public. It might be possible that many of the users in the public group are dormant: users who signed up, asked and answered some questions, and disappeared quickly. These users might skew the results of our study, thus, we only consider active users from our dataset— users who have asked and answered more than 10 questions during our observation interval. These active users are about 79% of our dataset. We note that our conclusions remain the same if we consider more active users by filtering users who have asked and answered more than 20 questions.

Based on Hofstede's Individualism index, the Hofstede Centre[3] has tagged countries as individualistic or collectivist. In our study, we use this classification. Figure 45 shows the percentage of user profiles with public privacy settings in a country, as function of the country's ranking in the collectivist and individualistic class. The figure shows that, on average, collectivist countries have a higher percentage of public profiles: collectivist countries such as Spain, Peru, Argentina, and Mexico have higher percentage of public profiles than individualistic countries such as United Kingdom, United States, Australia or Italy.

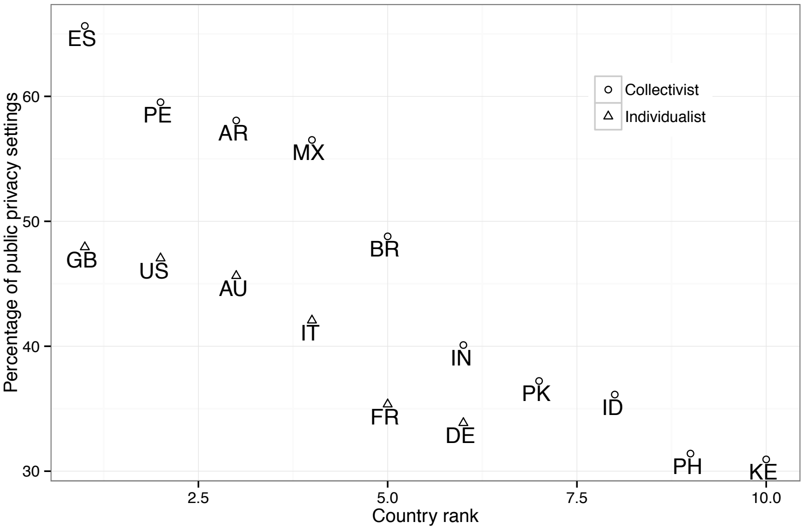


Figure 45 Percentage of public privacy settings vs. rank of collectivist and individualistic countries, respectively. Country ranks are based on the percentage of public privacy settings and they are separately done for collectivist and individualistic countries.

6.3.2 Power Distance Index (PDI)

PDI is the extent to which the less powerful members of an organization or society expect and accept that power is distributed unequally. This dimension sheds light on how a society handles inequalities among its members. In countries with high PDI, such as countries from Latin,

[3] http://geert-hofstede.com/countries.html

Asian, African and Arab world, everybody has a place in the social hierarchy and people accept the situation without questioning it. However, in Anglo and Germanic countries, which are low power distance countries, people seek distribution of power and ask for justifications of power inequality.

PDI essentially measures the distribution of wealth and power between people in a country or culture. In *YA*, we can use the indegree (number of followers) as a proxy of wealth and power. For example, the larger the number of followers users have, the larger an audience they have for direct communication. Higher indegree users are also found to be more central (thus more retained [152]) across a number of network centrality metrics [151]. Moreover, these users' questions are forwarded to more users, hence more likely to be getting an answer. A study [150] on *YA* shows that users receive more answers from close neighborhoods. Given the high number of questions that remain unanswered (42% in *YA* reported by a study [241]) in CQA platforms, bringing answers not only shows a user's potential capability, but also makes the platform mature and informative. Taking ideas from the unequal distribution of wealth and power in higher power distance countries, we expect that in *YA*, users from those countries also have inequality in their indegrees. Garcia et al. [104] have found similar indegree inequality in Twitter. We hypothesize the following in *YA*:

$H6$: Users from higher power distance countries show a larger indegree imbalance in follow relationships.

We correlate countries power distance index (higher index means power distance is high) with their users' indegree imbalance. A user's indegree imbalance is calculated as the difference between her friends' average indegree and her indegree. Finally, a country's indegree imbalance is the geometric mean of the indegree imbalance of its users.

For all countries, except Panama and Philippines, we obtained a positive indegree imbalance, meaning that for those countries, on average, a user's contacts have more contacts than the user. This supports a well-known hypothesis *friendship paradox* in sociology. The friendship paradox states that your friends have on average more friends than you have, however, most people think that they have more friends than their friends have [92]. It has been shown that the paradox holds for both Twitter [128] and Facebook [281]. Now we also show it for *YA*.

Figure 46 shows the relation between PDI and indegree imbalance (excluding Panama and Philippines). The figure indeed shows a positive correlation. We obtained a positive correlation $r = 0.65, p < 0.005$ between indegree imbalance and PDI for all countries (including Panama and Philippines). This supports the hypothesis that users from countries with higher PDI are more comfortable with indegree imbalance.

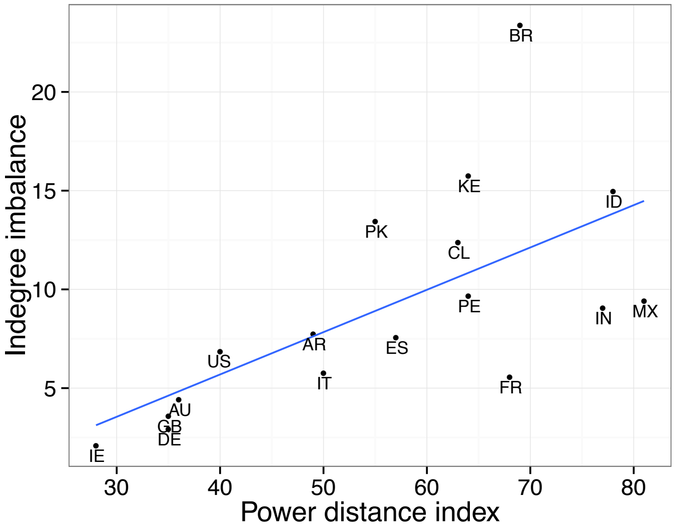


Figure 46 Power distance index vs. indegree imbalance. A regression line is also shown.

6.3.3 Uncertainty Avoidance Index (UAI)

UAI is the extent to which people feel uncomfortable with uncertainty and ambiguity. Individuals from countries exhibiting strong UAI tend to minimize uncertainty and ambiguity by careful planning, and enforcing rules and regulations. On the other hand, low uncertainty avoidance cultures maintain a more relaxed attitude in unstructured situations.

For example, Switzerland has a reasonably high uncertainty avoidance index (58) compared to countries such as Singapore (8) and Sweden (29). In fact, an online scheduling behavior study [239] on Doodle (http://doodle.com/) shows that Switzerland and Germany have a high advance planning time of 28 days. In YA, our related hypothesis is:

$H7$: Users from countries with higher uncertainty avoidance index exhibit more temporally predictable activities.

Figures 47, 48, 49 show the relationship between question, answer and abuse report entropy vs. uncertainty avoidance index, respectively. Note that a higher UAI means lower uncertainty and ambiguity. The negative relations in the figures indicate that users from countries with higher uncertainty avoidance index tend to have lower question, answer and abuse report entropies, thus they are more temporarily predictable. All the entropies have negative relation to uncertainty avoidance index: $r = -0.43$ for questions, $r = -0.55$ for answers, and $r = -0.51$ for abuse reports. All correlation values are statistically significant with $p < 0.05$.

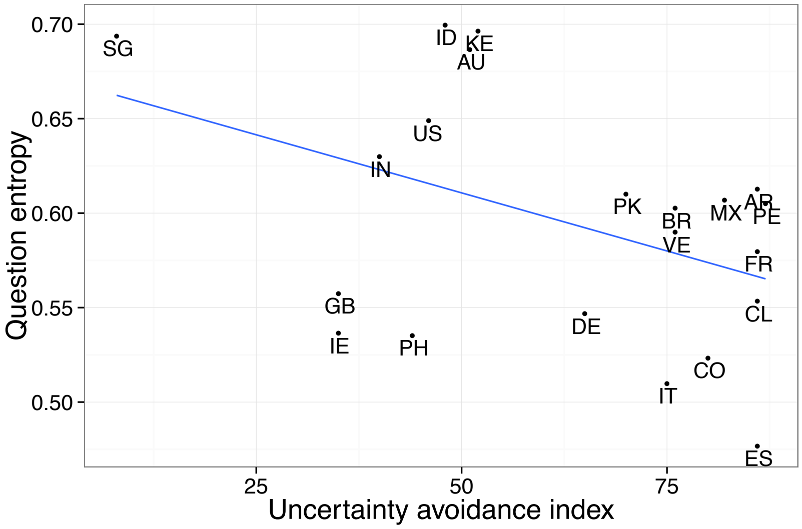


Figure 47 Question entropy vs. uncertainty avoidance index. Only countries having more than 300 users are plotted. A regression line is also shown.

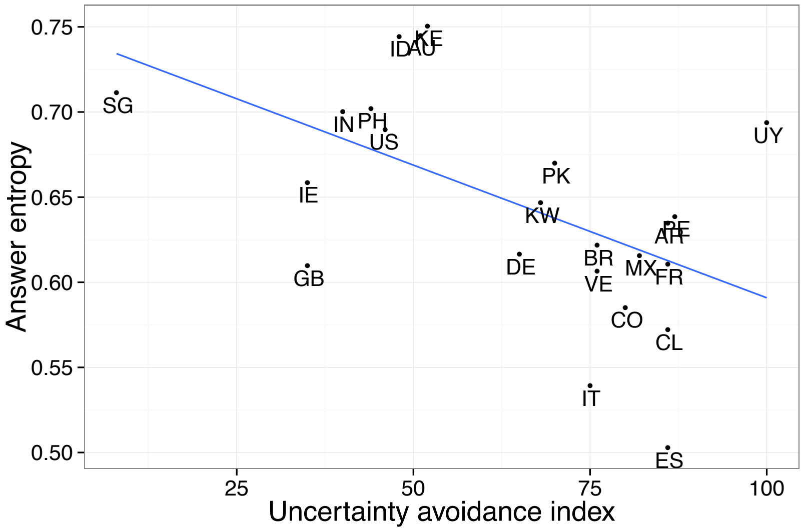


Figure 48 Answer entropy vs. uncertainty avoidance index. Only countries having more than 300 users are plotted. A regression line is also shown.

6.4 Summary and Discussions

In this chapter, we analyzed about 200 thousand sampled Yahoo Answers users from 67 countries. We studied users' behavioral patterns such as temporal predictability of activities, engagement, (un)ethical behavior, privacy concerns, and power inequality and how they compare with a number of cultural dimensions (Pace of Life, Individualism, Uncertainty Avoidance and Power Distance). We find that behavioral differences exist across cultures in *YA*. Table 10 shows a summary of all the hypotheses involving cultural indices and the results found.

Observing the global spread of information and communication technologies, researchers sometimes predicted that the online world would be converging into a "one-world culture" [175]. With the advent of the large-scale online behavioral datasets in the past decade from online platforms like Twitter, Facebook and Foursquare, researchers showed that the Internet does not have a homogeneous culture. Instead, country-specific cultural variations do exist. We showed the same non-homogeneity, but in a very different online context—community question answering.

We acknowledge that our study is observational and lacks controlled experimental ground truth data. Therefore, we cannot draw causal conclusions whether cultures shape the ecosystem

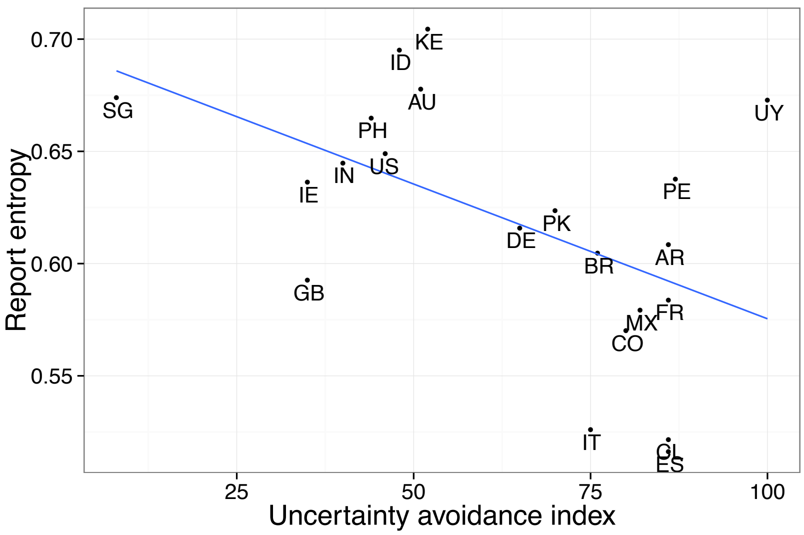


Figure 49 Report entropy vs. uncertainty avoidance index. Only countries having more than 300 users are plotted. A regression line is also shown.

of *YA*. However, our results hint at the importance of culture-aware CQA moderation. Note that CQA platforms like *YA* employ human moderators to evaluate reported abuses and determine the appropriate responses, from removing content to suspending user accounts. We find that collective cultures are more probable to provide bad answers. At a minimum, more attention of moderators are expected in these cultures to keep the environment clean.

We find that individualistic cultures are more engaged in YA, e.g., by providing more answers and contributing more than their take away. These results confirm the generalization that individualistic cultures are highly attracted to the Internet. Researchers often attribute the egalitarian, democratic nature of the Internet to this engagement [70].

The evidence of different engagement patterns and difference in pace of life across cultures in CQA platforms imply that some core functionalities such as *question recommendation* and *follow recommendation* could benefit from exploiting cultural factors. In *question recommendation*, questions are routed to the most appropriate answerers. To find out such answerers, factors such as followers, interests, question diversity and freshness [272] are considered. Our study suggests that including cultural variables such as individualism can be useful. For example,

Table 10 Pearson correlation coefficients in hypotheses related to pace of life, individualism, uncertainty avoidance and power distance. p-values are indicated as: $p < 0.005(***), p < 0.05(**), p < 0.1(*)$.

Pace of Life	**Correlation**
Users from countries with a higher Pace of Life score show more temporally predictable activities (asking, answering and reporting)	$r_q = 0.67$*** $r_a = 0.37$** $r_r = 0.18$
Individualism	**Correlation**
Users from higher individualism index countries provide more answers	$r = 0.46$***
Users from countries with higher individualism index contribute more to the community than what they take away from the community	$r = 0.37$**
Users from more collective (less individualistic) cultures have higher probability to violate CQA norms	$r = -0.48$**
Users from higher individualism index countries exhibit higher level of concern about their privacy	NA
Power distance	**Correlation**
Users from higher power distance countries show larger indegree imbalance in follow relationships	$r = 0.65$***
Uncertainty Avoidance	**Correlation**
Users from countries with higher uncertainty avoidance index exhibit more temporally predictable activities (asking, answering and reporting)	$r_q = -0.43$** $r_a = -0.55$** $r_r = -0.51$**

as users from collective cultures are less probable to answer, questions from those communities should be routed to a larger number of potential answerers.

Another variable, Pace of Life, could also be a factor in *question recommendation*. Our results show that users from countries with a higher pace of life are temporally more predictable. In those cultures, if questions are forwarded to answerers during the busy hours of the day (e.g., during office hours), the questions are less likely to get an answer. Solutions could include routing questions to a larger number of potential answerers, diversifying the set of answerers to include users from countries with a lower Pace of Life, or delaying routing for after work hours.

In the *follow recommendation*, CQA platforms recommend which other users one can follow based on shared interests, common contacts, and other related factors. We find that in YA, lower power distance countries show less indegree imbalance in follow relationships. For follow recommendation in those countries, users to be followed may be recommended to a user with the same level of indegree as them.

CQA platforms could also exploit cultural differentiations to improve targeted ads. Okazaki and Alonso [221] analyzed online advertising appeals such as “soft sell” appeal (that works by creating emotions and atmosphere via visuals and symbols) and “hard sell” appeal (that provides focus product features, explicit information, and competitive persuasion) across a number of cultures. They found individualistic cultures like the USA are more attracted to “hard sell” appeal, where collective cultures like Japan are attracted to “soft sell” appeal. Ju-Pak’s study [142] also confirms that fact-based appeal is dominant in the USA, but text-limited, visual layouts are popular in collective cultures like South Korea. Linguistic aspects in the ads might also be important. For example, focusing on ‘I’, ‘me’ in individualistic cultures and ‘us’ and ‘we’ in collective cultures. Finally, CQA sites could leverage cultural variations in their platforms by, for example, placing textual, informative feature ads to users from individualistic cultures and visual and symbolic ads to users from collective cultures.

CHAPTER 7: AEGIS: A SEMANTIC IMPLEMENTATION OF PRIVACY AS CONTEXTUAL INTEGRITY IN SOCIAL ECOSYSTEMS[1]

Our analysis of *YA* users' privacy settings in Chapter 5 reveals that 87.20% of users have platform-provided default privacy settings. Our work on bloggers' retention in a blog social network (*Blogster*) finds that 92% bloggers do not change their default privacy settings [152]. These results on default privacy are consistent with a variety of other social networks: it has been shown many times that while users are invited to change the default privacy settings, in reality very few do it. For example, more than 99% Twitter users retained the default privacy setting where their name, location, biographical information, website, and list of followers are visible [162]. Other studies [1, 116, 163] show that the majority of Facebook users have default or permissive privacy settings.

More worrisome, when the default settings are not matched with user preferences, most of the time they tend to be more open and visible, exposing the user provided content to more users than expected [184]. Users' unwillingness to change the default policy is sometimes exacerbated by the complexity of the process; default privacy controls are too cumbersome to properly understand and use [168, 180, 263]. Users' online privacy, in general, is already a hot issue due to lack of formal framing [218]. Results from our studies in *YA* [148] and Blogster [152], as well as supports from the literature, warn again about the importance of correct default settings.

Default privacy related privacy concerns have been further aggravated in *social ecosystems*. Social ecosystems [274] refer to the collection of rich datasets of user-to-user interactions in support of social applications. This data is collected from Internet-mediated social interactions

[1] Much of the work in this chapter was first published in [145, 146]. Permission is included in Appendix A.

(such as declared relationships in online social networks or tagging/contributing content in user-generated content platforms), from public profiles (to infer homophily relationships), and from phone-recorded real life interactions (such as co-location sensing and activity identification). Social ecosystems have enabled a large set of social applications, such as recommender systems [15, 49, 110], email filtering [109, 157], defending against Sybils [44, 307] and against large-scale data crawls [207]. The novel scenarios activated by social ecosystems, however, raise serious concerns regarding user privacy.

The primary aspect of social ecosystems, that of aggregating data from various sources to provide it (possibly processed) to a diversity of applications, significantly amplify the privacy concern. First, aggregated data from different contexts of activity presents a more complete and possibly uncomfortable picture of a person's life. Second, data is to be exposed to a variety of applications, themselves from different contexts of activity, from personal to professional.

Numerous solutions addressed privacy in social ecosystems, typically in the context of a particular system [17, 91, 97, 267] or for particular application scenarios [69, 95, 231]. Little addressed, however, is the setting of a *default* privacy policy that protects the user and, at the same time, allows the user to benefit from application functionality.

The privacy challenge is fundamentally due to the lack of a universal framework that establishes what is right and wrong [218]. Nissenbaum proposed such a framework in her formulation of privacy as contextual integrity [216]. To the best of our knowledge, one line of work approaches privacy as contextual integrity by proposing a formal language for expressing generic privacy expectations [22]. In this chapter, we ask the following research question:

R_9 : Can we generate default privacy policies based on contextual integrity theory that restrict user information to be shared or transferred inappropriately?

We employ semantic web techniques to implement Nissenbaum's framework for defining privacy as contextual integrity, with a specific focus on defining application and platform-independent default privacy settings. To this end, we propose Aegis, an extensible, fine-grained privacy model for social ecosystems based on semantic web technologies. The model implements the basic concepts of Nissenbaum's privacy framework: social contexts, norms of appropriateness, and norms of distribution. It builds on ontologies used to encode social data and implicitly represent social contexts, and on Resource Description Framework (RDF) statements/SPARQL queries to define and verify access to data.

The rest of the chapter is organized as follows. Section 7.1 introduces the contextual integrity theory and discusses its relevance to social ecosystems. Section 7.2 describes the system and data models, and the system architecture. We present our policy specification in Section 7.3. Section 7.4 presents our prototype implementation and experimental evaluation. Section 7.5 concludes the chapter with discussions.

7.1 Privacy as Contextual Integrity in Social Ecosystems

While notoriously difficult to define [217], privacy is understood as an individual's right to determine to what extent her data can be communicated to others. Privacy is typically seen not simply as the absence of information about us in the minds of others, but rather as the control we have over information about ourselves [99, 238].

Social ecosystems, which combine users' social information from diverse sources and incorporates richer semantics, pose a daunting task in terms of privacy enforcement. It has to exercise a more complex representation of users' social world, ranging from object-centric domains (e.g., common preferences) to people-centric domains (e.g., declared friendship relationships). Privacy-preserving default policy generation in such a complex system could be leveraged by contextual integrity, a social theory-based account of privacy proposed by Nissenbaum [216]. Instead of defining the term "privacy", Nissenbaum proposes a reasoning framework for privacy as contextual integrity, where privacy is seen as neither a right to secrecy nor a right to control, but a right to an appropriate flow of *information about an individual* (referred to as "personal information").

Nissenbaum's account of privacy as contextual integrity is based on two non-controversial facts. First, every transfer of personal information happens in a certain social context and all areas of life (and online activity makes no exception [218]) are governed by context-specific norms of information flow. Second, people move among a plurality of distinct contexts, thus altering their behavior to correspond with the norms of those contexts, aware to the fact that information appropriately shared in one context becomes inappropriately shared into a context with different norms. For example, it is appropriate to discuss romantic entanglements with friends, financial information with banks, and work-related goals with co-workers, but sharing romantic experiences with the bank is out of place.

On the basis of these facts, Nissenbaum suggests that contextual integrity is maintained when two types of norms are upheld: *Norms of appropriateness* and *Norms of distribution*. Norms of appropriateness circumscribe the type of information about persons that is appropriate to reveal in a particular context. So, it is appropriate to share medical information with doctors, but generally not appropriate to share it with employers. Implemented in social ecosystems, this type of norm specifies *where* context-specific data can be communicated. For example, if Alice is a colleague of Bob in the professional context, then requests from Alice regarding Bob's gaming context such as the games owned by Bob should be denied, as the requests do not comply with the norms of appropriateness.

Norms of distribution cover the transfer of a third party's personal information from one user to another. In a social ecosystem, the norm of distribution suggests a default policy that restricts the distribution of information. For example, if Alice and Bob have a shared content—e.g., Bob's picture that he shared with Alice—then a request from Charlie to Alice regarding the content should not succeed without Bob's consent, even if Alice owns Bob's picture now.

7.2 System Model and Architecture

Nissenbaum's framework is articulated for protecting the citizen from an overly curious government. For the digital context, her approach works best as a default privacy policy, which is precisely the focus of this chapter. Thus, our proposed system, Aegis, enforces default policy as contextual integrity by modeling two assumptions from real world. First, information is always tagged with the context in which it is revealed. Second, the scope of privacy norms is always internal to a context. To implement this, Aegis implements the constructs of user roles and actions, resources, contexts, and privacy norms.

7.2.1 System Model

Similar to Dey et al.'s definition of a context [75], we define the social context of a user as the collection of social information that describes the user in a domain. For example, data about Bob's education, skills, and LinkedIn connections describe Bob's *Professional* context.

Our system model is defined by the following:

1. There is an unrestricted set of disjoint social contexts in the system;

2. A user belongs to only one social context at any time;

3. A user can have one or more roles in every social context s/he is part of;

4. Each piece of data (resource) is assigned to only one context; however, users can share a resource with other users, in which case the resource is replicated in each of the other users' current contexts;

5. A request for a resource is made on behalf of the requester's role in the particular context in which the requester is when the request is made;

6. A request specifies an action, which could be *read*, *write*, *delete* or *replicate* to another user's ownership.

Note that in real life users can be simultaneously part of multiple contexts: for example, Alice is both a friend and a colleague for Bob. However, at any given time, only one of these contexts will be considered, perhaps the prominent one given the physical environment (e.g., at work) or based on a system-wide ranking of contexts (e.g., work has higher priority over friendship, to limit sensitive data exposure).

Implementing contextual integrity in the default privacy policies is thus reduced to implementing the norms of appropriateness and distribution in this system model.

7.2.2 Modeling Social Contexts and Roles

We model social contexts, and therefore the entire social ecosystem (consisting of a set of social contexts), based on ontologies. An ontology is a set of entities, instances, functions, relations and axioms, and is used as a vocabulary for expressing the knowledge of a domain.

The traditional advantages of using ontologies apply in this case as well: first, an ontology provides a common vocabulary, thus it ensures formal and structured representation of users' contextual data. Second, using ontologies provide semantic interoperability, thus data can be used by a variety of applications. Third, high-level logic inferences are possible as data model have semantics associated with it. For example, if Bob has content in his professional context (*contents subClassOf ProfessionalContext*) and *recommendations* are content, then *recommendations* are inferred to belong to Bob's professional context. Finally, a social ecosystem can be built incrementally by adding new context ontologies. We can also reuse existing web ontologies from different domains to meet the demand of an exhaustive scale social ecosystems ontology. For example, an ontology [210] is already available to model bloggers' interest in a blogging community.

To represent the data model, we classify online social contexts into entity classes (e.g., friendship context, professional context, blogging context). The context classification is inherent in the modeling process because context definition depends on underlying entities and relationships among the entities.

Our context ontology is divided into an upper ontology and domain-specific ontologies. The upper ontology is a high-level ontology which captures general contexts (e.g., Friendship, Professional, Gaming). The domain-specific ontologies are collections of low-level ontologies which define the details of general contexts and their properties in each sub-domain.

Figure 50 shows a class hierarchy of the entities considering some online contexts of a user, where the top level class (root) is the context itself. All generic contexts are subclasses of the root context entity and all domain-dependent descriptors (classes, properties) have some common properties to inherit from the root. The lower level sub-classification expresses the domain dependence of the contexts.

The addition of new social contexts to the ecosystem happens naturally, with the implementation of new sensors for new social signals (see next section): the developers of social sensors have to be aware of the ontology of the social contexts to which the sensors report, in order to maintain structural data representation. Another way to extend the social ecosystem is by extending a social context itself when new relevant social signals become available: for example, Facebook recently added a service called Gifts which allows users to buy presents for their friends. Consequently, the social context model needs to be adaptive to accommodate additions of new contexts. Ontologies help in designing a scalable context model.

Figure 51 gives an example of three contexts of a user's digital world: Professional, Friendship and Gaming. The representation of the ontology is *person*-centric which gives a user-oriented viewpoint of the data model. The three large circles in the model are the contexts; each circle encodes context-specific knowledge and they are subclasses of Context.

Roles are modeled as relationships: for example, *isColleagueOf* in Alice's ecosystem specifies that Bob has the role of a colleague in her professional context. Roles, as relationships,

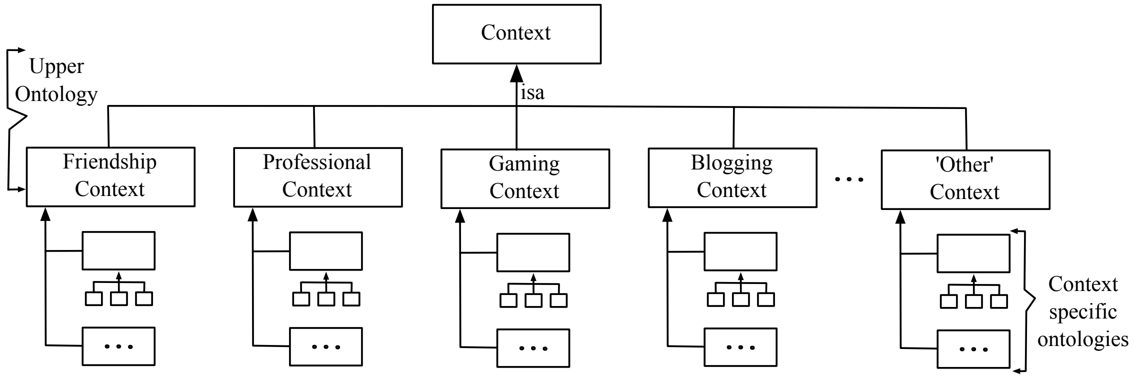


Figure 50 Context entities and their domain dependent elements.

are thus asymmetrical: Charlie might be a follower in Alice's followers ecosystem but Alice might not be Charlie's follower.

We use OWL [209] to model social contexts. OWL is more expressive than other ontology languages such as RDFS [209]. Moreover, W3C Web Ontology Working Group has defined OWL from an existing rich language DAML+OIL[2].

7.2.3 Aegis Architecture

Our general architecture fits the Social Hourglass infrastructure introduced in [135]. The focus of this work is the Privacy Management Layer, presented in Figure 52. It receives input from users' personal aggregators and outputs privacy-compliant social data to applications. The input from each user's personal aggregator is a labeled, directed ego net, that represents the user's recorded social interactions with other users, with each type of interaction semantically tagged.

A brief explanation of the *Social Data Acquisition and Aggregation Layer* at the bottom of Figure 52 (that belongs to the social hourglass infrastructure) is presented next. (For more details we refer the reader to [135].) A social sensor is an application running on behalf of a user on a user's device as an independent application or in the browser) and observing one or more social signals (for example, Facebook interactions of the user with other users). It reports

[2] http://www.daml.org/2000/12/daml+oil-index.html

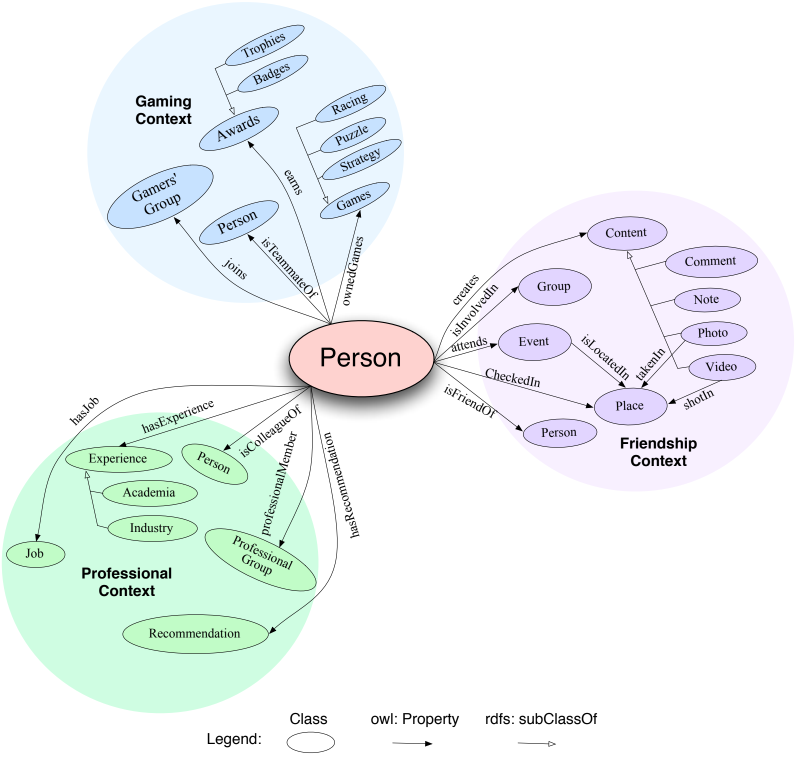


Figure 51 A partial definition of social ecosystems ontology considering Professional, Friendship and Gaming contexts. A circle represents a context, which contains context specific classes (e.g., declared social groups).

processed social data in the form of <contact ID, type of interaction, strength of interaction> to the user's aggregator. The aggregator runs on the user's trusted device (e.g., mobile phone or home computer, but not on a shared computer or a commercial service). It processes all such information received from the social sensors deployed on behalf of its user and reports an aggregated and personalized social edge to the *Social Data Management Layer* and to the *Contextual Policy Definer*. For user *ego*, this social edge is of the form <ego, alter, context, weight>, where *alter* is a user *ego* interacted with in *context* with the interaction strength *weight*. Social data management can be implemented by various solutions; to provide *surveillance privacy* protection [120], distributed solutions such as Prometheus [158] can be used.

Social sensor design and implementation are context specific: for example, a LinkedIn sensor observes its user's professional data and a Facebook sensor observes the user's friendship data based on the ontology shown in Section 7.2.2. In addition to requirements related to sensor accuracy and performance, sensor design should address the following. First, a particular sensor can target one context only (thus, report one label only), but is capable of collecting data from different social signals. For example, a gaming sensor could collect gaming related data from multiple services, e.g., Stream[3] and Giantbomb[4]. Moreover, by using the ontology vocabulary, sensors should be able to distinguish context-specific data from the wealth of social data existing in a service. Second, sensors should be able to cope with changes in ontology and act immediately.

The Privacy Management Layer in Figure 52 is responsible for managing and enforcing privacy policies, and thus for extracting and applying the default policies as well. This component communicates with the Social Data Management Layer which implements social contexts and roles.

[3] http://www.steamcommunity.com/

[4] http://www.giantbomb.com/

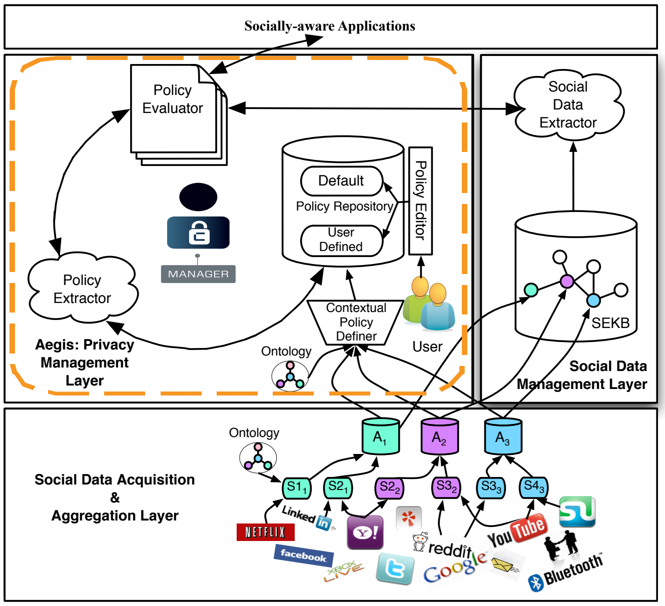


Figure 52 A layered architecture of social data collection, personalization and management for socially aware applications along with Aegis, as a form of privacy management layer.

The *Contextual Policy Definer* generates default access control policies based on a social ontology and the contextual integrity norms and stores them in the *Policy Repository*. Policies in the *Policy Repository* can be edited with the GUI-based *Policy Editor*. The *Contextual Policy Definer* generates default policies based on the following rule: only roles in a user's social context are allowed access to the user's data associated with that particular context. An example of a default policy extracted with this rule is the following: all users with a *Colleague* role in Bob's *Professional* context can access (all) his data associated to the *Professional* context. Our policy model is granular; it defines a policy for every resource covering all the contexts a user could belong to.

The *Policy Manager* consists of extractor and evaluator for handling access requests. In particular, any access request is intercepted by the *Policy Evaluator*, which evaluates the policy. Permitted access requests are finally fulfilled by returning data from the social ecosystem knowledge base (SEKB) through social data extractor. The policies are stored in the policy repository and the policy extractor extracts policies from the policy repository.

7.3 Policy Specification

A policy is defined as a set of RDF statements. As shown in the architecture, the contextual policy definer generates policies that obey the two information norms of contextual integrity: norms of appropriateness and distribution.

Let us take as example a policy generated by the policy definer for the resource *groups* in the *Professional* context: Bob's colleagues can read his professional group involvement in the *Professional* context. The policy can be formalized as the SPARQL query (Figure 53), <Policy>, where the prefixes `p:` and `se:` represent the namespace of the policy model and of the social ecosystems model, respectively:

```
<Policy>
ASK
where {
 ?req rdf:type p:requestor.
 ?req p:allowed p:read.
 p:read p:performedOn Bob.
 ?req se:isColleagueOf Bob.
 Bob se:professionalMember ?group.}

<Augmented Policy>
ASK
where {
  Alice rdf:type p:requestor.
  Alice p:allowed p:read.
  p:read p:performedOn Bob.
  Alice se:isColleagueOf Bob.
  Bob se:professionalMember ?group.}
```

Figure 53 A sample norms of appropriateness policy.

A basic access request is a triple *<rstr, rsc, act>*, where *rstr* is a user who requests the access (e.g., an instance of *se:Person*), *rsc* is the resource requested (e.g., *se: Photo*), and *act* is the type of action (read/insert/delete).

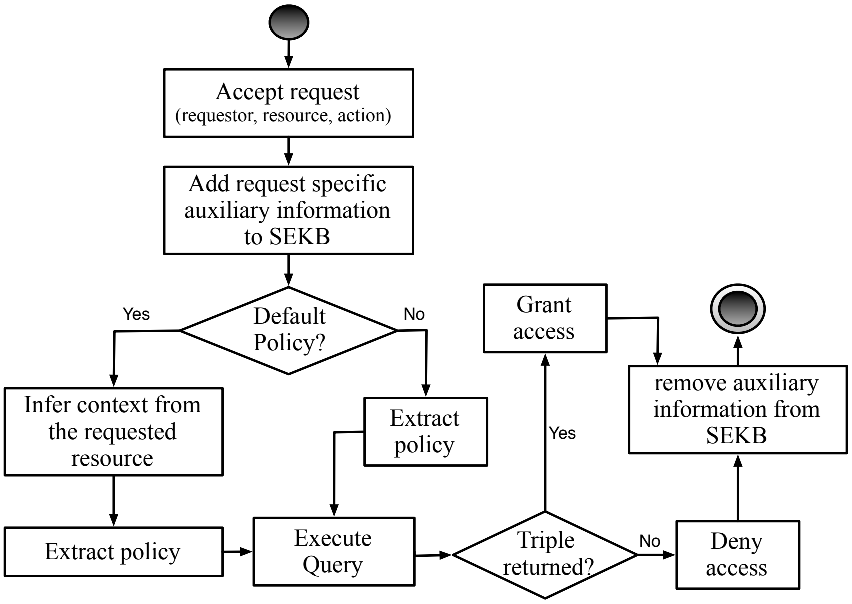

Accept request
(requestor, resource, action)
Add request specific auxiliary information to SEKB
Default Policy?
Yes
No
Infer context from the requested resource
Extract policy
Extract policy
Execute Query
Triple returned?
Yes
No
Grant access
Deny access
remove auxiliary information from SEKB

Figure 54 Request handling process.

When a request such as “Alice wants to see Bob’s professional group involvement” comes to the policy manager, the predefined policy variable `?req` will be replaced by Alice as shown by the augmented policy. The policy evaluator will temporarily insert policy-related auxiliary RDF statements to the knowledge base, such as the first three statements of the augmented SPARQL query, and executes the query over the modified knowledge base. The above policy representation states that the access request will be granted if Alice and Bob are colleagues. The same access request from Bob’s teammate in the *Gaming* context will be denied because of lack of appropriate triples in the SEKB, thus implementing the norm of appropriateness.

Similarly, the system will disallow access to a resource that is shared or co-owned with someone, upholding the norms of distribution. For example, the policy in Figure 55 restricts Charlie’s access to Bob’s photos that he previously shared with Alice.

Our policy representation is granular, as it allows policies for each resource. For the request, the policy manager will infer the context and will decide whether the default or personalized policy will be enforced (see policy evaluation flow chart from Figure 54). As in the data model we have hierarchy among classes (which eventually define resources) and a group of classes

```
<Policy>
ASK
where {
 ?req rdf:type p:requestor.
 ?req p:allowed p:read.
 p:read p:performedOn Bob.
 ?req se:isFriendOf Bob.
 Bob se:hasPhoto ?photo.
 ?photo se:status se:notShared}
```

Figure 55 A sample norms of distribution policy.

belong to a context, we can infer the context from requested resource. Note that a default auto generated policy could be personalized by the user and in this case, the personalized policy will be evaluated. For example, from a requested resource *recommendation*, a context inference is possible from the SPARQL query (Figure 56) to knowledge base:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-
rdf-syntax-ns#
PREFIX rdfs:<http://www.w3.org/2000/01/
rdf-schema#
PREFIX se:<http://www.dsg.cse.usf.com/se>

SELECT ?superClass
Where {
    se:requestedResource rdfs:subClassOf
?superClass .}
```

Figure 56 Context inference.

The policy engine will extract the policy of the inferred context and execute it.

7.4 Experimental Evaluation

We have implemented a prototype of Aegis in Java Platform Standard Edition 6 (Java SE 6). We used the capabilities offered by Jena[5] to implement both the knowledge base and the policy manager. Jena is a framework for building semantic web applications, and provides a col-

[5] http://jena.apache.org/index.html

lection of tools and Java libraries to develop semantic web and linked-data applications, tools and servers. Jena is currently the most comprehensive framework to manage RDF and Web Ontology Language (OWL) data in Java applications as it provides APIs for RDF data management, an ontology API for handling OWL and RDFS ontologies and a query engine compliant with the SPARQL specification. We leverage TDB[6] for persistent storage of knowledge base.

Aegis was deployed and evaluated on a machine equipped with 2 GHz Intel Core i7 processor, 4 GB 1333 MHz DDR3 RAM, Mac OS X Lion 10.7.5 operating system, and Java 1.6 runtime environment.

Our evaluation of the prototype implementation of Aegis had the following objectives. First, we wanted to evaluate the performance of the policy engine in executing default policies for realistic workloads with a realistically large number of users. For this, we chose three real social network datasets from different domains and experimented with default policy enforcement. Second, we wanted to investigate the scalability of the policy engine in executing default policies. Ideally, the policy engine should scale well with the size of the social ecosystem knowledge base. Finally, we wanted to measure the overhead induced by default policies.

7.4.1 Experimental Setup

We constructed social ecosystems knowledge base from three different real networks. Table 11 presents a summary of these datasets.

- soc-Slashdot0811 (from [172]): a network of friend/foe links between the users of Slashdot, a news website which features user-submitted and editor-evaluated technology-oriented news. Using Slashdot Zoo feature, users can tag each other as friends or foes.

- BlogCatalog (from [308]): a blogging website where registered users can create online profiles, post blogs, and automatically receive blogging updates from the users with whom they have declared “friend” relationships.

[6] http://jena.apache.org/documentation/tdb/index.html

- Facebook (from [286]): a highly popular online social network. The dataset contains friend links of the users.

To provide test cases, we selected 13 sizes ranging from 100 to 70,000 users from the above networks. To create a sub-graph of each size, we randomly picked a node as a seed in a network and applied snowball sampling algorithm. Although snowball sampling is biased toward high-degree nodes, it preserves the topological structure of a graph [136]. For each sampled sub-graph, we created a SEKB containing nodes of type *Person* and the relationships among them. More specifically, a ego's (user) connections are randomly labeled according to the data model (*se: isFriendOf, se: isColleagueOf, se: isTeammateOf*) to abstract a user's social ecosystem and contexts (*Friendship, Professional* and *Gaming*). Also, we added users *FriendshipGroup, ProfessionalGroup* and *GamingGroup* using a random string generator as resources in relevant contexts to invoke different test cases.

We considered two types of responses: (type1) positive authorization access control response and (type 2) negative authorization access control response. Type 1 accesses are allowed, while type 2 are denied by the default policies. To this end, we generated two types of access requests. They are as follows: User U1 belongs to the context C1 of user U2 and she requests U2's resource R1 from the same context C1. And, User U1 belongs to the context C1 of user U2 and she requests U2's resource R2 from different context C2. For each sample size of each dataset we evaluated both requests 10 times and report the average evaluation time. Moreover, we performed the same experiments with no policy enforcement to measure the policy enforcement overhead.

Table 11 Summary of the real networks used.

Network	Num. of Users	Num. of Edges
soc-Slashdot0811(Slashdot)	77,360	905,468
BlogCatalog	88,784	4,186,390
Facebook	63,731	1,545,686

7.4.2 Results

The performance results of the policy engine in executing default policies are shown in figures 57, 58, 59. They show positive (figure 57), negative (figure 57) authorization access time, and the number of requests answered by the policy engine per second (figure 59) for random authorizations. Our observations are as follows:

First, for all datasets and both types of authorizations, the time needed to fulfill access requests increases linearly with the size of the social ecosystem knowledge base (SEKB). As such, inference time of the default policies vary according to the size of the SEKB.

Second, for the same size of SEKB, positive and negative authorization take about the same time. Intuitively, a positive authorization should take less time than a denied request due to less scanning in the knowledge base. To assess the significance of the time difference, we ran a two sample *t*-test in which we compared the time taken for positive and negative authorization time for all sizes of SEKB. A *t*-test determines if two sets of data are significantly different from each other [317]. We obtained a *p*-value of 0.96, thereby confirming that the difference is not statistically significant. This is due to the implementation of the semantic data store, TDB: data structures in TDB use TDB B+Trees, a custom implementation of threaded B+Trees. The threaded nature implies that long scans such as negative authorizations of indexes (it uses triple and quad indexes) proceeds without needing to traverse the branches of the tree.

Third, for up to $10,000$ users in the SEKB, both accepted and denied access request execute fast on our tested machine (tens of milliseconds). However, as the knowledge base increases with the number of users, performance decreases. This is more visible for the BlogCatalog dataset, which has about three times more edges per node than the other datasets (see Table 11). This behavior is due to the stress SEKB puts on memory: a denser graph requires more memory, thus with the increase in the number of users represented, penalties related to swapping will take place. Obvious solutions to this performance limitation include 1) increasing system memory to realistic capacity for an in-production server and 2) employing distributed solutions for SEKB data management.

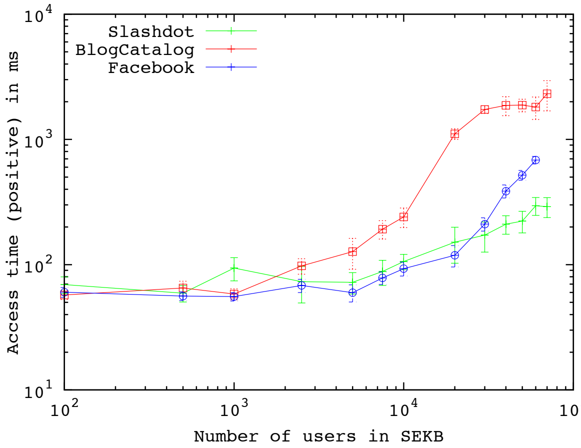


Figure 57 Access time for positive authorization.

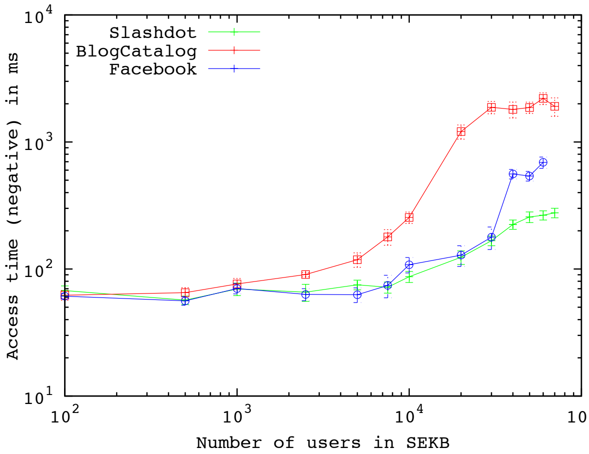


Figure 58 Access time for negative authorization.

To evaluate the overhead introduced by the policy engine for executing default policies, we tested the time needed to execute request with and without default policies in place. For each sampled size, we took the average access time for positive and negative authorizations both with and without default policies. Figure 60 shows the comparison. The difference between access requests with and without policies ranges from $3.17ms$ to $12.06ms$. To assess the significance of this overhead, we ran a two sample *t*-test in which we compared the access time with and with-

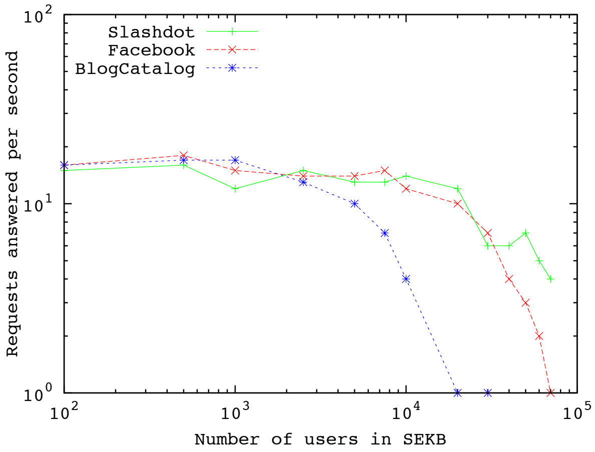


Figure 59 Number of requests answered per second.

out default policies. The *p*-value of this *t*-test is 0.81, which implies the overhead is statistically insignificant. So, we conclude that the action of default policy enforcement does not impose a significant burden on social ecosystems.

One of the limitations of the workloads is that they only contain ego-nets and social groups. However, a social ecosystem ontology is a more diverse collection of entity types and relations (as shown in Figure 51). A long chain of context inferences, such as “X is a photo, which is content, and the content belongs to the friendship context” will likely take longer time. Moreover, overlapping contexts (such as professional and friendship and gaming) will create denser ego-nets, hence more memory required per user. Consequently, the scalability plots shown in Figure 60 will change, also function of the available physical memory. However, the limited availability of appropriate real-world traces prevented us from doing more sophisticated performance analysis.

7.5 Summary and Discussions

In this chapter, we have proposed a privacy model for social ecosystems based on the semantic web standard. The privacy model leverages contextual integrity for generating default policies that protect user’s information from other users. We designed an architecture in sup-

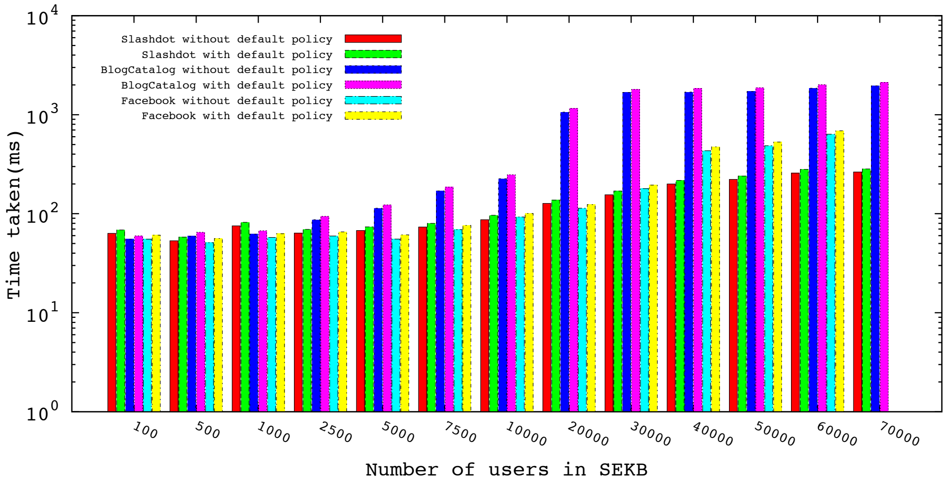


Figure 60 Performance overhead of the policy engine with and without default policy enforcement.

port of the proposed privacy model, demonstrated its feasibility by building Aegis, a prototype implementation, and evaluated its performance and scalability using three large real networks. The experimental evaluation shows that our system scales well, and policy enforcement does not impose significant overhead.

Aegis addresses “social privacy” [120] problems such as those that emerge through the necessary renegotiation of social spheres as social interactions get mediated by OSN providers. Social privacy problems occur when access to data is inappropriately protected due to wrong default or personalized settings. Often the default settings serve the business model of the service provider rather than the user’s interests, following the “opt out” model. Aegis mitigates social privacy threats by generating default privacy policies that restrict user information to be shared or transferred inappropriately. At the same time, Aegis does not restrict users from choosing personalized and maybe relaxed privacy settings.

Although our privacy model is designed for targeting user’s aggregated social data, the model is generic enough to be used in existing online social networks. For example, Google+ and Facebook allow users to select the type of relationship with another user. This information can be leveraged to provide higher granularity in social privacy and to implement privacy as contextual integrity for default privacy settings.

While Aegis addresses social privacy, it may aggravate *surveillance* and *institutional* privacy. Surveillance privacy threats emerge when users' social interactions and personal information are leveraged by authorities or OSN service providers. Institutional privacy [237] refers to those privacy problems related to users losing control and oversight over the aggregation, processing and mining of their online social information. The aggregation of social data in social ecosystems and the ontology-based labeling (thus, the addition of processed information) creates new sensitive data that would not have been directly available. For example, users' context-specific data (such as work, personal, etc.) would increase the accuracy of user profiling to an overly curious, possibly hostile political regime. These problems are alleviated by implementing the social ecosystem as a distributed architecture, as in [158]. A distributed architecture eliminates the need for a central, omniscient authority that is in a privileged position to observe all the activity in the system.

One of the limitations of our work is that we could not experiment with a real social ecosystem due to the unavailability of users' data from multiple sources. Ideally, a social ecosystem should be an aggregation of social data from various social network platforms (e.g., Facebook, LinkedIn, Steam). Instead, we took three large networks and constructed social ecosystems from those networks. Experiments on a real social ecosystem would give more insights on our system.

CHAPTER 8: A SURVEY OF PRIVACY AND SECURITY IN ONLINE SOCIAL NETWORKS[1]

In this chapter, we provide a comprehensive review of solutions to privacy and security issues in OSNs. While previous literature reviews on OSN privacy and security are focused on specific topics, such as privacy preserving social data publishing techniques [313], social graph-based techniques for mitigating Sybil attacks [305], or OSN design issues for security and privacy requirements [310], we address a larger spectrum of security and privacy problems and solutions. First, we introduce a taxonomy of attacks based on OSNs' stakeholders. We broadly categorize attacks as attacks on users and attacks on the OSN and then refine our taxonomy based on entities that perform the attacks. These entities might be human (e.g., other users), computer programs (e.g., social applications) or organizations (e.g., crawling companies). Second, we present how various attacks are performed, what counter-measures are available, and what are the challenges still to overcome.

8.1 A Taxonomy of Privacy and Security Attacks in Online Social Networks

We propose a taxonomy of privacy and security attacks in online social networks based on the stakeholders of the OSN and the forms of attack targeted at the stakeholders. We identify two primary stakeholders in online social networks: the OSN users and the OSN itself. One one hand, OSN users reveal an astonishing amount of information ranging from personal to professional; the misuse of this information can have significant consequences. In the ecosystem of an OSN, users interact with other users or strangers, use third-party social applications and clicks on ads placed by advertisers. Also, OSNs possess users' social data and sometimes publish sample

[1] Much of the work in this chapter was first published as a technical report in [147]. Permission is included in Appendix A.

datasets after doing an anonymization. Users' information leakage might happen to all of these entities.

On the other hand, OSN services handle users' information and manage all users' activities in the network, being responsible for the correct functioning of its services and maintaining a profitable business model. Indirectly, this translates into ensuring that their users continue to happily use their services without becoming victims of malicious actions. However, attacks such as Sybil, DDoS, spam and malware on OSNs may translate into reputation damage, service disruption, or other consequences with direct effect on the OSN.

We thus classify online social network privacy and security issues into the following attack categories (summarized in Table 12).

1. Attacks on Users: these attacks are are related to information disclosure threats. We identify various such attacks based on the attacker, i.e., an entity who can exploit user information :

   (a) Attacks from other users: Users might put themselves at risk by interacting with other users, specially when some of them are strangers or mere acquaintances. Moreover, some of these users may not even be human (e.g., social robots [134]), or may be crowdsourcing workers interacting with users for mischievous purposes [265]. Therefore, the challenge is to protect users and their information from other users.

   (b) Attacks from social applications: For enhanced functionality, users may interact with various third-party-provided social applications linked to their profiles. To facilitate the interaction between OSN users and these external applications, the OSN provides application developers an interface through which to access user information. Unfortunately, OSNs put users at risk by disclosing more information than necessary to these applications. Malicious applications can collect and use users' private data for undesirable purposes [93].

(c) Attacks from the OSN: Users' interactions with other users and social applications are facilitated by the OSN services, in exchange for, typically, full control over user's information published on the OSN. While this exchange is explicitly stated in Terms of Service documents that the user must agree with (and supposedly read first), in reality few users understand the extent of this exchange [94] and most users do not have a real choice if they don't agree with the exchange. Consequently, the exploitation by the OSN of user's personal information is seen as a breach of trust, and many solutions have been proposed to hide personal information from the very service that stores it.

(d) De-anonymization and inference attacks: OSN services publish social data for others (e.g., researchers, advertisers) to analyze and use for other purposes. Typically, this data is anonymized to protect user information. However, an attacker can de-anonymize social data and infer attributes that the user did not even mention in the OSN (such as sexual or political orientation inferred from the association with other users).

(e) Crawling attacks: Large-scale distributed data crawlers from professional data aggregators exploit the OSN-provided APIs or scrape publicly viewable profile pages to build databases from user profiles and social links. Professional data aggregators sale such databases to insurance companies, background-check agencies, credit-ratings agencies, or others [31]. Crawling users' data from multiple sites and multiple domains increases profiling accuracy. This profiling might lead to "public surveillance", where an overly curious agency (e.g., government) could monitor individuals in public through a variety of media [216].

2. Attacks on the OSN: these attacks are aimed at the service provider itself, by threatening its core business.

(a) Sybil Attacks: Sybil attacks are characterized by users assuming multiple identities to manipulate the outcome of a service [77]. Not specific to OSNs, Sybil attacks were used, for example, to determine the outcome of electronic voting [242], to artificially boost the popularity of some media [236], or to manipulate social search results [143]. However, OSNs have also become vulnerable to Sybil attacks: by controlling many accounts, Sybil users are illegitimately increasing their influence and power in the OSNs [307].

(b) Compromised Accounts: Compromised accounts are legitimate user accounts that are created and used by their fair owners, but have been compromised by attackers [84]. Unlike Sybil accounts, these accounts already have established social connections and normal social network usage history. But suddenly they are hacked by attackers are later used for the ill purposes of the attacker.

(c) Social Spam: Social spam are contents or profiles that an OSN's "legitimate" users don't wish to receive [127]. Spam undermines resource sharing and hampers interactivity among users by contributing phishing attacks, unwanted commercial messages, and promoting websites. Social spam spreads rapidly via OSNs due to the embedded trust relationships among online friends, which motivates a user to read messages or even click on links shared by her friends.

(d) Distributed Denial-of-service attacks (DDoS): DDoSes are common forms of attacks, where a service is sent a large amount of seemingly inoffensive service requests that overload the service and deny access to it [202]. As many popular services, OSNs are also subjected to such coordinated, distributed attacks.

(e) Malware Attacks: Malware is the collective name for programs that gain access, disrupt computer operation, gather sensitive information, or damage a computer without the knowledge of the owner. ONSs are being exploited for propagating malware [89].

Table 12 Categories of attacks.

Attacks on Users	Attacks on the OSN
Attacks from other users	Sybil attacks
Attacks from social applications	Compromised accounts
Attacks from the OSN	Social spam
De-anonymization and inference attacks	Distributed Denial-of-service attacks (DDoS)
Crawling attacks	MalwareAttacks

Like social spam, malware propagation is rapid due to the trust relationships in social networks.

The rest of the chapter is organized as follows. Mitigating attacks on users (Sections 8.2 to 8.6) include discussions of attacks from other users (Section 8.2), from social applications (Section 8.3), from the OSN itself (Section 8.4), de-anonymization and inference attacks (Section 8.5), and crawling attacks (Section 8.6). A summary of the mitigating solutions for the attacks on users is presented in Figure 61. Mitigating attacks on the OSN (Sections 8.7 to 8.11) includes a discussion of Sybil attacks (Section 8.7), attacks from compromised accounts (Section 8.8), social spam (Section 8.9), distributed denial-of-service attacks (Section 8.10) and malware (Section 8.11). A summary of the mitigating solutions for the attacks on OSNs is presented in Figure 62. Section 8.12 highlights some challenges and discusses future research directions. Finally, we conclude the paper in Section 8.13.

8.2 Mitigating Attacks from Other Users

Users reveal an astonishing amount of personally identifiable information on OSNs, including physical, psychological, cultural and preferential attributes. For example, Gross and Acquisti's study [116] shows that 90.8% of Facebook profiles have an image, 87.8% of profiles have posted their birth date, 39.9% have revealed phone number, and 50.8% profiles show their current residence. The study also shows that the majority of users reveal their political views,

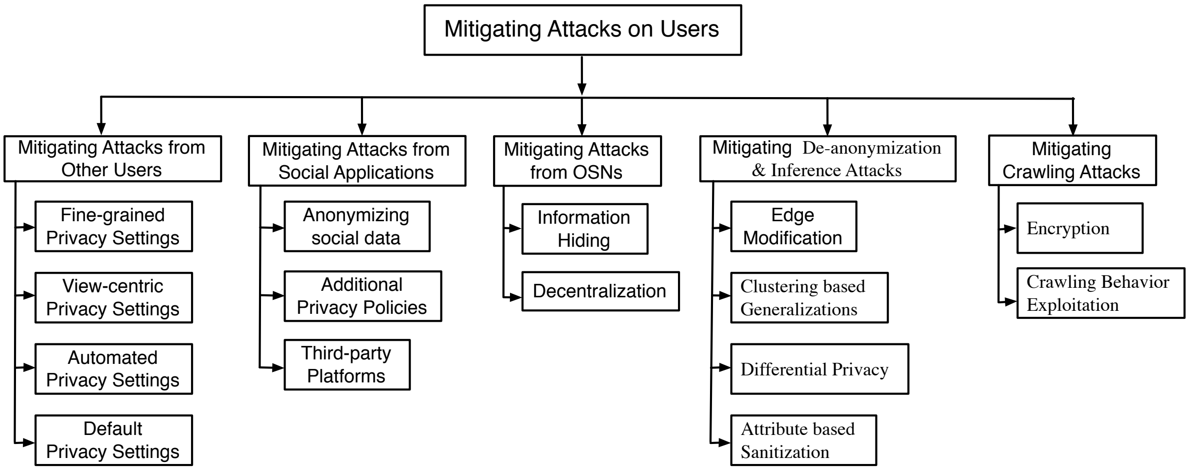

Mitigating Attacks on Users
Mitigating Attacks from Other Users
Fine-grained Privacy Settings
View-centric Privacy Settings
Automated Privacy Settings
Default Privacy Settings
Mitigating Attacks from Social Applications
Anonymizing social data
Additional Privacy Policies
Third-party Platforms
Mitigating Attacks from OSNs
Information Hiding
Decentralization
Mitigating De-anonymization & Inference Attacks
Edge Modification
Clustering based Generalizations
Differential Privacy
Attribute based Sanitization
Mitigating Crawling Attacks
Encryption
Crawling Behavior Exploitation

Figure 61 Mitigating solutions for the attacks on users.

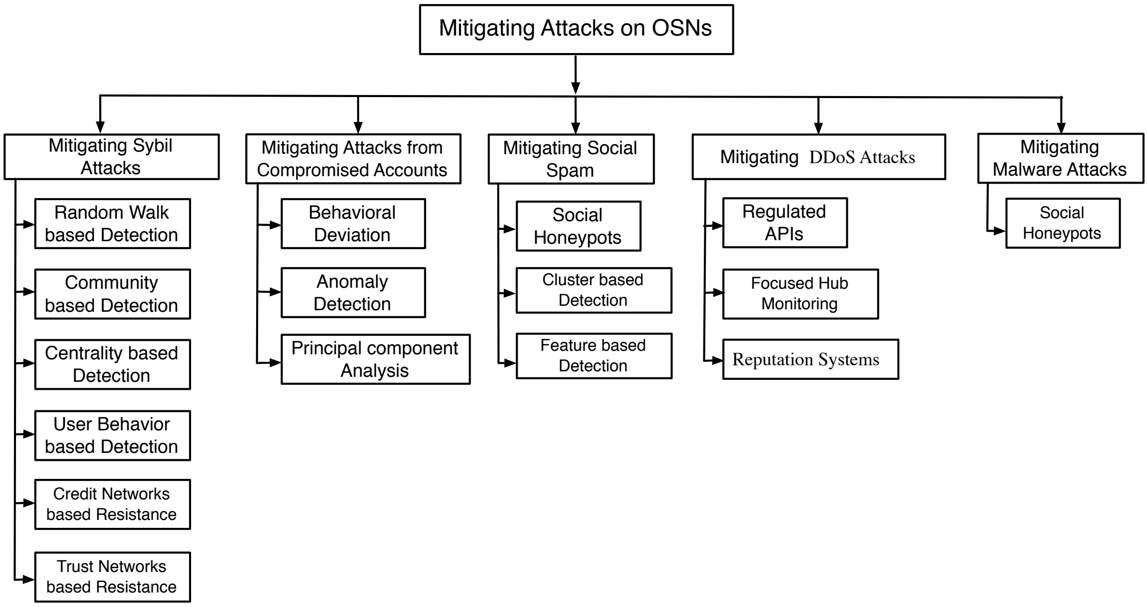

Mitigating Attacks on OSNs
Mitigating Sybil Attacks
Random Walk based Detection
Community based Detection
Centrality based Detection
User Behavior based Detection
Credit Networks based Resistance
Trust Networks based Resistance
Mitigating Attacks from Compromised Accounts
Behavioral Deviation
Anomaly Detection
Principal component Analysis
Mitigating Social Spam
Social Honeypots
Cluster based Detection
Feature based Detection
Mitigating DDoS Attacks
Regulated APIs
Focused Hub Monitoring
Reputation Systems
Mitigating Malware Attacks
Social Honeypots

Figure 62 Mitigating solutions for the attacks on OSNs.

dating preferences, current relationship status, and various interests (including music, books, and movies).

Due to the diversity and specificity of the personal information shared on OSNs, users put themselves at risk for a variety of cyber and physical attacks. Stalking, for example, is a common risk associated with unprotected location information [121]. Demographic re-identification was shown to be doable: 87% of the US population can be uniquely identified by gender, ZIP code and full date of birth [270]. Moreover, the birth date, hometown, and current residence posted on a user's profile are enough to estimate the user's social security number and thus expose the user to identity theft [116]. Unintended revealing of personal information brings other online risks, including scraping and harvesting [179, 266], social pushing [139], and automated social engineering [27].

Given the amount of sensitive information users expose on OSNs and the different types of relationships in their online social circles, the challenge OSNs face is to provide the correct tools for users to protect their own information from others while taking full advantage of the benefits of information sharing. This challenge translates into a need for fine-grained settings, that allow flexibility within a type of relationships (as not all friends are equal [20, 59]) and flexibility with the diversity of personal data. However, this fine granularity in classifying bits of personal information and social relationships leads to an overwhelmingly complex cognitive task for the user. Such cognitive challenges worsen an already detrimental user tendency of ignoring settings all together, and blindly trusting the default privacy configurations that serve the OSN's interests rather than the user's.

Solutions to these three challenges are reviewed in the remainder of this section. Section 8.2.1 surveys solutions that allow fine tunings in setting protection of personal data. The complexity challenge is addressed in the literature on two planes: by providing a visual interface in support of the complex decision that the user has to make (Section 8.2.2) and by automating the privacy settings (Section 8.2.3). To address the problem of users not changing the platform's default settings, researchers proposed various solutions presented in Section 8.2.4.

8.2.1 Fine-grained Privacy Settings

Fine-grained privacy advocates [163, 251] argue that fine-grained privacy controls are crucial features for privacy management. Krishnamurthy et al. [163] introduce privacy "bits"—pieces of user information grouped together for setting privacy controls in OSNs. In particular, they categorize a user's data into multiple pre-defined bits, namely thumbnail (e.g., user name and photo); greater profile (e.g., interests, relationships and others); list of friends; user-generated content (such as photos, videos, comments and links) and comments (e.g., status updates, comments, testimonials and tags about the user or user content). Users can share these bits with a wide range of pre-defined users, including friends, friends of friends, groups, and all. Current OSN services (e.g., Facebook and Google+) have implemented this idea by allowing users to create their own social circles and to define which pieces of information can be accessed by which circle.

To help users navigate the amount of social information necessary for setting correct fine-grained privacy policies, researchers suggest various ways to model the social graph. One model is based on ontologies that exploits the inherent level of trust associated with relationship definition to specify privacy settings. Kruk [165] proposes Friend-of-a-Friend (FOAF)-Realm, an ontology-based access control mechanism that uses RDF to describe relations among users. The system uses a generic definition of relationships ("knows") as a trust metric and generate rules that control a friend's access to resources based on the degree of separation in the social network. Choi et al. [53] propose a more fine-grained approach, which considers named relationships (e.g., "worksWith", "isFriendOf", "knowsOf") in modeling the social network and the access control. A more nuanced trust-related access control model is proposed by Carminati et al. [47]

based on relationship type, degree of separation, and a quantification of trust between users in the network.

For more fine-grained ontology-based privacy settings, semantic rules have been used. Rule-based policies represent the social knowledge base in an ontology and define policies as Semantic Web Rule Language (SWRL) rules. SWRL is a language for the Semantic Web, which can represent rules as well as logic. Researchers used SWRL to express access control rules that are set by the users. Finally, access request related authorization is provided by reasoning on the social knowledge base. Systems that leverage OWL and SWRL to provide rule-based access control framework are [46, 86, 192]. Although conceptually similar, [46] provides richer OWL ontology and different types of policies; access control policy, admin policy and filtering policy. A more detailed semantic rule-based model is developed by Masoumzadeh and Joshi [192]. Rule-based privacy models have two challenges to overcome. First, authorization is provided by forward reasoning on the whole knowledge base, challenging scalability with the size of the knowledge base. Second, rule management is complex and requires a team of expert administrators [88].

Role and Relationship-Based Access Control (ReBAC) are other types of fine-grained privacy models that employ roles and relationships in modeling the social graph. The working principle of these models is two-fold: 1) track roles or relationships between resource (e.g., photos) owner and the resource accessor; 2) enforce access control policies in terms of the roles or relationships. Fong [98] proposes a ReBAC model based on the context-dependent nature of relationships in social networks. This model targets social networks that are poly-relational (e.g., teacher-student relationships are distinct from child-parent relationships), directed (e.g., teacher-student relationships are distinct from student-teacher relationships) and tracks multiple access contexts that are organized into a tree-shaped hierarchy. When access is requested in a context, the relationships from all the ancestor contexts are combined with the relationships in the target access context to construct a network on which authorization decisions are made. Giunchiglia et al. [108] propose RelBac, another relation-based access control model to support sharing of data among large groups of users. The model defines permissions as relations between users and data, thus separating them from roles. The entity-relationship model of RelBac enables description logics and as well as the reasoning for access control policies.

In practice, many online social networks (such as Facebook) have already implemented fine-grained controls. A study of Bonneau et al. [32] on 29 general purpose online social network sites shows that 13 of them offer a line-item setting where individual data items could be set with different visibility. These line-item settings are granular (one data item is one ‘bit’) and flexible (users can change social circles).

8.2.2 View-centric Privacy Settings

Lack of appropriate visual feedback has been identified as one of the reasons for confusing and time consuming privacy settings [263]. View-centric privacy solutions are built on the intuition that a better interface for setting privacy controls can impact users’ understanding of privacy settings and thus their success in correctly exercising privacy controls. These solutions visually inform the user of the setting choices and consequences of his choices.

In [180], the authors propose an alternative interface for Facebook privacy settings. This interface is a collection of tabbed pages, where each page shows a different view of the profile as seen by a particular audience (e.g., friends, friends of friends, etc.), along with controls for restricting the information shared with that group. While this solution provides visual feedback on how other users will see her profile, it’s management is tedious for users with many groups.

A simpler interface is proposed by *C4PS* (Colors for Privacy Settings) [225], which applies color coding for different privacy visibilities to minimize the cognitive overhead of the authorization task. This approach applies four color schemes for different groups of users; red–visible to nobody; blue–visible to selected friends; yellow–visible to all friends; and green–visible to everyone. A user can change the privacy setting for a specific data item by clicking the buttons on the edge of the attribute. The color of the buttons shows the visibility of the data. If users click "selected friend" (blue) button, a window will open in which friends or groups (a pre-defined set of friends) are granted access to the data item. A recent work of Sternel et al. [262] also have used colors for privacy settings and proposed a new interface in the shape of a wheel.

A similar approach is implemented in today's most popular OSNs in different ways. For example, Facebook provides a dropdown of viewers (e.g., only me, friends, and public) with icons as visual feedback. In the custom setting, users can set more granular scales, e.g., share the data item with friends of friends, friends of those tagged and restrict sharing with specific people or lists of people. A qualitative study [283] of teenage OSN users shows that colorful privacy settings enable to have more control over the sharing of their information.

8.2.3 Automated Privacy Settings

Automated privacy settings methods employ machine learning to automatically configure a user's privacy setting with minimal user effort.

Fang and Lefevre's *privacy wizard* [91] iteratively asks a user about his privacy preferences (*allow* or *deny*) for specific (*data item, friend*) pairs. The wizard constructs a classifier from these preferences, which automatically assigns privileges to the remaining of the user's friends. The classifier considers two types of features: community structure (e.g., to which community a friend of the user belongs) and profile information (such as age, gender, relationship status, education, political and religion views, work history). The classifiers employed (Naive-Bayes, NearestNeighbors and Decision Tree) use uncertainty sampling [176], an active learning paradigm, acknowledging the fact that users may quit labeling friends at any time. Bilogrevic et al. [28] also have employed machine learning techniques and proposed *SPISM* for privacy-aware information sharing in mobile social networks. Their system uses personal and contextual features and automatically defines what information to be shared with others and with what granularity.

Social Circles [2] is an automated grouping technique that analyzes the users' social graph to identify "social circles", clusters of densely and closely connected friends. The authors posit social circles as uniform groups from the perspective of privacy settings. The assumption is that users will share the same information with all friends in a social circle. Hence, friends are automatically categorized into social circles for different circle-specific privacy policy settings. To find the social circles, they used a (α, β) clustering algorithm proposed in [204]. While convenient, this approach limits users' flexibility in changing the automate settings.

Danezis [66] aims to infer the *context* within which user interactions happen, and enforces policies to prevent users that are outside that context from seeing the interaction. Conceptually similar to Social Circles, contexts are defined as cohesive groups of users, e.g., groups that have many links within the group and fewer links with non-members of the group. The author used a greedy algorithm to extract the set of groups from a social graph.

An inherent tradeoff for this class of solutions is ease of use vs. flexibility: while the average user might be satisfied with an automatically-generated privacy policy, the more savy user will want more transparency and possibly more control. To this end, the privacy wizard [91] provides for advanced users the visualization of a decision tree model and tools to change it. Another challenge for some of these solutions is bootstrapping: a newcomer in the online social network has no history of interactions to inform such approaches.

8.2.4 Default Privacy Settings

Studies have shown that users on OSNs often do not take advantage of the privacy controls available. For example, more than 99% Twitter users retained the default privacy setting where their name, list of followers, location, website, and biographical information are visible [162]. Similarly, the majority of Facebook users has default settings [1, 116, 163]. Underutilization of privacy options are mostly due to poor privacy setting interface [180], intricate privacy settings [189], and inherent trust in OSNs [1, 35]. The problem with not changing the default settings is that they almost always tend to be more open that the users would prefer [184]. To overcome this situation, approaches to automatically generate more appropriate default privacy settings have been proposed.

PriMa [258] automatically generates privacy policies, acknowledging the fact that the average user will find the task of personalizing his access control policies overwhelming, due to growing complexity of OSNs and the diversity of user content. The policies in PriMa are generated based on the average privacy preference of similar and related users, the accessibility of similar items from similar and related users, closeness of owner and accessor (measured by the number of common friends), the popularity of the owner (i.e., popular users have sensitive profile items), etc. However, a large number of factors and their parametrized tuning contribute to longer policy generation and enforcement time. A related approach, *PolicyMgr* [248], uses supervised learning of user-provided example policy settings and builds classifiers that are then used for automatically generating access control policies.

Aegis [145, 146] is a privacy framework and implementation that leverages the *'Privacy as Contextual Integrity'* theory proposed by Nissenbaum [216] for generating default privacy policies. Unlike the approaches just presented above, this solution does not need user input or access history. Instead, it aggregates social data from different OSNs in an ontology-based data store and then applies the two norms of Nissembaum's theory to regulate the flow of information between social spheres and access to information within a social sphere.

8.3 Mitigating Attacks from Social Applications

Social applications, written by third-party developers and running on OSN platforms, provide enhanced functionality linked to a user profile. For example, *Candy Crush Saga* (a social game) and *Horoscopes* (users can check horoscope) are two popular social applications on Facebook.

The social networking platform works as a proxy between users and applications and mediates the communication between them. To better understand this proxy, we show data flow between a third-party social application and the Facebook platform in Figure 63. An application is hosted on a third-party server and runs on user's data that are taken from the Facebook platform. When a user installs the application on Facebook, it takes permission from the user to use some of her profile information. Application developers write the application pages of an application using Facebook mark-up language (FBML)—a subset of HTML and CSS extended with proprietary Facebook tags.

When a user interacts with an application, such as clicks an application icon on Facebook to generate horoscopes (step 1 on Figure 63), Facebook requests the page from the third-party server where the application is actually hosted (step 2). The application requests the user's profile information using secret communication with Facebook (step 3). The application uses the information (e.g., birth date may be used to create horoscopes) and returns a FBML page to Facebook (step 4). Facebook finally transforms the application page from the server by replacing the FBML page with standard HTML, JavaScript (step 5), and transmits the output page to the end user (step 6).

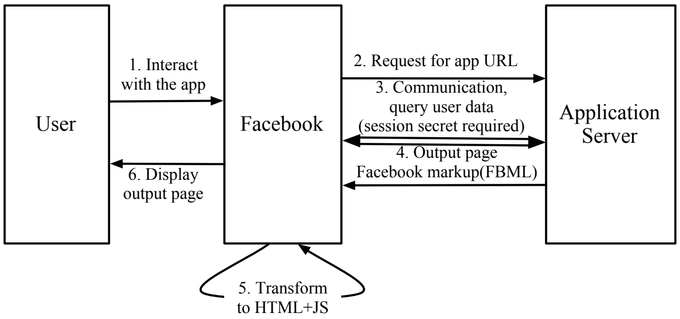


Figure 63 Data flow in a Facebook application.

OSN users are facing multiple risks while using social applications. First, an application might be malicious; it could collect a high volume of user data for unwanted usage. For example, to show this vulnerability, BBC News developed a malicious application that could collect large amounts of user data in only three hours [154].

Second, application developers can violate developer policies to control user data. Application developers are supposed to abide by a set of rules set by the OSNs, called *"developer policies"*. Developer polices are intended to prohibit application developers from misusing personal information or forwarding it to other parties. However, reported incidents [200, 260] show that applications violate these developer policies. For example, a Facebook application, "Top Friends" enabled everyone to view the birthday, gender and relationship status of all Top Friends users, even though those users kept their privacy for those information to private [200], violating the developer policies that private information of friends are not accessible. The Wall Street Journal finds evidence that Facebook applications transmit identifying information to advertising and tracking companies [260].

Third, third-party social applications can query more data about a user from an OSN, regardless whether needed or not for proper operation. A study by Felt and Evans [93] of 150 of the top applications on Facebook shows that most of the applications only needed user name, friends, and their networks. However, 91% of social networking applications have accessed data that they do not need for operation. This violates the principle of least privilege [244], which states that every user should only get the minimal set of access rights that enables him to complete his task.

Finally, a poorly designed API might lead to application impersonation attacks, where an attacker successfully assumes the identity of a legitimate application and possess users' data shared with the application. For example, recently, many OSNs use *OAuth 2.0* protocol to grant access to API endpoints. Hu et al. [132] show that the application impersonation attack is possible due to OAuth's multiple authorization flows and token types. Their investigation on 12 major OSN providers show that 8 of them are vulnerable to application impersonation attacks.

We identified three classes of solutions that attempt to minimize the privacy risks stated above: (i) by anonymizing social data made available to applications (Section 8.3.1); (ii) by defining and enforcing more granular privacy policies that the third-party applications have to respect (Section 8.3.2); and (iii) by providing third-party platforms for executing these applications and limiting the transfer of the social data from applications to other parties (Section 8.3.3).

8.3.1 Anonymizing Social Data for Third-party Applications

Privacy-by-proxy [93] uses special markup tags that abstract user data and handle user input. Third-party applications do not have access to users' personal data, rather they use users' IDs and tags to display data to users. For example, to display a user's hometown, an application would use a tag <hometown id="3125"/>. The social network server would then replace the tag with real data value (e.g., "New York") while rendering the corresponding page to the user. However, applications might rely on private data for operations, for example a horoscope application might require users' gender information. A conditional tag handles this dependency (e.g., <if-male> tag can choose the gender of an avatar). Privacy-by-proxy ensures privacy by limiting what applications can access, which might also limit the social value and usability of the applications. Data availability through proxy also means that application developers have to expose the business logic to social network sites (in a form of Javascript to end users). This might discourage third-party developers in the first place. Moreover, applications could still develop learning mechanisms to infer attributes of a user. For example, developers might include scripting code in the personal data dependent conditional execution blocks (if-else) that could send information to an external server when the block executes.

Similar to Privacy-by-proxy, PESAP [240] provides anonymized social data to applications. However, PESAP secures the information flow inside the browser, so that applications cannot do information leakage though outgoing communications with other third-parties. The anonymization is provided by encrypting the IDs of the entities of the social graph with an application-specific symmetric key. Applications use a REST API to get access to the anonymized social graph. PESAP provides a re-identification end-point in order to enable users to see the personal information of their friends in the context of social applications. Secure information flow techniques protect the private information in the browser of a user. This is done by a dynamic, secure multi-execution flow technique [74], which analyzes information flow inside a browser and ensures that the flow complies with certain policies. The multi-execution flow technique labels the inputs and the outputs of the system with security labels and runs a separate sub-execution of the program for each security label. The inputs have designated security labels and can be accessed by a sub-execution having the same or a higher security label. Figure 64 shows the data flow in a PESAP-aware browser.

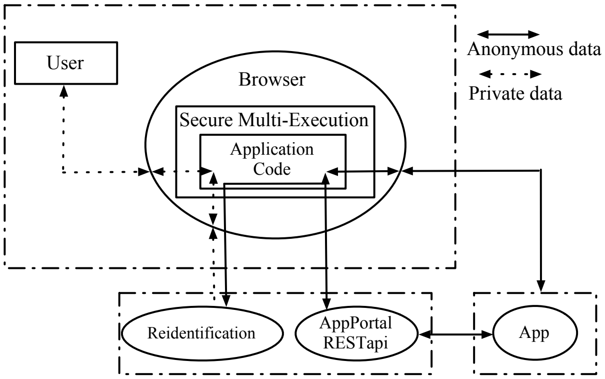


Figure 64 Data flow in a PESAP aware browser [240].

8.3.2 Enforcing Additional Privacy Policies to Social Applications

Besmer et at. [26] propose an access control framework for applications, which adds a new user-application policy layer on the top of the user-user policy to restrict the information applications can access. Upon installing an application, a user can specify which profile information the application could access. However, the framework still uses user-to-user policy to additionally govern an application's access to friends' information on behalf of the user (Alice's installed applications will not get her friend Bob's private data if user-user policy of Bob denies Alice to do so). An additional *friendship-based protection* restricts the information the application can request of a user's friends. For example, Alice installs an application which requests her friend Bob's information and Bob did not install the application. Consider that Bob's default privacy policy is very permissive. But Alice is a privacy conscious and she allows applications to access only the Birth Date attribute. According to friendship-based protection, when the application will request Bob's information via Alice, it will only be able to get Bob's birth date. So, friendship-based protection enables Alice's privacy policies to extend to Bob. The model works well for privacy-savvy concerned users who make informed decisions about an application's data usage while installing an application. An additional functionality could be a set of restrictive default policies for average users.

Similar to the previous work, Cheng et al. [50] also propose an access control framework. However, Besmer et at.'s approach allows applications to transmit users' data to their servers. On the contrary, Cheng et al.'s framework only permits privacy-nonsensitive data to be transmitted, if any functionality of the application runs outside of the OSN. Applications (or functions of an application) that run under the surveillance of the OSN can only get the raw private data.

8.3.3 Third-party Platforms for Running Social Applications

Egele et al. [85] note that, since popular OSN services such as Facebook did not implement user-defined access control mechanisms to date, pragmatic solutions should not rely on the help of OSNs. They introduce PoX, a browser extension for Facebook applications that runs on a client machine and works as a proxy to provide fine-grained access controls. PoX works as a reference monitor which sits between applications and the Facebook server and controls an application's access to users' data stored on the server. In so doing, an application requests the proxy for users' profile data. Upon receiving the request, the proxy performs access control checks based on user-provided privacy settings. If the request is allowed, the proxy signs the access request with its key, sends the request to the OSN server, and finally replays the result from the server to the application. This application to server data flow is shown in Figure 65. An application developer needs to use the PoX server-side library instead of the Facebook server-side library. One potential challenge is to motivate application developers to write PoX-aware applications when existing mechanisms (e.g., Facebook application environment) are perfectly in place.

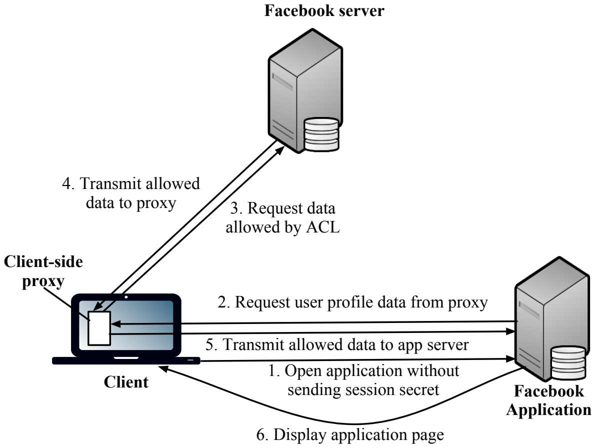

Facebook server
4. Transmit allowed data to proxy
3. Request data allowed by ACL
Client-side proxy
2. Request user profile data from proxy
5. Transmit allowed data to app server
1. Open application without sending session secret
Client
Facebook Application
6. Display application page

Figure 65 A data-flow between applications and server with PoX [85].

xBook [252] is a restricted ADSafe-based JavaScript framework that provides a server-side container in which applications are hosted and a client-side environment to render the applications to users. xBook is different than PoX in that it not only controls third-party applications' access to user data (which PoX also does), but also it limits what applications do with the data. Applications are developed as a set of components; a component is a smallest granular building block of codes monitored by xBook. A component also reveals the information that the component can access and the external entity with which it communicates. During the deployment of an application in xBook, an application developer requires to specify these information. From the specification, xBook generates a manifest for the application. A manifest is a set of statements that specifies what user data the application will use and with which external services it will share the data. At the time of installing the application, the manifest will be presented to the user. In this way, a user will be able to make a more informed decision before installing an application. Although xBook controls third-party applications' access to user data and limits application's data usage, it has to deal with two challenges. First, the platform itself has to be trusted by users and by applications, as it is expected to protect users' personal data and enable third-party applications to execute. Second, hosting and executing applications in xBook requires resources (storage, computation and maintenance) that may be difficult to provide in the absence of a business model. A recent and similar approach of xBook is MUTT [247], however, it has the challenges related to xBook to overcome.

8.4 Protecting User Data from the OSN

The "notice-and-consent" approach to online privacy is the status-quo for practically all online services, OSNs included. This approach informs the user of the privacy practices of the service and provides the user a choice whether to engage in the service or not.

The limitations of this approach have been acknowledged for long. First, the long and abstruse privacy policies offered for reading are virtually impossible to understand, even if the user is willing to invest the time for reading them. For example, on August 2014, we found 4389 words on Facebook's privacy policies and 3473 words on Twitter's privacy policies. Second, such policies always leave room for future modifications; therefore, the user is expected to read them repeatedly in order to practice informed consent. And third, long as they are, these privacy policies tend to be incomplete [55], as they often cannot include all the parties to which user's private information will be allowed to flow (such as advertisers). Consequently, generally people do not read the Terms of Service and when they do, they do not understand them [94].

A second serious deterrent for users protecting their online privacy is the "take-it-or-leave-it" "choice" the users are offered. While it may seem as a free choice, in reality the cost of not using the online service (whether email, browsing, shopping, etc.) is unacceptably high.

Cornered in this space of falsely informed and lack of choice, users may look for solutions that allow them to use the online service without paying the information cost associated with it. Researchers built on this intuition in two directions. The first direction tends to hide the user information from the very service that stores it (Section 8.4.1). The second taps into different business models than the ones that make a living from user's private information and replaces the centralized service provider with a fully decentralized solution that is privacy-aware by design (Section 8.4.2).

8.4.1 Protection by Information Hiding

This line of work is empirically supported by the Acquisti and Gross's study [1] that shows that while 60% of users trust their friends completely with their private and personal information, only 18% of users trust Facebook to the same degree.

The general approach for hiding information from the OSN is based on the observation that OSNs can run on *fake* data. If the operations that OSNs perform on the fake data are mapped back to original data, users can still use the OSNs without providing them real information. Fake data could be ciphertext (encrypted) or obtained by substituting the original data with pre-mapped data from a dictionary. Encrypted data can be stored on a user's trusted device (including third-party servers or a friend's computer). Access controls are provided by allowing authorized users (e.g., friends) to get the original data from the fake data. Different implementations of this idea are presented next.

flyByNight [187] is a Facebook application that enables users to communicate on Facebook without storing a recorded trace of their communication in Facebook. The flyByNight Facebook application generates a public/private key pair and a password during configuration. The password is used as a key to encrypt the private key and the key is stored on flyByNight server. When a user installs the application, it downloads a client-side JavaScript from the FlyByNight server. This JavaScript does key generation and cryptographic operations. The application knows a user's friends and their public keys who have also installed the flyByNight application. To send messages to friends, a user enters the message into the application and selects the recipient friends. The client-side JavaScript encrypts the content of the message with other users' public keys, tags the encrypted message with the Facebook ID numbers of their recipients, and sends them to a flyByNight message database server. The encrypted messages reside on the flyByNight server. When a user reads a message, she provides the password to get the private key (stored in the flyByNight key database). The private key is used to decrypt the message. flyByNight operates under the regulation of Facebook, as it is a Facebook application. It is possible that the computation load on the Facebook servers due to encryption, as well as the suspicious lack of communication among users might attract Facebook's attention and lead to deactivating the application. In the worst case, users lose their ability of hiding their communication, but previous messages remain hidden from the OSN.

Persona [17] hides user data from the OSN by combining attribute-based encryption (ABE) and public key cryptography. The core functionalities of current OSNs such as profiles, walls, notes, etc., are implemented in Persona as applications. Persona uses an application "Storage" to enable users to store personal information, and share them with others through an API. Persona application in Facebook is similar to any third-party Facebook application, where users log-in by authenticating to the browser extension. The browser extension translates Persona's special markup language. User information is stored in Persona storage services rather than on Facebook and other Persona users can access the data given that they have the necessary keys and access rights. Similar to the flyByNight, Persona's operation depends on the OSN, as core functionalities are implemented as applications.

NOYB [119] distorts user information in an attempt to hide real identities from the OSN, allowing only trusted users (e.g., friends) to get access to the restored, correct information. To implement this idea, NOYB splits a user's information into atoms. For example, Alice's name, gender and age (Alice, F, 26) are split into two atoms: (Alice, F) and (26). Instead of encrypting the information, NOYB replaces a user's atom with pseudorandomly picked another user's atom. So, Alice's first atom is substituted with, for example, the atom (Athena, F) from Athena's profile, and the second atom with Bob's atom from the same class (38). All atoms from the same class for all users are stored in a dictionary. NOYB uses ciphered index of a user's atom to substitute an atom from this dictionary. Only an authorized friend knows the encryption key and can reverse the encryption. A proof-of-concept implementation of NOYB as a Firefox browser plugin adds a button to ego's Facebook profile that encrypts his information and another button on alter's page that decrypts alter's profile. The cleverness of NOYB is that it stores legitimate atoms of information in plain text, thus not raising the suspicions of the OSN. The challenge, however, is the scalability of the dictionaries: the dictionaries are public, contain atoms from both NOYB users and non-users, and are maintained by a third party with unspecified business/incentive model.

FaceCloak [188], implemented as a Firefox browser extension, protects user information by storing fake data in the OSN. Unlike NOYB, it does not replace a user's information with another user's information, rather it uses dictionaries and random Wikipedia articles as replacements. A user, say Alice, can protect information from the OSN by using a special marker predefined by FaceCloak (@@ in their implementation). When Alice submits the form to the OSN, FaceCloak intercepts the submitted information, replaces the fields that start with the special marker by appropriate fake text and stores the fake data in the OSN. It uses a dictionary (for profile information) and random Wikipedia articles (for walls and notes) to provide fake data. Now, using Alice's master key and personal index key, FaceCloak does the encryption of the real data, computes MAC keys, computes the index, and sends them to a third-party server. Now consider one of Alice's friends Bob, who has installed FaceCloak in his browser, and Bob wants to see Alice's information. After downloading Alice's page (which also includes fake data from the OSN), FaceCloak computes indexes of relevant fields using master and personal index key of Alice. Then it downloads the corresponding values from the third-party server. Upon receiving the value, FaceCloak checks the integrity of the received cipher-text, decrypts it, and substitutes the real data for the fake data. If the value is not found, then the data is left unchanged. Face-

Cloak depends on a “parallel” centralized infrastructure to store the encrypted data, which means that a third-party has to maintain all users’ data, probably without getting any benefits from it. And, users have to trust the reliability of the third-party server, which also represents a single point of failure.

Virtual Private Social Networks) (VPSN) [57], unlike flyByNight, FaceCloak, and NOYB, does not require third-party services to protect users’ information from an OSN. Instead, they leverage the computational and storage resources of the OSN users to store real profile data of other users, while storing fake profile data on Facebook. *FaceVPSN* is a Firefox browser extension that implements VPSN for Facebook. In FaceVPSN, user Alice changes her profile information to some fake information and stores the fake information in Facebook and sends by email her correct and fake profiles in a prespecified XML format to her friends. In order to access Alice’s real profile, her friends have to have FaceVPSN installed (as a regular Firefox extension) and use its GUI to add Alice’s XML file. When Alice’s friend Bob requests Alice’s Facebook page, Facebook sends an HTML response that has Alice’s fake data from Facebook. FaceVPSN’s JavaScript code is triggered when “page load” event is fired. The JavaScript code of FaceVPSN searches the profile information of Alice in Bob’s stored XML file and replaces the fake information with real information.

Unlike other solutions presented above, FaceVPSN does not risk being suspended by the OSN (since it is not an application running with the OSN’s support). Like FaceCloak, however, FaceVPSN requires a user’s friends to install the FaceVPSN extension in order to see the user’s profile. Moreover, FaceVPSN demands a high degree of user interaction that might affect usability. In particular, upon the addition of a new contact to the friend list, the user has to explicitly exchange profile information with the new friend and upload it into the FaceVPSN application. On top of it, every change of profile information has to be emailed as an XML file to all friends, and the friends are required to go through the XML update process in order to see the changes. This entire process affects usability, given the high number of friends a user might have in OSNs (e.g., half the Facebook users have more than 200 friends, and 15% have more than 500 friends [253])

While the various implementations of the idea of hiding the personal information from the OSN have different tradeoffs, as discussed above, there are also risks associated with the approach itself. First, because the OSN operates on fake data (whether encrypted or randomized), it will not be able to provide some personalized services such as social search and recommendation. Second, users end up transferring their trust from the OSN to either a third-party server or friends' computers for unclear benefits. The third-party server provides yet another service whose terms of use are probably presented in yet another incomprehensible Terms of Service document, with an opt-out "choice". Friends' computers require extra care for fault tolerance and malicious attacks. In fact, a recent user study [18] finds that higher usability costs, lack of trust, and poor performance are the main causes of poor or no adoption of these services.

8.4.2 Protection Via Decentralization

An alternative to obfuscate information from the OSN is to migrate to another service that is especially designed for user privacy protection. Research in this area explored the design space of decentralized (peer-to-peer) architectures for managing user information, thus avoiding the centralized service with a global view of the entire user population. The typical overlay used in most of these solutions is based on distributed hash tables, preferred over unstructured overlays for their performance guarantees. In addition, data is encrypted and only authorized users get access to the plain text. In this section, we discuss decentralized solutions for OSNs. There are three dimensions that differentiate the solutions: (1) how the distributed hash table has been implemented (e.g., OpenDHT, FreePastry, Likir DHT)? (2) where to store users' content (e.g., nodes run by the user, by the friends or cloud infrastructures)? (3) how to manage encryption keys for access controls (e.g., public-key infrastructure, out-of-band)?

PeerSooN's [42] architecture has two-tiers. One tier, implemented using OpenDHT, serves as a look-up service to find a user. It stores users' meta-data for example, the IP address, information about files, and notifications for users. A peer can connect to another peer asking the look-up service directly to get all required information. The second tier is formed by peers and it contains users' data, such as user profiles. Users can exchange information either through the DHT (e.g., a message is stored within the DHT if receiver of a message is offline) or directly between their devices. The system assumes a public-key infrastructure (PKI) for privacy protection. A user encrypts data with the public keys of the intended audience, i.e., the friends of the user.

Safebook [60, 61] is a decentralized OSN, which uses a peer-to-peer architecture to get rid of a central, omniscient authority. Safebook has three main components: a trusted identification service for certification of public keys and the assignment of pseudonyms; matryoshkas, a set of concentric shells around each user, which serve to replicate the profile data and anonymizes traffic; and a peer to peer substrate (e.g., DHT) for the location of matryoshkas that enables access to profile data and exchange messages.

LifeSocial.KOM [114] is another P2P-based OSN. It implements common functionalities in OSNs using OSGi-based software components called "plugins". As a P2P overlay, it uses FreePastry for interconnecting the participating nodes and PAST for reliable, replicated data storage. The system uses cryptographic public keys as user ID. To protect privacy, a user encrypts a private data object (e.g., profile information) with a symmetric cryptographic key. She then encrypts the symmetric cryptographic key individually with the public keys of authorized users (e.g., her friends) and appends to the data object. The object and the list of encrypted symmetric keys are also signed by the user and they are stored in the P2P overlay. Other users in the system can authenticate the data object by using the public key of the author. But only authorized users (e.g., friends) can decrypt the symmetric key and thus, the content of the object.

LotusNet [6] is a framework for the implementation of a P2P based OSN on a Likir DHT [5]. It binds a user identity to both overlay nodes and published resources for robustness of the overlay network and secures identity based resource retrieval. Users' information is encrypted and stored in the Likir DHT. Access control responsibility is assigned to overlay index-nodes. Users issue signed grants to other users for accessing their data. DHT returns the stored data to the requestor only if the requestor can provide a proper grant, signed by the data owner.

Vis-a-Vis [246] targets high content availability. Users store their personal data in Virtual Individual Servers (VISes), which are kept on the user's computer. The server data are also replicated on a cloud infrastructure so that the data is available from the cloud when a user's computer is offline. Users can share information with other users using peer-to-peer overlay networks that connect VISes of the users. The cloud service needs to be rented (considering the high volume of the data users store in OSNs), which makes the scheme monetary dependent.

Prometheus [159] is a peer-to-peer social data management system for socially-aware applications. It does not implement traditional OSN functionalities (e.g., profile creation, management, contacts, messaging, etc.), rather it manages users' social information from various sources and exposes APIs for social applications. Users' social data are encrypted and stored in a group of trusted peers selected by users for high service availability. Prometheus architecture is based on Pastry, a DHT-based overlay, and it uses Past to replicate social data. An inference on social data is subject to user defined access control policy enforced by the trusted peers. Prometheus relies on a public-key infrastructure (PKI) for user authentication and message confidentiality.

The toughest challenge for decentralized OSNs is to convince traditional OSN users to migrate to their systems. Centralized social networks have large, established user bases and they are accessible from anywhere. Moreover, they already have a mature infrastructure, making good revenues from users' data and maintaining excellent usability. However, decentralized OSNs are still an alternative for centralized OSNs, specially for privacy-concerned users. For example, Diaspora (https://joindiaspora.com/) is a fully operating open source, stable and decentralized OSN, which relies on user contributed local servers to provide all the major centralized OSN functionalities.

8.5 Mitigating De-anonymization and Inference Attacks

Analysis of social data has become immensely popular in a variety of domains. Researchers and agencies collect or purchase social data to do the analysis. For example, Kwak et al. [167] collected the entire Twitter network as of 2010: 41.7 million Twitter profiles, 1.47 billion follower-following relations, and 106 million tweets. In addition, some organizations and OSN service providers publish social data for others to analyze. For example, the Federal Energy Regulatory Commission published a repository of approximately $500,000$ email messages of Enron Corporation.

However, publishing and allowing the collection of social network data involves privacy disclosure risks. For example, in 2006 AOL released an anonymized dataset of twenty million search keywords for over 650,000 users [12]. The dataset was published for research purpose and novel findings emerged (e.g., [220]) However, despite the fact that the data released was anonymized, users' privacy was compromised. To make the point, the New York Times identified an individual from this dataset by cross referencing users with phonebook listings.

Privacy attacks in published or collected social network data can be categorized into two categories: de-anonymization attacks and inference attacks. In de-anonymization attacks, an attacker uses external background knowledge and published social data to de-anonymize/identify users in the social graph, and thus learn sensitive user information. We find four types of privacy breaches due to de-anonymization attacks in the literature: (1) *Identity disclosure* reveals the identity of a user and makes him vulnerable in the real world; (2) *Social attributes disclosure* refers to the disclosure of sensitive data associated with a user; (3) *Relationship disclosure* refers to the situation when the relationships of a user are exposed and this information exploited; (4) *Social graph property disclosure* refers to the disclosure of various graph metrics, such as degree, betweenness centrality, closeness centrality, or clustering co-efficient.

Typically, two types of methods are available for privacy preserving social network data anonymization: (1) edge modification-based approaches and (2) clustering-based generalization. Edge modification-based approaches preserve privacy by modifying the social graph structure via addition, deletion or randomization of the edges. Clustering-based generalizations cluster nodes and edges into groups and anonymize a subgraph into a super-node [315]. So, details about users are hidden. More in-depth discussion on de-anonymization attacks can be found in [313]. In the following, we discuss the types of de-anonymization attacks, how these attacks take place, and what solutions were proposed to combat such attacks. In this work, we include the latest work on de-anonymization attacks.

8.5.1 De-anonymization Attacks

In de-anonymization attacks, an attacker uses external background knowledge and published social data to de-anonymize/identify users in the social graph, and thus learn sensitive user information.

We categorize privacy breaches due to de-anonymization attacks into four classes:

1. *Identity disclosure* reveals the identity of a user and makes him vulnerable in the real world. For example, although a published dataset on the disease-infection network could advance research on how the disease transmits in communities, an adversary (e.g., an insurance company) that can identify an individual and his disease could exploit this information in unintended ways (for example, for denying insurance).

2. *Social attributes disclosure* refers to the disclosure of sensitive data associated with a user. For example, disclosure of a user's date of birth, gender and home address could allow the inference of the user's social security number (SSN) and hence could lead to identity theft [116].

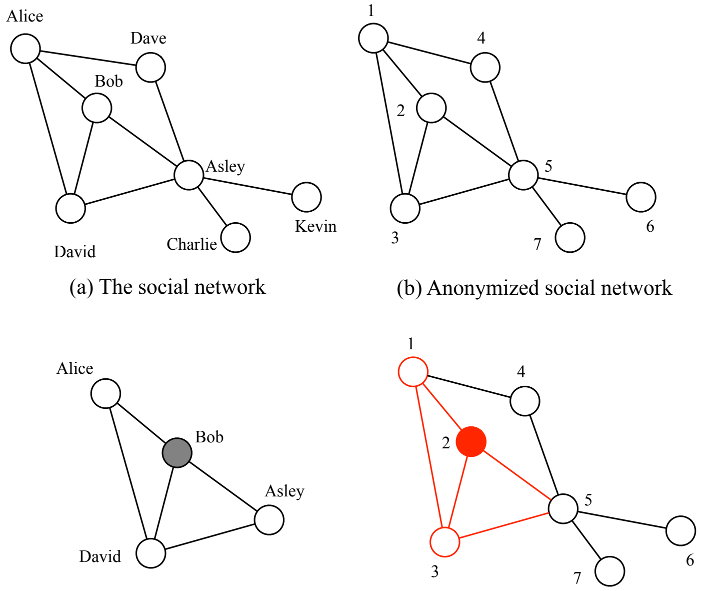
3. *Relationship disclosure* refers to the situation when the relationships of a user are exposed and this information exploited. For example, two nodes (e.g., companies) in a transaction network are connected by an edge and a weight (e.g., transaction expense) if they are involved in a financial transaction. An adversary, for example a competitor company, can detect whether two target companies have done a financial transaction if it can infer whether an edge exists between the two companies in the network. The adversary can learn the transaction expense from the edge weight and can exploit that information to get advantages.

4. *Social graph property disclosure* refers to the disclosure of various graph metrics, such as degree, betweenness centrality, closeness centrality, or clustering co-efficient. An attacker can find out the most central users in the network and can make the network structurally vulnerable. For example, an attacker can identify and remove the highest betweenness centrality nodes to disrupt communications between other nodes in the network [213].

8.5.1.1 De-anonymization Attack Techniques

Anonymization is usually done by substituting personally identifying information associated with each user with a random ID [296]. However, this substitution is not sufficient to preserve users' privacy. For example, consider a social network in Figure 66(a) that has been anonymized in Figure 66(b) by replacing user names with random IDs. Now, if an attacker knows that Alice, David and Asley are friends of Bob and Alice-David and Asley-David are also friends, a subgraph shown in Figure 66(c), then the attacker can uniquely identify the subgraph in the anonymized network (shown in Figure 66(d)). So, the attacker will be able to re-identify Bob in the anonymized and published social network.

Researchers have shown different techniques to perform de-anonymization attacks. Backstorm et al. [16] present two types of attacks—*active* and *passive*. In active attacks, the attacker is assumed to be able to modify the network prior to social network data release. The attacker



(a) The social network

(b) Anonymized social network

(c) The subgraph of Bob

(b) Bob is re-identified

Figure 66 Anonymization and de-anonmymization attacks.

chooses an arbitrary set of target individuals (whose privacy she wants to compromise), creates a small number of new user accounts, makes connections with target individuals (thus forms edges), and establishes a highly distinguishable pattern comprising nodes and edges among the new accounts. The attacker can then efficiently find the subgraph in the released anonymized network, thus can expose the identities of the target individuals.

In passive attacks, an attacker does not have to create new accounts or connections. The intuition is that most nodes in social networks form small uniquely identifiable subgraphs. So, the attacker simply has to form a coalition with other users. The attacker recruits $k - 1$ number of his neighbors and forms a coalition of size k. The users in the coalition know names of their neighbors outside of the coalition. Finally, the attacker tries to identify the subgraph (formed by the coalition) in the published social network, and compromises the privacy of neighboring nodes.

Narayanan et al. [211] demonstrate the feasibility of a large-scale de-anonymization attack under the assumption that the attacker has background knowledge of a different network whose membership partially overlaps with the target network. Using a de-anonymization algorithm, the authors show that a third of the common users of both Twitter and Flickr can be identified in the anonymous Twitter social graph with a low (12%) error rate.

Both attacks [16, 211] presented above use a subgraph as background knowledge. However, the way an attacker achieves this knowledge is different. In [16], an attacker creates the background knowledge by adding nodes and edges in the social graphs or forming a coalition among nodes. In [211], the authors propose to collect network data by crawling OSNs, or deploying third-party malicious applications.

8.5.1.2 Privacy Preserving Anonymization Methods

In order to combat de-anonymization attacks, a social network should be anonymized properly before publishing. We categorize privacy preserving social network data anonymization methods into two categories: (1) edge modification-based approaches and (2) clustering-based generalization.

Edge modification-based approaches preserve privacy by modifying the social graph structure via addition, deletion or randomization of the edges. Zhour and Pei [314] consider a de-anonymization attack, where an attacker, equipped with the background knowledge about the target's 1-hop neighbors, attempts to re-identify the target in the anonymized dataset using neighborhood matching. Their anonymization method is inspired by k-anonymity model. Although not targeted to social network data publishing, k-anonymity model ensures that each user's information in a released dataset cannot be identified from at least $k-1$ other individuals in the dataset [271] .

Zhour and Pei extend k-anonymization to social networks. The goal of the anonymization is to ensure that even knowing the neighborhood of a node, an attacker will not be able to re-identify the node in the anonymized dataset with confidence higher than $\frac{1}{k}$. Let we have a social network $G = (V, E)$ and the anonymized network $\hat{G} = (\hat{V}, \hat{E})$, where there exists a bijection function $f : V \rightarrow \hat{V}$ and for each $(u, v) \in E$, $(f(u), f(v)) \in \hat{E}$. The authors assume that an attacker has the knowledge of the neighborhood subgraph of a node $u \in V(G)$, denoted by $Neighbor_G(u)$. The goal of the k-anonymization is to ensure that there exist at least $k - 1$ other nodes $v_1, v_2, v_3, \ldots, v_{k-1} \in V(G)$ such that $Neighbor_{\hat{G}}(f(v_1)), \ldots, Neighbor_{\hat{G}}(f(v_{k-1}))$ are isomorphic. The anonymization method first extracts the neighborhoods of all nodes in the network. Then it greedily combines nodes into groups and anonymizes the neighborhoods of the nodes in the group, until k-anonymity conditions are met. To anonymize neighborhoods of two nodes such as $Neighbor_G(u)$ and $Neighbor_G(v)$, the method first finds all perfect matches of neighborhood components in $Neighbor_G(u)$ and $Neighbor_G(v)$. For those unmatched components, it tries to pair similar components based on anonymization cost and anonymizes them (this might involve an addition of an edge between two nodes).

Liu and Terzi [181] propose k-degree anonymity to combat de-anonymization attacks. They assume that an attacker has the background knowledge of the degree of a target node. The attacker could search the degrees of the nodes in the published network and could re-identify a target node. k-degree anonymity ensures that for every node u in the graph, there exist at least $(k - 1)$ other nodes having the same degree as u. Similar to the work of Zhour and Pei, even having the degree background knowledge, an attacker will not be able to re-identify the node in the anonymized dataset with confidence higher than $\frac{1}{k}$. Their anonymization algorithm has two steps. In the first step, it starts from a degree sequence $\mathbf{d}$ of the original network $G(V, E)$ and constructs a new degree sequence $\hat{\mathbf{d}}$ that is k-degree anonymous, so that degree anonymization cost is minimized. In the second step, the algorithm constructs a graph $\hat{G} = (\hat{V}, \hat{E})$ such that $\mathbf{d}_{\hat{\mathbf{G}}} = \hat{\mathbf{d}}$, $\hat{V} = V$ and $\hat{E} = E$. To solve the first step, the authors used a dynamic programming method, while the second step is based on a set of graph construction algorithms given a degree sequence with constraints. To construct the new degree sequence the algorithm uses a randomized edge swap transformation strategy.

Clustering-based generalizations cluster nodes and edges into groups and anonymize a subgraph into a super-node [315]. So, details about users are hidden.

Hay et al. [123] propose a vertex clustering-based generalization approach to combat de-anonymization attacks. They model an attacker's background knowledge as the access to an entity that answers a restricted knowledge query about a target node in the network. They assume three types of queries: *vertex refinement queries* return the local structure of a node in an iterative refined way (e.g., degree of a node, the set of neighbors degrees of a node); *subgraph queries* confirm a subgraph around a target node; *hub fingerprint queries* for a target node returns the vectors of distances between the node and a set of hubs (note that in social networks a hub is defined as a node with high betweenness and degree centrality [213]). The anonymity method is based on structural similarity. The intuition is that structurally similar nodes may be indistinguishable to an attacker. The anonymity method generalizes a social graph by grouping nodes into partitions and publishes the number of nodes in each partition including the densities of edges across and within the partitions. The size of the partition is at least k (a positive integer), which is similar to k-anonymity in relational data. The method used a simulated annealing algorithm for partitioning [243].

Zheleva and Getoor [311] consider a *link re-identification* attack, where nodes have multiple types of edges and an attacker attempts to re-identify sensitive edges. As a background knowledge, they assume that an attacker can predict a sensitive edge based on other non-sensitive edges. They describe five anonymization techniques: i) remove all sensitive edges; (ii) remove some non-sensitive edges which significantly contribute to the prediction of a sensitive edge; (iii) collapse the anonymized nodes into a single node for each equivalence class (they assume that nodes are clustered into equivalence classes) and publish the count of same types of edges between two equivalence class nodes; (iv) similar technique as (iii), but it needs the equivalence class nodes to have the same constraints as any two nodes in the social network; (v) remove all edges.

Campan and Truta [43] propose an *edge generalization* technique, leveraging the k-anonymity model. In their model, each node is similar to at least other $(k-1)$ nodes considering attributes and associated structured information (e.g., neighborhood structure of nodes). Nodes are partitioned into clusters and nodes from a cluster are combined into one single node. Edges between two clusters are collapsed into a single edge. An edge between two clusters are labeled with the number of edges between them. While this approach is similar to [311], two major differences are: (i) [43] considers all relationships are the same type, but in [311] there are different types of relations; (ii) [43] considers both generalization and structural information loss while clustering.

The main challenge for anonymization methods is providing sufficient anonymity while preserving (all) the relevant structural properties of the network. Without preserving enough of the structural properties of the original network, publishing anonymized social network datasets loses its value. Moreover, Aggarwal et al. [3] show that social graphs reveal robust statistical information about linkage behavior, which is hard to hide using structural-based anonymization, such as the approaches we have already discussed. *Differential privacy* [83] has seen much research attention recently as an alternate of aforementioned structural-based anonymization techniques. Differential privacy makes no assumption about an attacker's background knowledge. It adds controlled levels of "noise" to the original graph and generates statistically similar, but anonymized graphs. Qian et al. [297] employ a statistical hierarchical random graph (HRG) model to infer the social graph structure and achieve the differential privacy by sampling possible HRG structures via Markov chain Monte Carlo (MCMC). Lu and Miklau [186] estimate parameters for the exponential random graph model (EGRM) under differential privacy. ERGMs are a powerful statistical modeling tool for social networks.

8.5.2 Mitigating Inference Attacks

The goal of an inference attack is to infer undisclosed private information about a user using other published details of that user. For example, a person might not want to state her political affiliation in Facebook because of privacy concerns. But if he is a member of "ban the same sex marriage" group, then from this group membership an inference may be possible regarding his political affiliation.

Zheleva and Getoor [312] study four social networks (Facebook, Flickr, Dogster and BibSonomy) and show how an attacker can exploit public and private user profiles to learn private attributes such as user location and gender. They show that declared social relationships and inferred group memberships are enough to predict undisclosed private information. Using the classification model *LINK-GROUP*, a combination of link and group-based classification models, they were able to accurately discover the information of private-profile users.

Heatherly et al. [125] describe three sanitization techniques to prevent undisclosed private information inference from a released social network dataset. First, they build classification models to accurately predict private data from the available details (attributes) of a user. Then they apply the sanitization techniques to reduce the accuracy of the models. In brief, the techniques are as follows: (i) remove some details (e.g., attributes) to decrease the classification accuracy of sensitive attributes; (ii) alter the link structure of the social graph by adding and removing links and (iii) provide a generalization of details. For example, if a user inputs a favorite activity as "Boston Celtics", the name will be replaced by a more generalized term "Basketball". Experimenting on a Facebook dataset, the authors conclude that removal of attributes and friendship links together in the published data is the best way to reduce classifier accuracy.

Dey et al. [76] attempt to infer the age of over one million Facebook users in New York city. Exploiting the Facebook social graph, they design an iterative algorithm which estimates a user's age based on her friends' ages (e.g., from inferred high school graduation year), friends of friends' edges and so on. They find that for most users, including users who take maximal measures to prevent privacy leakage by hiding their friend lists, it is possible to estimate ages with an error of only a few years. The authors recommend to hide high school graduation year and friend lists to other users who are not friends from users' profiles as a solution.

Minkus et al. [201] show how an attacker could infer, in an automated fashion, attributes about babies and young children —who do not even have their own OSN accounts— due to their parents' OSN behavior. First, they use age detection algorithms to the parents' public Facebook photos to automatically identify photos of their children. Then they are able to learn personally identifiable information, such as, name, birth date, and address of the child using automated textual analysis of their parents posts and linking them with publicly available data (voter registration list). To mitigate the risks, their suggestion includes enforcement of more privacy preserving mechanisms by showing messages to parents, or automatically restricting photos containing children to a more private setting if a child's face is detected in a photo.

8.6 Mitigating Attacks from Large-scale Crawlers

OSNs enhance social browsing experience by allowing users to view public profiles of others. This way a user meets others, gets a chance to know strangers and eventually befriends some of them. Unfortunately, attackers are there in the vast landscape of OSNs, who exploit this functionality. Users' social data are always invaluable to marketers. Professional data aggregators build databases using public views of profiles and social links and sale the databases to insurance companies, background-check agencies and credit-ratings agencies [31]. For example, crawling 100 million public profiles from Facebook created news recently [39]. Sometimes crawling is a violation of terms of service. Facebook states that someone should not collect "...users' content or information, or otherwise access Facebook, using automated means (such as harvesting bots, robots, spiders, or scrapers) without our prior permission" [90].

One solution of the problem could be the removal of the public profile view functionality. But removal of the public profile view functionality is against the business model of OSNs. Services like search and targeted advertisements bring new users and ultimately revenues to OSNs, but openly accessible contents are necessary for their operation. Moreover, removal of the public view functionality will undermine user experience, as it makes a connection, communication and sharing easy with unknown people in the network.

OSN operators such as Facebook and Twitter attempt to defend large-scale crawling by limiting the number of user profiles a user can see from an IP address in a time window [261]. However, tracking users with low level network identifiers (e.g., IP address, TCP port numbers or SSL session IDs) is fundamentally flawed as a solution of this problem [295]. Aggressive attackers may gather a large vector of those identifiers by creating a large number of fake user accounts, gaining access to compromised accounts, virtualizing in a cloud, employing botnets, and forwarding requests to proxies. Until now, researchers have leveraged encryption based technique [295] and crawler's observational behavior [207] to combat the problem.

ONS's anti-crawling techniques suffer from the fact that web clients can access a particular page using a common URL accessible to all clients [295]. This can be exploited by a distributed crawler e.g., a crawling thread can download and parse a page for links using a session key and can deliver those links to another crawling thread to download and parse using different session keys. So, if some crawlers get banned from the OSNs for malicious activities, the links they have parsed are still valid and a fresh start is possible from those links. SpikeStrip [295] overcomes the problem by creating unique, per-session "views" of the protected website that forcibly tie each client to their session keys. SpikeStrip is a web server add-on that leverages link encryption technique. It allows OSN administrators to moderate data access, and it defends against large-scale crawling by securely identifying and rate limiting individual sessions.

When a crawler visits a page, it receives a new session key and a copy of the page whose links are all encrypted. SpikeStrip appends each user's session key to those links and then encrypts the result using a server-side, secret symmetric key. It also appends a *salt* to the link after encryption to make each link unique. As time passes, the crawler progressively covers more pages and collects links. However, at a point, the crawler requires to change the session key due to the expiration of the session or due to a ban from the OSN. As SpikeStrip couples all URLs to the browser's session key, this switching of sessions invalidates all the links collected for future traversals. Thus, a fresh start to reconstruct the collection should be started from the beginning. The authors implemented mod_spikestrip, a SpikeStrip implementation for Apache 2.x and showed that it imposes only 7% performance penalty on Apache.

PUBCRAWL [138] is based on the observation that the traffic that a crawler generates is significantly different from fair users. It uses content-based and timing-based footprints to distinguish crawler traffic from regular traffic. Content-based features are extracted from URLs (e.g., access errors, page revisits) and HTTP headers (e.g., cookies, referrers). Timing-based features are obtained from the analysis of the time series produced by the stream of requests. Finally, PUBCRAWL relies on machine learning techniques and trains classifiers using the features that can separate crawler traffic from user traffic.

Genie [207] exploits browsing patterns of honest/real users and crawlers and thwarts large-scale crawls using Credit Networks [64, 107]. While PUBCRAWL uses physical network layer differences, Genie uses social network layer differences. Genie's design is based on three observations from real-world datasets: (i) there is a balance between the number of profiles a honest user views and views requested by other users to her profile, but crawlers view many more profiles than the number of times their profiles are viewed; (ii) a honest user views profiles of socially close users.(iii) a honest user repeatedly views a small set of profiles in the network, but unless re-crawling, the crawlers avoid repeating viewing of other users' profiles. Genie leverages these observations and enforces a viewer to make a "credit payment" in the credit network if a user wants to view a profile. It allows a user (also might be a crawler) to view a profile if a max-flow between them has at least a threshold value. The required credit payment to view a profile depends on the shortest path length from viewer to viewee; a user has to pay more to view the profile of a distant user in the social graph. As a legitimate user usually views one or two hop distant profiles, and also other users also view her profile, her liquidity of credits remains almost the same. On the other hand, a crawler views a lot of distant profiles and gets fewer views. Eventually it lacks credit liquidity to view the profiles of others. As such, the credit network poses a strict rate limit on profile views of the crawlers.

Genie might see a large number of honest users' activities (profile viewing) flagged due to the existence of outliers in a social network. This might limit the usability of social networks, because without viewing a profile an outlier will not be able to befriend others. Genie also might require a fast computation of shortest paths, as for each profile viewing request, it computes all the shortest paths from viewer to viewee. Intuitively, this operation is too costly in a modern social network (more than one billion users), even considering the state of the art shortest path algorithms.

Both SpikeStrip and Genie limit crawlers' ability to quickly aggregate a significant portion of OSNs user data. Unfortunately, equipped with a large number of user profiles (fake or compromised) and employing dedicated crawlers for a long time, attackers could still collect a huge amount of users' social data.

8.7 Mitigating Sybil Attacks

The Sybil attack is a fundamental problem in distributed systems. The term *Sybil* was first introduced by Douceur [77], inspired from a 1973 book after the same name about the treatment of a person Sybil Dorsett, who manifests sixteen personalities. In Sybil attacks, an attacker creates multiple identities and influence the working of the system. OSNs including Digg, YouTube, Facebook and BitTorrent have become vulnerable to Sybil attacks. For example, Facebook anticipates that up to 83 million of its users may be illegitimate [23], which is far more than what it anticipates (54 million) earlier [48]. Researchers found that Sybil users affect the correct functioning of the system by contributing malicious contents [115, 143] and illegitimately increasing influence and power [212, 236].

Malicious activities from Sybil users are posing serious threats to OSN users, who trust the service and depend on it for online interactions. Sybils cost OSN providers, too, in terms of monetary losses and time. OSN providers spend significant resources and times to detect, verify, and shut down Sybil identities. For example, Tuenti, the largest OSN in Spain, dedicates 14 full-time employees to manually verify user reported Sybil identities [44].

Two categories of solutions are available to defend Sybils: Sybil detection and Sybil resistance. Sybil detection schemes [44, 112, 290, 303, 307] leverage the social graph structure to identify whether a given user is Sybil or non-Sybil (Section 8.7.1). On the other hand, Sybil resistance schemes do not explicitly label users' as Sybils or non-Sybils, rather they use application-specific knowledge to mitigate the influence of the Sybils in the network [228, 228, 287] (Section 8.7.2). In a tutorial and survey Haifeng Yu [305] compiles social graph-based Sybil detection techniques. In this paper, we report latest works on that category, as well as Sybil resistance schemes.

8.7.1 Sybil Detection

Sybil detection techniques model an online social network (OSN) as an undirected graph $G = (V, E)$, where a node $v \in V$ is a user in the network and an edge $e \in E$ between two nodes corresponds to a social connection between the users. This connection could be a friendship relationship on Facebook or a colleague relationship on LinkedIn, and is assumed to be trusted.

The social graph has $n = |V|$ nodes and $m = |E|$ edges. By definition, if all nodes correspond to different persons, then the system should have n users. But, some persons have multiple identities. These users are Sybil users and all the identities created by a Sybil user are called *Sybil identities*. An edge between a Sybil user and a non-Sybil user may exist if a Sybil user is able to create a relationship (e.g., friend, colleague) with a non-Sybil user. These types of edges are called *attack edges* (see Figure 67).

Attackers can launch Sybil attacks by creating many Sybil identities and creating attack edges with non-Sybil users. Detection systems against Sybil attacks provide mechanisms to detect whether a user (node) $v \in V$ is Sybil or non-Sybil. Those mechanisms are based on the authority (e.g., the OSN provider) knows the topology of the network (a centralized solution), or a node only knows its social connections (a decentralized solution). Some common assumptions of Sybil detection schemes are below.

- Assumption 1: Attackers can create a large number of Sybil identities in OSNs and can create connections among those Sybil identities, but they lack trust relationships because of their inability to create an arbitrary number of social relationships to non-Sybil users. Intuitively, a social relationship reflects trust and an out-of-band social interaction. So, it requires significant human efforts to establish such a relationship. The limited number of attack edges differentiates Sybil and non-Sybil regions in a social graph as shown in Figure 67.

- Assumption 2: The non-Sybil region of a social graph is fast-mixing. Mixing time determines how fast a random walk's probability of landing at each node reaches the stationary distribution [38, 96]. A limited number of the attack edges causes sparse cut between Sybil and non-Sybil regions. Non-Sybil regions do not show sparse cut as non-Sybils are well connected. As such, there should be a difference in terms of mixing time of the non-Sybil regions compare to the entire social graph.

- Assumption 3: The defense mechanism knows at least one non-Sybil. This assumption is essential in a sense that without this knowledge the Sybil and non-Sybil regions become identical to the system.

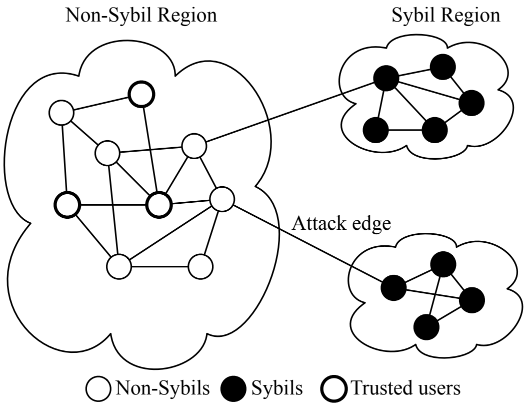


Figure 67 The system model for Sybil detection.

Most of the Sybil detection techniques are based on social graphs. Social graph-based approaches leverage random walks [44, 65, 112, 307], social community [288], and network centrality [299] to detect Sybils in the network. SybilGuard [307] is a decentralized Sybil detection scheme, which uses Assumption 1, Assumption 2 and Assumption 3. A social graph with a small quotient cut has a large mixing time, which implies that a random walk should be long in order to converge to the stationary distribution. So, the presence of too many Sybil nodes in the network disrupts the fast mixing property, in a sense that they increase social network mixing time by contributing small quotient cuts. Thus, a verifier, which is itself a non-Sybil node, can break this symmetry by examining the anomaly of the mixing time in the network. In order to detect Sybils, a non-Sybil node (say a verifier) can perform a random route starting from itself and of a certain length w (a theoretically identifiable quantity, but the paper experimentally shows that this is 2000 for a topology of one-million nodes). A suspect (a node that is in question) is identified as non-Sybil if it's random route intersects with the verifier's random route. As the underlying assumption is that the number of attack edges should be limited, the verifier's route should remain within the non-Sybil region with high probability, given the appropriate choice of w.

SybilInfer's [65] assumptions are also Assumption1, Assumption 2 and Assumption 3. Moreover, it assumes that a modified random walk over a social network, that yields a uniform distribution over all nodes, is also fast mixing. The core of SybilInfer is a Bayesian inference that detects approximate cuts between non-Sybil and Sybil regions in social networks. These identified cuts are used to infer the labels (Sybil or non-Sybil) of the nodes, with an associated probability.

SybilRank [44] is also a random walk-based Sybil detection scheme, which uses all three assumptions and ranks user according to their perceived likelihood of being Sybils. Using early terminated power iteration, SybilRank computes landing probability of random short walks and from that it ranks users, so that substantial portion of the Sybil users have low rank. The design of SybilRank is influenced by an observation on early terminated random walks in social graphs—if a walk of this kind starts from a non-Sybil node, then it has a high degree-normalized landing probability to land at non-Sybil node than a Sybil node. SybilRank terms the probability of a random walk to land on a node as the node's *trust*, ranks nodes based on that and filters lower ranked nodes as potential Sybil users. Rather than keeping computationally intensive a large number of random walk traces used in other graph-based Sybil defense schemes [306, 307], it uses power iteration [169] in calculating the landing probability of random walks.

However, one potential problem with all of the previous approaches is that they don't tolerate noise in the prior knowledge about known non-Sybil or Sybil nodes. SybilBelief [112], another random walk-based semi-supervised learning framework, overcomes the limitation. SybilBelief propagates Sybil/non-Sybil label information from known Sybil/non-Sybil labels to the remaining nodes in the system by modeling the social network as Markov Random Fields. However, the twist is, a portion of these known Sybil/non-Sybil labels might be inaccurate, which earlier approaches fail to address.

Viswanath et al. [288] suggest to use community detection algorithms for Sybils' detection. They show that although other graph property based Sybil defense schemes have different working principles, the core of those works revolves around detecting local communities around a trusted node. So, existing community detection algorithms could be used to defend the Sybils also. Although, not explicitly mentioned, their approach is centralized, because community detection requires a central authority to have the knowledge of the entire topology.

Xu et al. [299] propose Sybil detection based on the betweenness rank of the edges. The betweenness of an edge is defined as the number of shortest paths in the social graph passing the edge [40]. The scheme assumes that the number of attack edges is limited and Sybil and non-Sybil regions are separate clusters of nodes. So, intuitively betweenness scores of the attack edges should be high as they connect the clusters. Their scheme exploits this social network property and uses a Sybil Resisting Network Clustering (SRNC) algorithm to detect Sybils. The algorithm computes the betweenness of each edge and identifies the edges with high betweenness as attack edges.

Social graph based approaches still have some challenges to overcome. First, as graph-based Sybil detection schemes exploit trust relations, the success of the identification highly depends on the trust related assumptions. If an assumption is not right in a network, social graph-based Sybil detection techniques might work poorly in that network (e.g., [156]). For example, the assumption that Sybils' have problems in creating social connections with legitimate users (non-Sybils) is not well established. Although study [208] shows that most of a Sybil identity's connections are also Sybil identities and Sybils' have less relationships with non-Sybil users, several other studies [27, 34, 137] show that users are not careful while accepting friendship requests and Sybil identities can easily befriend with them. Moreover, Sybil users are using advanced techniques to create more realistic Sybil identities, either by copying profile data from existing accounts, or by assigning real users to customize them. Social graph-based Sybil detection techniques are vulnerable to such adversarial social engineering attacks. So, recently researchers have focused on combining user-level activity footprint and graph-level structures (e.g., Íntegro [33], VoteTrust [301]).

Also, another assumption that a social network is fast-mixing may not be right for all social networks. Study [206] shows that many of the social networks are not fast-mixing, especially where edges represent strong real-world trust (e.g., DBLP, Epinions, etc.).

Second, the performance of random walk-based Sybil detection techniques depends on the various relevant parameters of the random walks (e.g., the length of a random walk). These factors will work for a fixed network size (as all the schemes have shown), but they have to be updated with the evolution of the social networks.

8.7.2 Sybil Resistance

Sybil resistance schemes do not explicitly label users' as Sybils and non-Sybils, rather they attempt to mitigate the impact that a Sybil user can have on others. Sybil resistance schemes have been effectively used in applications from diverse domains including content rating systems [51, 275], spam protection [205], online auctions [228], reputation systems [73], and collaborative mobile applications [233].

Note two assumptions of Sybil detection schemes: 1) non-Sybil region is fast mixing, 2) Sybils can not create an arbitrary number of social relationships with non-Sybils. Sybil resistance schemes also assume that non-Sybils' have a limited number of social connections, but they do not rely on the fast mixing nature of the non-Sybil regions. However, Sybil resistance schemes take an additional application related information such as users' interactions/transactions/votes etc. Using the underlying social network of the users and system information, Sybil resistance schemes determine whether an action performed by a user should be allowed or denied.

Most of the Sybil resistance schemes [205, 228, 287] share a common approach in resisting Sybils—they use a *credit network* built on the top of the social network of users [284]. Originally proposed in the electronic commerce community, *Credit Networks* [64, 107] create mutual trust protocols in a situation where there is pairwise trust between two users, and a centralized trusted party is unavailable. Nodes in a credit network trust each other by providing

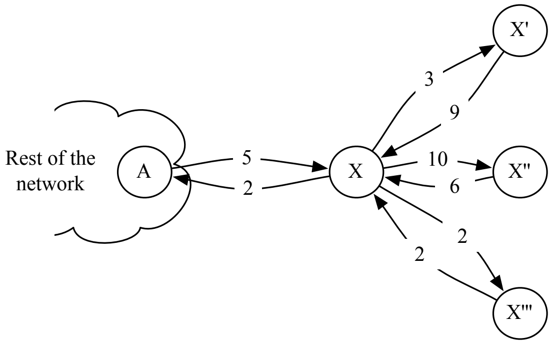


Figure 68 Credit network based Sybil resistance [284]. The network contains four Sybil identities as nodes X, X', X'', X''' of a Sybil user. A directed edge (X,Y) represents how much credit is available to X from Y. If X wants to pay credits from other three nodes, the credits must be deducted from X's single legitimate link to A. So, a Sybil's other identities do not provide any additional credits in the rest of the network.

credits up to a certain limit. Nodes use these credits to pay for services (e.g., sending a message, purchase items, vote casting) that they receive from one another. These schemes assign credits to the network links, and allow an action between two nodes if there is a path between them that has enough credit to satisfy the operation. As such, these schemes find a credit assignment strategy in the graph and apply the credit payment scheme to allow a limited number of illegitimate operations in the system. A Sybil user has limited number of edges with non-Sybils (hence, limited credits available), which restricts her to gain additional advantages by creating multiple Sybil identities. This scenario is shown in figure 68, which is a core defense philosophy of some resistance schemes. In the following, we provide a brief overview of the Sybil resistance schemes.

Ostra [205] leverages existing trust relationships among users to thwart unwanted communication (e.g., spam). It bounds the total number of unwanted communications a Sybil user can produce by assigning credit values to the trust links. If a user sends a message to another user, Ostra finds a path with enough credit from the sender to the receiver. If a path is available, credit is assigned along all the links in the path, which is refunded if the receiver considers the messages as not unwanted. However, if no such path exists, Ostra blocks the communication, but the credit is paid. In this way, Ostra ensures that a user with multiple identities cannot send a large number of unwanted communications, unless she also has additional trust relationships.

Bazaar [228] is targeted to strengthen the users' reputation in online marketplaces like eBay. The opportunity to create accounts freely leads Sybil users to create multiple accounts and causes the waste of time and significant monetary losses for defrauded users. To mitigate Sybil users, Bazaar creates transaction network by linking users who have made a successful transaction. The weight of a link is the amount that has been successfully transferred due to the transaction. Prior to a transaction, using a max flow based technique, Bazaar computes the reputation of the users doing the transaction and compares with the new transaction value. If it finds available flow, it removes the value of the transaction between the users as credits, and eventually adds back if the transaction is a fraud. However, a new transaction is denied if essential flow is not found.

Canal [287] complements Ostra and Bazaar credit networks-based Sybil resistance schemes by applying landmark routing-based techniques in calculating credit payments over a large network. One of the major problems of Ostra and Bazaar is that they require computing max-flow over a graph. However, the huge size of present day network (Facebook has over billion of nodes in social graph) leads to significant computation complexity to compute the max-flow between two nodes in the network. As such, this poses a bottleneck to those techniques to practically deploy in a real-world social network. Canal efficiently computes an approximate max-flow (compromising accuracy with speed-up) path using existing landmark routing-based algorithm [118, 278]. The main components of Canal are *universe creator processes* and *path stitcher processes*. Universe creator processes continuously select new landmarks and path stitcher processes continuously process incoming credit payment requests. Using real-world network datasets the authors show that Canal can perform payment calculations efficiently (within a few milliseconds), even if the network contains hundreds of millions of links.

MobID [233] makes co-located mobile devices resilient to Sybil attackers. Portable devices in close proximity of each other could collaborate various services (e.g., run localization algorithms to get a precise street map), which is severely disrupted by Sybil users (Sybils could inject false information). MobID uses mobile networks to Sybil resilience. More specifically, a device manages two small networks as it meets with other devices. A network of friends contains non-Sybil devices and a network of foes contains suspicious devices. Using two networks, MobID determines whether an unknown device is attempting a Sybil attack. MobID ensures that a non-Sybil device accepts, and accepted by most other non-Sybil devices with high probability. So, a non-Sybil device could successfully trade services with other non-Sybil devices.

8.8 Mitigating Attacks from Compromised Accounts

A compromised account is a legitimate account that has been hacked and taken over by an attacker [84]. The attacker can exploit the account for a number of mischievous purposes, such as, spreading contents via wall posts or direct messages, liking commercial social pages, and following others. Note that compromised accounts are different than Sybil or fake accounts in that compromised accounts have an established trust relationship with others. So, they are not deemed suspicious to OSNs. Attackers exploit this embedded trust and use the accounts for increasing influence and power.

User accounts can be compromised in a number of ways. Users might trust third-party websites or applications with their OSN credentials and those third-parties might be malicious. Users' account passwords are sometimes weak and bots could guess them. Attackers also use cross-site scripting and social phishing to compromise users' accounts.

Compromised accounts have negative consequences on the OSN. They damage the reputation of the system by providing fake like, following and promoting unwanted content. Victims of the compromised accounts lose their accounts and hence their social connections. A research [309] on Twitter compromised accounts shows that about 27% of the compromised users change to a new account once their accounts are compromised.

A number of solutions [45, 84, 285] have been proposed to detect compromised accounts in OSNs. These solutions exploit behavioral deviation of the account before and after an account is compromised. Compa [84] detects compromised accounts using statistical modeling of user behavior and anomaly detection. It makes two assumptions—(1) a compromised account will show noticeable behavioral differences compared to the behavior of the legitimate owner of the account, and (2) an attacker will spread the same malicious content (e.g., tweet or messages) from a subset of account it has compromised. Compa makes behavioral profile of a user and checks for a sudden and significant violation of disseminated content from the profile. It makes the behavioral profile of a user considering content features, such as time of posting, the source of the content (e.g., third-party applications), language, links of the content, interaction and proximity to other users. Then it computes an anomaly score for a new content comparing it to the user's already established profile. The user is put in a suspicious category if a significant portion of new messages has higher anomaly scores. Finally, Compa groups similar content and hence compromised accounts using two similarity measures: content similarity and url similarity. One potential problem with Compa is that attackers can still dodge the system by not posting the same messages from the accounts it has compromised.

SynchroTrap [45] also assumes that the compromised accounts act together in different social network contexts (e.g., page like, photo upload, follow others). It further assumes that those actions are loosely synchronized. SynchroTrap is a more generalized version of Compa for any social network context, as it assumes that actions (e.g., photo upload) are only coordinated, the action might be content independent (e.g., follow others). However, SynchroTrap makes the real difference in terms of scalability. As it is built as an incremental processing system, it can efficiently process massive OSN user activity data. (The system was successfully deployed on Facebook and Instagram). First, SynchroTrap abstracts users' actions using tuples—a combination of user ID, timestamp of actions, type of action (e.g., posting), IP address of the users. Then, for each pair of users, using Jaccard similarity, it computes similarity between two users for their actions during a period of time. Finally, it uses hierarchical clustering algorithms to group users having the similar actions.

Viswanath et al. [285] also uses anomaly detection techniques to detect compromised accounts (anomalous behavior in general). But the difference with Compa is that it doesn't make any assumptions about attack strategy (e.g., Compa assumes coordinated posting of the same content from compromised accounts). They focus on modeling Facebook *Like* activity behavior of normal users. In so doing, they use features, such as, *temporal* (number of likes per day), *spatial* (number of likes in different categories), and *spatio-temporal* (summary of the distribution of like categories using entropy). Of these features, they use principal component (PCA) analysis technique to detect the features that best explain normal user behavior. They experimented with Facebook, Yelp and Twitter datasets and found that three to five principal components are enough to model normal users' behavior. These components are later used to flag anomalous users, whose behavior don't fit the components. Using ground truth data from Facebook compromised users, the authors showed that the technique worked well.

8.9 Mitigating Social Spam

Spam is a news in web-based systems (e.g., [197, 219]). However, OSNs have added a new flavor to it by acting as effective tools for spamming activities and propagation. Social spam (e.g., [318], [178]) is unwanted content that is directed specifically at users of the OSN. The worst consequences of social spam include phishing attacks [139] and malware propagation [36].

Spamming activity is pervasive in OSNs and spammers are successful. For example, about 0.13% of spam tweets in Twitter generate a page visit [115], which is only 0.003%-0.006% for spam email [144]. This high click-through is due to the fact that OSNs expose intrinsic trust relationship among online friends. As such, users read and click messages or links that are shared from their friends. Study [27] shows that 45% of users on OSNs click on links posted by their friends' accounts.

Defending spam in OSNs can improve user experience. OSN service providers will also be benefited as this will lessen the system workload in terms of dealing with unwanted communications and contents. Defense mechanisms against spam in OSNs can be classified into two categories: 1) spam content and profile detection, and 2) spam campaign detection. Spam content and profile-level detection involve checking individual accounts or contents for an evidence of spam contents (Section 8.9.1). On the other hand, a spam "campaign" is a collection of malicious content having a common goal, for example, selling backdoor products [102] (Section 8.9.2).

8.9.1 Spam Content and Profile Detection

Some early spam profile detections [170, 256, 292] used social honeypots. A honeypot is a trap deployed to capture examples of nefarious activities in networked systems [256]. For years, researchers have used honeypots to characterize malicious hacker activities [255], to obtain footprints of email address crawlers [229], and to create intrusion detection signatures [160]. Social honeypots are used to monitor spammers' behaviors and store their information from the OSNs [170].

Webb et al. [292] took the first step to characterize spam in OSNs using social honeypots. They created honeypot MySpace profiles in different geographic locations for harvesting deceptive spam profiles on MySpace. An automated program (commonly known as bots) worked on behalf of a honeypot profile and collected all of the traffic it received (via friend requests). After four months of the deployment and operation, the bots collected a representative sample of friend requests (and corresponding spam profiles). Through statistical analysis the authors showed the followings: (i) spam profiles follow distinct temporal patterns in spamming activity; (ii) 57.2% of the "About me" contents of the spam profiles are duplicated; (iii) spam profiles redirect users to predefined web pages.

In [170], the authors also collected spam profiles using social honeypots. But this work is different from the previous one in that it not only collects and characterizes spam profiles, it extracts features from the gathered spam profiles and builds classifiers to detect potential spam profiles. The authors consider four categories of features such as demographics, content, activity and connections from the spam profiles collected from MySpace and Twitter. These features are later used to train machine learning classifiers that are able to distinguish spam and fair profiles.

One of the limitations of these honeypot-based solutions [170, 292] is that they consider all profiles that sent friend requests to honeypots are spam profiles. But in social networks, it is common to receive friend requests from unknown person, who might be legitimate users in the network. The solutions would be more rigorous if legitimate users were not considered. Also, the methods are effective when spammers become friends with the honeypots. Otherwise the honeypots will be able to target only a small subset of the spammers. As such, recent research on honeypot-based spam detection is focusing more on how to build more effective social honeypots (e.g., [302]). Another problem is that, in social networks, friendship is not always required for spamming. For example, in twitter, a spammer can use mention (e.g., @user) tag to send spam tweets to a user.

Stringhini et al.'s solution [264] overcomes some limitations of the previous two honeypot-based papers by using richer feature sets. The authors deployed honeypots accounts on Facebook, Twitter and MySpace; 300 on each platform for about one year and logged the traffic (e.g., friend requests, messages, and invitations). They build classifiers from the following six features: (i) FF ratio: the ratio of the number of friend requests sent by a user and the number of friends she has; (ii) URL ratio: the ratio of the number of messages containing URLs and total messages; (iii) Message similarity: similarity among the messages sent by a user; (iv) Friend choice: the ratio of the total number of names among the profiles' friends, and the number of distinct first names; (v) Messages sent: the number of messages sent by a profile as a feature; and (vi) Friend number: the number of friends a profile has. Finally, the authors manually inspected and labeled profiles as spam and used *Random Forest* algorithm for classification.

Benevenuto et al. [24] detect video polluters such as spammers and promoters in YouTube online video social networks using machine learning techniques. The authors considered three attribute sets: user attributes, video attributes, and social network (SN) attributes in classification. Four volunteers manually analyzed the videos and built a test set of the dataset labeling users as spammers, promoters and legitimate users. They proposed a flat classification approach, which was able to detect correctly 96% of the promoters, 57% of spammers, and wrongly classifying only 5% of the legitimate users. Interestingly, social network attributes performed the worst in classification—only one feature (UserRank) was within the top 30 features.

Kayes et al. [150] identify abusive content providers in community question answering social networks. Similar to the previous approaches, they have used a number of platform-related features to train machine learning classifiers. But the difference is that they not only used users' social network and activity-specific features, but also leveraged the crowd-sourced rule violations reports contributed by the members of the network.

8.9.2 Spam Campaigns Detection

Chu et al. [54] detect social spam campaigns on Twitter using tweet URLs. They collected a dataset of 50 million tweets from 22 million users. They considered tweets having the same URL as a campaign and clustered the dataset into a number of campaigns. The ground truth was produced through manual inspection using Twitter's spam rules and automated URL checking in five services. They obtained a variety of features ranging from individual tweet/account levels to a collective campaign level and built a classification model to detect spam campaigns. Using several classification algorithms they were able to detect spam campaigns with more than 80% success rate. The focus of this solution is spam tweets with URLs. However, Twitter spammers can post tweets without any URL. Even obfuscated URLs (e.g., somethingDOTcom) will make the detection inefficient.

Gao et al. [102] conduct a rigorous and extensive study on detecting spam campaigns in Facebook wall posts. They crawled 187 million Facebook wall posts from about 3.5 million users. Inspired by a study [161] which shows that spamming bot-nets create email spam messages using templates, they consider wall posts having similar texts as a spam campaign. In particular, they model the wall posts as a graph: a post is a node and two nodes are connected by an edge if they have the same destination URL or their texts are very similar. As such, posts from the same spam campaign will make connected subgraphs or clusters. To detect which clusters are from spammers, they use “distribute” coverage and “bursty” natures of spam campaigns. The “distributed” property is characterized based on the number of user accounts posting in the cluster under the intuition that spammers will use a significant number of registered accounts for a campaign. The intuition behind the “bursty” property is that most spam campaigns are the results of coordinated actions of many accounts within short periods of time. Using threshold filters on these two properties they found clusters of wall posts and classified them as potentially malicious spam campaigns. Template-based spam campaign detection has been also done in Twitter [103].

8.10 Mitigating Distributed Denial-of-service (DDoS) Attacks

A denial-of-service (DOS) attack is characterized by an explicit attempt to monopolize a computer resource, so that an intended user cannot use the resource [202]. A Distributed Denial-of-Service attack (DDoS) deploys multiple attacking entities to simultaneously launch the attack (we refer readers [203] for a taxonomy of web-based DDoS attacks and defenses). DDoS attacks in social networks are also common. For example, on August 6, 2009, Twitter, Facebook, LiveJournal, Google’s Blogger, and YouTube were attacked by a DDoS attack [194]. Twitter experienced interrupted service for several hours, users were complaining of not being able to send their Tweets. Facebook users were experiencing longer periods of time (delays) in loading Facebook pages.

Several papers evaluated how a social network could be leveraged to launch a bot-net based DDoS on any target of the Internet, including the social network itself. Athanasopoulos et al. [14] introduce a bot-net “FaceBot” that uses a social network to carry out a DDoS attack against any host on the internet (including the social network itself). They created a real-world Facebook application, “Photo of the Day”, that presents a different photo from National Geographic to Facebook users every day. Every time a user clicks on the application, an image from the National Geographic appears. However, they placed special codes in the application’s source code. Every time a user views the photo, this code sends a HTTP request towards a victim host, which causes the victim to serve a request of 600 KBytes. They used a web server as a victim and observed that the server recorded 6 Mbit per second of traffic. They introduce defense mechanisms which include providing application developers with a strict API that is capable of giving access to resources only related to the system.

Ur and Ganapathy [282] showed how malicious social network users can leverage their connections with hubs to launch DDoS attacks. They created MySpace profiles which befriended hubs in the network. Those profiles posted “hotlinks” to large media files hosted by a victim web server to Hubs’ pages. As hubs receive a large number of hits, a significant number of the visitors would click those hotlinks. As a consequence, it staged a scenario where a flash crowd was sending requests to the victim web server—a denial of service was the result. They proposed several mitigating techniques. One approach is to restrict some privileges of a user when he becomes a hub (e.g., friends of a hub might no longer be able to post comments containing HTML tags to the hub’s page). But this approach unfortunately restricts the user’s freedom on the OSN. So, they propose a focused automated monitoring on a hub or creating a hierarchy of a hub’s friend, so that only close friends will be able to post on a Hub’s profile (the intuition is that close friends will not exploit the hub). Furthermore, they recommend a reputation based system for social networks that scores user behavior. Only users with a higher reputation scores are allowed to post on the Hub's profile.

However, bot-net based DDoS attacks are difficult to mitigate, because of the difficulty to distinguish legitimate communications from those that are part of the attack. As social networks are flourishing, bot-net based DDoS attacks are becoming stronger, because more legitimate users are unwillingly becoming part of an attack.

8.11 Mitigating Malware Attacks

Malicious software (malware) is a program that is specifically designed to gain access, disrupt computer operation, gather sensitive information or damage a computer without the knowledge of the owner. Participatory Internet technologies (e.g., AJAX) and applications (e.g., RSS) have expedited malware attacks, because they enable the participation of the users. OSNs (all of them use participatory technologies and applications) are providing themselves as infrastructures for propagating malware. The "Koobface" is probably the best example of malware propagation using social networks [89]. It spread rapidly through Facebook social networks. The malware used Facebook credentials on a compromised computer and sent messages to the owner's Facebook friends. The messages redirected the owner's friends to a third-party website and they were asked to download an update of the Adobe Flash player. If they would download and install the file, Koobface would install and infect their system using the same process.

In a survey, Gao et al. [101] discuss a number of methods in which malware propagates through social networks. For example, using cross-site request forgery (CSRF or XSRF) malware invites legitimate users to click on a link. If a user clicks, it opens an exploited page containing malicious scripts. Eventually, the malware submits a message with a URL for a wall post on the user's profile and clicks on the "Share" button so that all of her friends can see this message as well as the link. URL obfuscations are also widely used for malware attacks. An attacker uses commonly known URL shorteners to obfuscate the true location of a link and lures other users to click it.

Unfortunately, malware propagation on social networks exhibits unique propagation vectors. As such, existing Internet worm detection techniques (e.g., [87]) cannot be applied to them. In the context of OSNs, Xu et al. [300] proposed an OSN malware detection system by leveraging both the propagation characteristics of the malware and the topological properties of OSNs. They introduced a “maximum coverage algorithm” that picks a subset of legitimate OSN users to whom the defense system attaches “decoy friends” to monitor the entire social graph. When the decoy friends receive suspicious malware propagation evidence, the detection system performs local and network correlations to distinguish actual malware evidence from normal user communication. However, the challenge for this honeypot-based approach is to determine how many social honeypots (in this context decoy friends) large-scale OSNs (e.g., billions of Facebook users) should deploy.

8.12 Challenges and Future Research Directions

As the OSNs are enjoying unprecedented popularity, keeping users engaged with new functionalities, new privacy and security threats are emerging. The dynamic landscape of privacy and security attacks have enabled researchers to continuously looking forward for new threats and provide mitigating techniques. However, there are open problems that still withstand the plethora of solutions in the literature. An overview of the problems is presented below.

The privacy solutions reviewed in Section 8.2 are focused on specific aspect of privacy, such as, enabling granular settings, providing visual feedback through designing user friendly and informed graphical interface, or generating automated or default privacy policies. However, an integrated privacy solution covering all the planes is still expected. Such a solution might face multiple challenges, e.g., too much granular privacy settings would be a problem in designing succinct interfaces, automated or default privacy policies might have different interface requirements. Moreover, automated and default privacy solutions also have bootstrapping problems to overcome. A newly joined OSN user has no history of interactions that could be used as an input of automation.

A body of literature has used third-party platforms for protecting users from social applications (Section 8.3) and from OSNs (Section 8.4.1). The third-party platforms have been used for appropriate norm following execution of social applications and limiting the transfer of the social data from applications to other parties. Third-party platforms have been also used to hide users' real data to protect them from the OSN. However, those solutions themselves have to be trusted by users and by applications, as they are expected to protect users' personal data and enable a third-party application's execution. Moreover, research needs to propose promising business models for those platforms, because hosting and executing applications on those platforms have a high requirement of logistics and maintenance.

One possible future research direction includes understanding the privacy leakage and associated risks when OSNs work as a Web tracker. OSNs (e.g., Facebook, Twitter) continue to be the login of choice for many websites and applications. As such, OSNs can track their users in third-party websites by placing cookies to users' devices on behalf of those websites. Note that OSNs already know what users do in their platforms. Tracking the users in third-party websites enables them to create a more detailed user profile. As such, OSNs could essentially work as a traditional third-party Web aggregator by offering advertisers targeted advertising in publishers' websites. In general, third-party Web tracking has seen much policy debate recently [164, 193, 273], and OSNs have aggravated the tracking. Research could explore a comprehensive risk assessment and solutions considering OSNs as potential trackers.

The attacks discussed in this chapter are often closely intertwined. User data collected though crawling attacks or via social applications may help an attacker to create background knowledge for launching de-anonymization attacks. An attacker might possess an unprecedented number of user accounts using malware and Sybil attacks and could use those accounts for social spam propagation and distributed denial-of-service attacks. Social spam can also be used to propagate malware. Some attacks might be a pre-requisite for another attacks. For example, a de-anonymization attack can reveal the identity of an individual. That identity could be used to launch an inference attack and to learn unspecified attributes of an individual. Researchers still need to explore the attacks that are are synergies of attacks.

8.13 Summary and Discussions

Millions of Internet users are using OSNs for communication and collaboration. Many companies rely on OSNs for promoting their products and influencing the market. It becomes harder and harder to imagine life without the use of OSN tools, whether for creating an image of oneself or organization, for selectively following news as filtered by the group of friends, or for keeping in touch. However, the growing reliance on OSNs is impaired by an increasingly more sophisticated range of attacks that undermine the very usefulness of the OSNs.

This chapter reviews online social networks' privacy and security issues. We have categorized various attacks on OSNs based on social network stakeholders and the forms of attack targeted at them. Specifically, we have categorized those attacks as attacks on users and attacks on the OSN. We have discussed how the attacks are launched, what are the available defense techniques and what are the challenges involved in such defenses.

In online social networks, privacy and security issues are not separable. In some contexts privacy and security goals may be the same, but there are other contexts where they may be orthogonal, and there are also contexts where they are in conflict. For example, in an OSN, a user wants privacy when she is communicating with other users though the messaging service. She will expect that non-recipients of the message will not be able to read it. OSN services will ensure this by providing a secure communication channel. In this context, the goals of security and privacy are the same. Consider another context where there is a security goal of authenticating a user's account. OSNs usually do this by sending an activation link as a message to the user's e-mail address. This is not a privacy issue—OSNs are just securely authenticating that malicious users are not using the legitimate user's e-mail to register. In this context, security and privacy goals are orthogonal. However, anonymous views in OSNs (e.g., LinkedIn) present a context where security and privacy goals are in conflict. Users may want to have privacy (e.g., anonymization) while viewing other users' profiles. However, the viewee might want to secure her profile from anonymous viewing.

There are also several functionality-oriented attacks that we did not discuss in this chapter. Functionality-oriented attacks attempt to exploit specific functionalities of a social network. For example, Location-based Services (LSP) such as Foursquare, Loopt and Facebook Places utilize geo-location information to publish users' checked-in places. In some LSP, users can accumulate "points" for "checking in" at certain venues or locations and can get real-world discounts or freebies in exchange for these points. There is a body of research that analyzes the technical feasibility of anonymous usage of location-based services so that users are not impacted by location sharing [117, 298]. Moreover, real-world rewards and discounts give incentives for users in LSP to cheat on their locations, and hence research [124, 316] has focused on how to prevent users from location cheating.

OSNs and social applications are here to stay, and while they mature, new security and privacy attacks will take shape. Technical advances in this area can only be of limited effect if not supported by legislative measures for protecting the user from other users and from the service providers [218].

CHAPTER 9: CONCLUSIONS

This dissertation studies content abuse and privacy in online social networks. Abusive behavior has a negative impact on the OSNs, victims of abuse refrain temporarily from the community and ultimately they might leave the community. We focused on CQA social networks and used user-contributed abuse reports to understand the behavior of content abusers and eventually leverage the reports to automatically identify content abusers. Analyzing more than one year of recorded user activity from a mature CQA platform, Yahoo Answers, we found that the use of flags is overwhelmingly correct, and correctly flagged contents are removed quickly. We proposed deviance scores by quantifying how much a user deviates from the norm in terms of received flags based on her activity and found that extreme deviant users are suspended from the community. However, the presence of moderate deviant (not suspended) users is not necessarily bad for the community: they create engagement and have a higher qualitative contribution compared to the suspended users. The social network by which the users are connected channels user attention, monitors behavior and shows homophily. Our empirical investigations lead us to build classifiers that used activity-based and social network-based features and successfully identified fair and suspended users with an accuracy as high as 83%.

Our analysis on the association between users' engagement and privacy concerns in *YA* social networks revealed that privacy preference is correlated with behavior. We found that privacy-concerned users showed higher retention, had higher qualitative and quantitative contributions, reported more abuses, had higher perception on answer quality and had larger social circles. However, these users also exhibited more deviant behavior than the users with public profiles.

Next, this dissertation has shed light on the collective user behavior that we already discussed, such as contribution, privacy concerns in CQA social networks. Using Geert Hofstede's *cultural dimensions* and Robert Levine's *Pace of Life*, we showed that cultural differences exist in *YA*. We found that temporal predictability of activities, contribution-related behavioral patterns, privacy concerns, and power inequality significantly vary across countries in *YA*.

Our analysis of *YA* also found that the majority of users do not change platform-provided default privacy settings. In fact, this is a common scenario in other general-purpose or interest-based social networks also (e.g., Facebook, Twitter, Blogster). Social ecosystems—the collection of rich datasets of user-to-user interactions in support of social applications—further aggravates the problem by collecting richer and contextual user data. We presented *Aegis*, a privacy model for social ecosystems based on the semantic web standard, to limit the vulnerabilities associated with traditional default permissive privacy policies. The privacy model leverages contextual integrity and generates extensible, fine-grained and expressive default policies to protect users' information from other users. We provided an architecture and a prototype implementation of the privacy model that enforced access control policies on a social ecosystem knowledge base. Our experimental evaluation on three real-world large networks showed that it scales well, and policy enforcement does not impose significant overhead.

There are several directions of future work related to this dissertation. Our content abuser work in Chapter 4 shows that moderately deviant users in the CQA community create engagement by attracting more answers and unique answerers to their posted questions. One study [122] finds that users who ask conversational questions tend to have more neighbors (with whom the asker has interaction) than users who ask informational questions. This might suggest that deviant users tend to ask more conversational questions, which engage a larger number of responders. Understanding what makes deviant users engaging can be helpful in designing strategies potentially applicable to a variety of communities. However, due to the observational nature of the study, we couldn't draw any causal relationships. We want to further investigate whether deviant users ask conversational questions. This requires a linguistic analysis of the questions posted by the users.

We also want to characterize and identify *prosocial* users in CQA social networks. In social psychology, prosocial behavior has been related to multilevel perspective of behaviors that manifest concerns for the well-being of others [79]. This includes volunteering, cooperative behavior, organizational citizenship behavior. In CQAs, a number of behaviors could be identified as prosocial: providing answers to questions, participating in the community, voting to select best answers, reporting abuses correctly etc. In our work, we have characterized the efficient abuse reporters only. In the future, we want to characterize and identify *prosocial* users considering a wide spectrum of prosocial behaviors and also using the social networks.

While discussing unethical/deviant behaviors in CQA social networks, we only consider the content (question and answer) abuse. There are other CQA-specific deviant behaviors, that warrant further investigation. One such behavior is *free riding*—getting answers to own questions without answering much. A question-answering community, like *YA*, expects its users to resolve questions. However, a large number of questions still remain unanswered. For example, in our dataset, 35.09% of the questions do not receive any answer. Figure 69 shows the distribution of the questions and answers ratio of users. The distribution shows that about 60% users have asked more questions than they answered and 5.7% users have asked ten times more than they answered. In the future, we want to further understand the free riding behavior in CQAs.

We want to understand Aegis policy framework presented in Chapter 7 in emerging application scenarios, more specifically targeting applications that are built on aggregated social data. These social applications will run on users trusted devices and their access to social data store will be managed by Aegis. In order to experiment with Aegis in a real social ecosystem, we plan to create an “intelligent” mapping of the users in different social network datasets. This mapping will create a unification of identities from datasets and abstract a single user on multiple data sources. We also want to understand the system in different platform settings, such as peer-

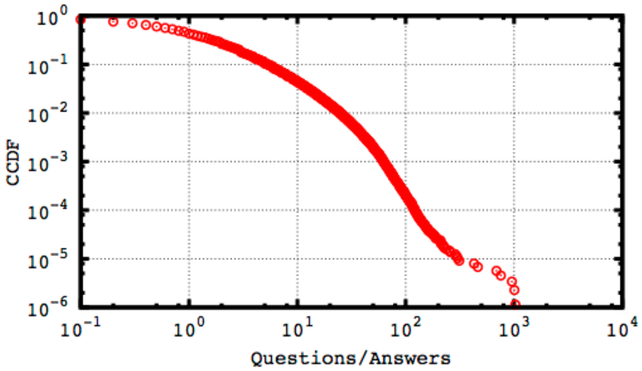


Figure 69 Distribution of the questions and answers ratio per user in *YA*.

to-peer and mobile computing. Moreover, a user study is also required to assess the usability of the framework.

In Chapter 5, we find that privacy-concerned users are more engaged in the community. We want to understand what makes users to change their default privacy settings on a CQA platform and also be more engaged.

REFERENCES

[1] Alessandro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Privacy enhancing technologies*, pages 36–58. Springer, 2006.

[2] Fabeah Adu-Oppong, Casey K Gardiner, Apu Kapadia, and Patrick P Tsang. Social circles: Tackling privacy in social networks. In *Symposium on Usable Privacy and Security (SOUPS)*, 2008.

[3] Charu C Aggarwal, Yao Li, and Philip S Yu. On the hardness of graph anonymization. In *IEEE 11th International Conference on Data Mining (ICDM)*, pages 1002–1007. IEEE, 2011.

[4] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 183–194, New York, NY, USA, 2008. ACM.

[5] L.M. Aiello, M. Milanesio, G. Ruffo, and R. Schifanella. Tempering kademlia with a robust identity based system. In *Peer-to-Peer Computing , 2008. P2P '08. Eighth International Conference on*, pages 30–39, 2008.

[6] Luca Maria Aiello and Giancarlo Ruffo. Lotusnet: tunable privacy for distributed online social network services. *Computer Communications*, 35(1):75–88, 2012.

[7] Alexa. The top 500 sites on the web. http://www.alexa.com/topsites, September 2015.

[8] Yahoo Answers. Yahoo answers community guidelines. http://answers.yahoo.com/info/community_guidelines, 2013.

[9] Yahoo Answers. Points table. https://answers.yahoo.com/info/scoring_system, 2015.

[10] Sinan Aral, Lev Muchnika, and Arun Sundararajana. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *PNAS*, 106(51):21544–21549, 2009.

[11] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[12] Michael Arrington. Aol proudly releases massive amounts of private data. http://tcrn.ch/1pkS6Mi/, August 2006.

[13] Çiğdem Aslay, Neil O'Hare, Luca Maria Aiello, and Alejandro Jaimes. Competition-based networks for expert finding. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 1033–1036. ACM, 2013.

[14] Elias Athanasopoulos, Andreas Makridakis, Spyros Antonatos, Demetres Antoniades, Sotiris Ioannidis, Kostas G Anagnostakis, and Evangelos P Markatos. Antisocial networks: Turning a social network into a botnet. In *11th International Conference on Information Security*, pages 146–160. Springer, 2008.

[15] Paolo Avesani, Paolo Massa, and Roberto Tiella. Moleskiing.it: a trust-aware recommender system for ski mountaineering. *International Journal for Infonomics*, 2005.

[16] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[17] Randy Baden, Adam Bender, Neil Spring, Bobby Bhattacharjee, and Daniel Starin. Persona: an online social network with user-defined privacy. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 135–146, 2009.

[18] Ero Balsa, Laura Brandimarte, Alessandro Acquisti, Claudia Diaz, and Seda Gurses. Spiny cactos: Osn users attitudes and perceptions towards cryptographic access control tools. 2014.

[19] World Bank. Internet users. http://data.worldbank.org/indicator/IT.NET.USER.P2, 2014.

[20] L. Banks and S.F. Wu. All friends are not created equal: An interaction intensity based approach to privacy in online social networks. In *International Conference onComputational Science and Engineering*, pages 970–974, 2009.

[21] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[22] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: framework and applications. In *IEEE Symposium on Security and Privacy*, pages 184–198, 2006.

[23] BBC. Facebook has more than 83 million illegitimate accounts. http://www.bbc.co.uk/news/technology-19093078, Aug 2012.

[24] F. Benevenuto, T. Rodrigues, J Almeida, M. Goncalves, and V. Almeida. Detecting spammers and content promoters in online video social networks. In *INFOCOM Workshops 2009, IEEE*, pages 1–2, 2009.

[25] Shea Bennett. Social networking accounts for (at least) 28% of all media time spent online. http://www.adweek.com/socialtimes/online-activities-2014/500746, 2014.

[26] Andrew Besmer, Heather Richter Lipford, Mohamed Shehab, and Gorrell Cheek. Social applications: exploring a more secure framework. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 2:1–2:10, New York, NY, USA, 2009. ACM.

[27] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 551–560, New York, NY, USA, 2009. ACM.

[28] Igor Bilogrevic, Kévin Huguenin, Berker Agir, Murtuza Jadliwala, Maria Gazaki, and Jean-Pierre Hubaux. A machine-learning based approach to privacy-aware information-sharing in mobile social networks. *Pervasive and Mobile Computing*, 21:1–18, 2015.

[29] Jeremy Blackburn, Nicolas Kourtellis, John Skvoretz, Matei Ripeanu, and Adriana Iamnitchi. Cheating in online games: A social network perspective. *ACM Transactions on Internet Technology*, 13(3):9, 2014.

[30] Jeremy Blackburn, Ramanuja Simha, Nicolas Kourtellis, Xiang Zuo, Matei Ripeanu, John Skvoretz, and Adriana Iamnitchi. Branded with a scarlet ”c”: cheaters in a gaming social network. In *Proceedings of the 21st international conference on World Wide Web*, pages 81–90. ACM, 2012.

[31] J. Bonneau, J. Anderson, and G. Danezis. Prying data out of a social network. In *Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances in*, pages 249–254, 2009.

[32] Joseph Bonneau and Sören Preibusch. The privacy jungle: On the market for data protection in social networks. In *Economics of information security and privacy*, pages 121–167. Springer, 2010.

[33] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. Íntegro: Leveraging victim prediction for robust fake account detection in osns. In *Proc. of NDSS*, 2015.

[34] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 93–102, New York, NY, USA, 2011. ACM.

[35] D Boyd. Friendster and publiclly articulated social networking. In *Extended Abstracts of the Conference on Human Factors and Computing Systems (CHI 2004)*, pages 1279–1282, 2004.

[36] D. Boyd and J. Heer. Profiles as conversation: Networked identity performance on friendster. In *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 3, pages 59c–59c, 2006.

[37] Danah M Boyd and Nicole B Ellison. Social network sites: Definition, history, and scholarship. *Engineering Management Review, IEEE*, 38(3):16–31, 2010.

[38] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1653 – 1664 vol. 3, march 2005.

[39] Tony Bradley. 45,000 facebook accounts compromised: What to know. http://bit.ly/TUY3i8, Jan 2012.

[40] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 40(2):163–177, 2001.

[41] Amy Bruckman, Pavel Curtis, Cliff Figallo, and Brenda Laurel. Approaches to managing deviant behavior in virtual communities. In *CHI Conference Companion*, pages 183–184, 1994.

[42] Sonja Buchegger, Doris Schiöberg, Le-Hung Vu, and Anwitaman Datta. Peerson: P2p social networking: early experiences and insights. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, SNS '09, pages 46–52, New York, NY, USA, 2009. ACM.

[43] Alina Campan and Traian Marius Truta. Data and structural k-anonymity in social networks. In *Privacy, Security, and Trust in KDD*, pages 33–54. Springer, 2009.

[44] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, NSDI'12, 2012.

[45] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 477–488. ACM, 2014.

[46] Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. A semantic web based framework for social network access control. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, SACMAT '09, 2009.

[47] Barbara Carminati, Elena Ferrari, and Andrea Perego. Rule-based access control for social networks. In *Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems*, pages 1734–1744, 2006.

[48] Rory Cellan-Jones. Facebook 'likes' and adverts' value doubted. http://www.bbc.co.uk/news/technology-18813237, July 2012.

[49] Oscar Celma. Foang the music: Bridging the semantic gap in music recommendation. In *Proceedings of the International Semantic Web Conference*, pages 927–93, 2006.

[50] Yuan Cheng, Jaehong Park, and Ravi Sandhu. Preserving user privacy from third-party applications in online social networks. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, pages 723–728. ACM, 2013.

[51] Nitin Chiluka, Nazareno Andrade, Johan Pouwelse, and Henk Sips. Leveraging trust and distrust for sybil-tolerant voting in online social media. In *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, PSOSM '12, pages 1:1–1:8, New York, NY, USA, 2012. ACM.

[52] Hichang Cho, Milagros Rivera-Sánchez, and Sun Sun Lim. A multinational study on online privacy: global concerns and local responses. *New media & society*, 11(3):395–416, 2009.

[53] Hee Chul Choi, Sebastian Ryszard Kruk, Slawomir Grzonkowski, Katarzyna Stankiewicz, Brian Davis, and John Breslin. Trust models for community aware identity management. In *Proceedings of the Identity, Reference and Web Workshop, in conjunction with WWW 2006*, page 140154, 2006.

[54] Zi Chu, Indra Widjaja, and Haining Wang. Detecting social spam campaigns on twitter. In *In Proceedings of Conference on Applied Cryptography and Network Security*, Lecture Notes in Computer Science, pages 455–472. Springer Berlin Heidelberg, 2012.

[55] Privacy Commissioner. Facebook needs to improve privacy practices, investigation finds. https://www.priv.gc.ca/media/nr-c/2009/nr-c_090716_e.asp, July 2009.

[56] Mia Consalvo. *Cheating: Gaining advantage in videogames*. Mit Press, 2007.

[57] Mauro Conti, Arbnor Hasani, and Bruno Crispo. Virtual private social networks and a facebook implementation. *ACM Transactions on the Web (TWEB)*, 7(3):14, 2013.

[58] Thomas M Cover and Joy A Thomas. Elements of information theory 2nd edition. *Willey-Interscience: NJ*, 2006.

[59] Jonathon N. Cummings, Brian Butler, and Robert Kraut. The quality of online social relationships. *Commun. ACM*, 45(7):103–108, July 2002.

[60] L.A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, pages 145–152, 2009.

[61] L.A. Cutillo, R. Molva, and T. Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *Communications Magazine, IEEE*, 47(12):94–101, 2009.

[62] Wis Beaver Dam. School teacher suspended for facebook gun photo. http://www.foxnews.com/story/2009/02/05/schoolteacher-suspended-for-facebook-gun-photo/, February 2009.

[63] danah boyd and Eszter Hargittai. Facebook privacy settings: Who cares? *First Monday*, 15(8), 2010.

[64] Pranav Dandekar, Ashish Goel, Michael P. Wellman, and Bryce Wiedenbeck. Strategic formation of credit networks. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 559–568, New York, NY, USA, 2012. ACM.

[65] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Network and Distributed System Security Symposium (NDSS)*, 2009.

[66] George Danezis. Inferring privacy policies for social networking services. In *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*, AISec '09, pages 5–10, New York, NY, USA, 2009. ACM.

[67] John P Davis. The experience of 'bad' behavior in online social spaces: A survey of online users. *Social Computing Group, Microsoft Research*, 2002.

[68] Norberto Nuno Gomes de Andrade, Aaron Martin, and Shara Monteleone. "all the better to see you with, my dear": Facial recognition and privacy in online social networks. *IEEE Security & Privacy*, 11(3):21–28, 2013.

[69] E. De Cristofaro, A. Durussel, and I. Aad. Reclaiming privacy for smartphone applications. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 84 –92, march 2011.

[70] Marieke De Mooij. The future is predictable for international marketers: converging incomes lead to diverging consumer behaviour. *International Marketing Review*, 17(2):103–113, 2000.

[71] Marieke De Mooij. *Global marketing and advertising: Understanding cultural paradoxes*. Sage Publications, 2013.

[72] David Dearman and Khai N. Truong. Why users of yahoo!: Answers do not answer questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 329–332, New York, NY, USA, 2010. ACM.

[73] Dd.B. DeFigueiredo and E.T. Barr. Trustdavis: a non-exploitable online reputation system. In *E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, pages 274–283, July.

[74] Dominique Devriese and Frank Piessens. Noninterference through secure multi-execution. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP '10, pages 109–124, Washington, DC, USA, 2010. IEEE Computer Society.

[75] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, December 2001.

[76] R. Dey, Cong Tang, K. Ross, and N. Saxena. Estimating age privacy leakage in online social networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 2836–2840, 2012.

[77] JohnR. Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin Heidelberg, 2002.

[78] Jack D Douglas and Frances Chaput Waksler. *The sociology of deviance: An introduction*. Little, Brown Boston, MA, 1982.

[79] John F Dovidio, Jane Allyn Piliavin, David A Schroeder, and Louis Penner. *The social psychology of prosocial behavior.* Lawrence Erlbaum Associates Publishers, 2006.

[80] David M Downes and Paul Rock. *Understanding deviance: a guide to the sociology of crime and rule-breaking*. Oxford University Press, 2011.

[81] Gideon Dror, Yoelle Maarek, and Idan Szpektor. Will my question be answered? predicting "question answerability" in community question-answering sites. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 499–514. Springer, 2013.

[82] Patrick D Dunlop and Kibeom Lee. Workplace deviance, organizational citizenship behavior, and business unit performance: The bad apples do spoil the whole barrel. *Journal of Organizational Behavior*, 25(1):67–80, 2004.

[83] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Third Theory of Cryptography Conference*, pages 265–284. Springer Berlin Heidelberg, 2006.

[84] M Egele, G Stringhini, C Kruegel, and G Vigna. Compa: Detecting compromised social network accounts. In *Symposium on Network and Distributed System Security (NDSS)*, 2013.

[85] Manuel Egele, Andreas Moser, Christopher Kruegel, and Engin Kirda. Pox: Protecting users from malicious facebook applications. *Computer Communications*, 35(12), 2012.

[86] N. Elahi, M. Chowdhury, and J. Noll. Semantic access control in web based communities. In *Proceedings of the 2008 The Third International Multi-Conference on Computing in the GlobalInformation Technology*, pages 131 –136, 27 2008-aug. 1 2008.

[87] Daniel R. Ellis, John G. Aiken, Kira S. Attwood, and Scott D. Tenaglia. A behavioral approach to worm detection. In *Proceedings of the 2004 ACM workshop on Rapid malcode*, WORM '04, pages 43–53, New York, NY, USA, 2004. ACM.

[88] Robert Engelmore, editor. *Readings from the AI magazine*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1988.

[89] Facebook. Facebook's continued fight against koobface. http://on.fb.me/y5ibe1, January 2012.

[90] Facebook. Statement of rights and responsibilities. https://www.facebook.com/legal/terms, January 2015.

[91] Lujun Fang and Kristen LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 351–360, New York, NY, USA, 2010. ACM.

[92] Scott L Feld. Why your friends have more friends than you do. *American Journal of Sociology*, pages 1464–1477, 1991.

[93] Adrienne Felt and David Evans. Privacy protection for social networking apis. In *2008 Web 2.0 Security and Privacy (W2SP08)*, 2008.

[94] Casey Fiesler and Amy Bruckman. Copyright terms in online creative communities. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 2551–2556. ACM, 2014.

[95] Joshua Finnis, Nalin Saigal, Adriana Iamnitchi, and Jay Ligatti. A location-based policy-specification language for mobile devices. *Pervasive and Mobile Computing*, 8:402–414, 2010.

[96] AbrahamD. Flaxman. Expansion and lack thereof in randomly perturbed graphs. *Algorithms and Models for the Web-Graph*, 4936:24–35, 2008.

[97] Philip Fong, Mohd Anwar, Zhen Zhao, Michael Backes, and Peng Ning. *A Privacy Preservation Model for Facebook-Style Social Network Systems*, pages 303–320. Springer Berlin / Heidelberg, 2009.

[98] Philip W.L. Fong. Relationship-based access control: protection model and policy language. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 191–202, 2011.

[99] Charles Fried. Privacy. *Yale Law Journal*, 77(3):475–483, 1968.

[100] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[101] Hongyu Gao, Jun Hu, Tuo Huang, Jingnan Wang, and Yan Chen. Security issues in online social networks. *Internet Computing, IEEE*, 15(4):56–63, 2011.

[102] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

[103] Hongyu Gao, Yi Yang, Kai Bu, Yan Chen, Doug Downey, Kathy Lee, and Alok Choudhary. Spam ain't as diverse as it seems: Throttling osn spam with templates underneath. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 76–85. ACM, 2014.

[104] Ruth Garcia-Gavilanes, Yelena Mejova, and Daniele Quercia. Twitter ain't without frontiers: Economic, social, and cultural boundaries in international communication. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1511–1522, 2014.

[105] Ruth Garcia-Gavilanes, Daniele Quercia, and Alejandro Jaimes. Cultural dimensions in twitter: Time, individualism and power. *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, 13, 2013.

[106] Diego Garlaschelli and Maria I Loffredo. Patterns of link reciprocity in directed networks. *Physical Review Letters*, 93(26):268701, 2004.

[107] Arpita Ghosh, Mohammad Mahdian, DanielM. Reeves, DavidM. Pennock, and Ryan Fugger. Mechanism design on trust networks. In Xiaotie Deng and FanChung Graham, editors, *Internet and Network Economics*, volume 4858 of *Lecture Notes in Computer Science*, pages 257–268. Springer Berlin Heidelberg, 2007.

[108] F. Giunchiglia, Rui Zhang, and B. Crispo. Relbac: Relation based access control. In *Fourth International Conference on Semantics, Knowledge and Grid*, pages 3 –11, 2008.

[109] J. Golbeck and J. Hendler. Reputation network analysis for email ltering. In *Proceedings of the First Conference on Email and Anti-Spam*, 2004.

[110] Jennifer Golbeck and Michael M. Wasser. Socialbrowsing: integrating social networks and web browsing. In *CHI '07 extended abstracts on Human factors in computing systems*, pages 2381–2386, 2007.

[111] Scott A Golder and Michael W Macy. Diurnal and seasonal mood vary with work, sleep, and day length across diverse cultures. *Science*, 333(6051):1878–1881, 2011.

[112] Neil Zhenqiang Gong, Michael Frank, and Payal Mittal. Sybilbelief: A semi-supervised learning approach for structure-based sybil detection. *Information Forensics and Security, IEEE Transactions on*, 9(6):976–987, 2014.

[113] Samuel D Gosling, Oliver P John, Kenneth H Craik, and Richard W Robins. Do people know how they behave? self-reported act frequencies compared with on-line codings by observers. *Journal of personality and social psychology*, 74(5):1337, 1998.

[114] K. Graffi, C. Gross, D. Stingl, D. Hartung, A. Kovacevic, and R. Steinmetz. Lifesocial.kom: A secure and p2p-based solution for online social networks. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 554–558, 2011.

[115] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[116] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[117] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, MobiSys '03, pages 31–42, New York, NY, USA, 2003. ACM.

[118] Andrey Gubichev, Srikanta Bedathur, Stephan Seufert, and Gerhard Weikum. Fast and accurate estimation of shortest paths in large graphs. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 499–508, New York, NY, USA, 2010. ACM.

[119] Saikat Guha, Kevin Tang, and Paul Francis. Noyb: privacy in online social networks. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 49–54, New York, NY, USA, 2008. ACM.

[120] S. Gürses and C. Diaz. Two tales of privacy in online social networks. *Security Privacy, IEEE*, PP(99):1–1, 2013.

[121] Teri Hansen. Social media gives stalkers unprecedented access to victims. http://www.mcphersonsentinel.com/article/20150112/NEWS/150119927, January 2015.

[122] F Maxwell Harper, Daniel Moy, and Joseph A Konstan. Facts or friends?: distinguishing informational and conversational questions in social q&a sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 759–768. ACM, 2009.

[123] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, August 2008.

[124] Wenbo He, Xue Liu, and Mai Ren. Location cheating: A security challenge to location-based social network services. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 740–749. IEEE, 2011.

[125] R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Preventing private information inference attacks on social networks. *Knowledge and Data Engineering, IEEE Transactions on*, 25(8):1849–1862, 2013.

[126] Alex Hern. Twitter announces crackdown on abuse with new filter and tighter rules. http://bit.ly/1K2pMt7, April 2015.

[127] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, November 2007.

[128] Nathan O Hodas, Farshad Kooti, and Kristina Lerman. Friendship paradox redux: Your friends are more interesting than you. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, 2013.

[129] Geert Hofstede. National cultures in four dimensions: A research-based theory of cultural differences among nations. *International Studies of Management & Organization*, pages 46–74, 1983.

[130] Geert Hoftede, Gert Jan Hofstede, and Michael Minkov. *Cultures and organizations: software of the mind: intercultural cooperation and its importance for survival*. McGraw-Hill, 2010.

[131] HostIP. Using the ip addresses database-ip address lookup. http://www.hostip.info/use.html2, 2014.

[132] Pili Hu, Ronghai Yang, Yue Li, and Wing Cheong Lau. Application impersonation: problems of oauth and api design in online social networks. In *Proceedings of the second edition of the ACM conference on Online social networks*, pages 271–278. ACM, 2014.

[133] Bryan W Husted. The impact of national culture on software piracy. *Journal of Business Ethics*, 26(3):197–211, 2000.

[134] Tim Hwang, Ian Pearce, and Max Nanis. Socialbots: Voices from the fronts. *interactions*, 19(2):38–45, March 2012.

[135] Adriana Iamnitchi, Jeremy Blackburn, and Nicolas Kourtellis. The social hourglass: An infrastructure for socially aware applications and services. *IEEE Internet Computing*, 16:13–23, 2012.

[136] Johannes Illenberger and Gunnar Fltterdb. Estimating network properties from snowball sampled data. *Social Networks*, 34(4):701 – 711, 2012.

[137] Danesh Irani, Marco Balduzzi, Davide Balzarotti, Engin Kirda, and Calton Pu. Reverse social engineering attacks in online social networks. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 55–74. Springer, 2011.

[138] Gregoire Jacob, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. Pubcrawl: Protecting users and businesses from crawlers. In *USENIX Security Symposium*, pages 507–522, 2012.

[139] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, October 2007.

[140] Leonard M Jessup, Terry Connolly, and Jolene Galegher. The effects of anonymity on gdss group process with an idea-generating task. *Mis Quarterly*, pages 313–321, 1990.

[141] Prashant Jha. Facebook users could swing the results in 160 lok sabha constituencies. http://goo.gl/9A6FR, April 2013.

[142] Kuen-Hee Ju-Pak. Content dimensions of web advertising: a cross-national comparison. *International Journal of Advertising*, 18(2):207–231, 1999.

[143] Matt Jurek. Google explores +1 button to influence search results. http://www.tekgoblin.com/2011/08/29/google-explores-1-button-to-influence-search-results/, Aug 2011.

[144] Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage. Spamalytics: an empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 3–14, New York, NY, USA, 2008. ACM.

[145] Imrul Kayes and Adriana Iamnitchi. Aegis: A semantic implementation of privacy as contextual integrity in social ecosystems. In *IEEE 11th International Conference on Privacy, Security and Trust (PST)*, pages 88–97. IEEE, July 2013.

[146] Imrul Kayes and Adriana Iamnitchi. Out of the wild: On generating default policies in social ecosystems. In *IEEE ICC'13 - Workshop on Beyond Social Networks: Collective Awareness*, pages 204–208. IEEE, June 2013.

[147] Imrul Kayes and Adriana Iamnitchi. A survey on privacy and security in online social networks. *arXiv preprint arXiv:1504.03342*, 2015.

[148] Imrul Kayes, Nicolas Kourtellis, and Adriana Iamnitchi. Privacy concerns vs. user behavior in community question answering. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM'15. IEEE, 2015.

[149] Imrul Kayes, Nicolas Kourtellis, Daniele Quercia, Adriana Iamnitchi, and Francesco Bonchi. Cultures in community question answering. In *Proceedings of the ACM 26th Conference on Hypertext and Social Media*, HT '15. ACM, 2015.

[150] Imrul Kayes, Nicolas Kourtellis, Daniele Quercia, Adriana Iamnitchi, and Francesco Bonchi. The social world of content abusers in community question answering. In *Proceedings of the 24th International World Wide Web Conference*, WWW '15, pages 570–580. ACM, 2015.

[151] Imrul Kayes, Xiaoning Qian, John Skvoretz, and Adriana Iamnitchi. How influential are you: Detecting influential bloggers in a blogging community. In *Proceedings of the 4th international conference on Social Informatics*, pages 29–42. Springer, 2012.

[152] Imrul Kayes, Xiang Zuo, Da Wang, and Chakareski Jacob. To blog or not to blog: Characterizing and predicting retention in community blogs. In *Proceedings of the 7th ACM/ASE International Conference on Social Computing (SocialCom'14)*, pages 7:1–7:8. ACM, 2014.

[153] Heather Kelly. Police embrace social media as crime-fighting tool. http://www.cnn.com/2012/08/30/tech/social-media/fighting-crime-social-media/, August 2012.

[154] S. Kelly. Identity 'at risk on facebook. http://news.bbc.co.uk/2/hi/programmes/click_online/7375772.stm, 2008.

[155] Eyun-Jung Ki, Byeng-Hee Chang, and Hyoungkoo Khang. Exploring influential factors on music piracy across countries. *Journal of Communication*, 56(2):406–426, 2006.

[156] D. Koll, Jun Li, J. Stein, and Xiaoming Fu. On the state of osn-based sybil defenses. In *IFIP Networking Conference*. IEEE, 2014.

[157] J.S. Kong, B.A. Rezaei, N. Sarshar, V.P. Roychowdhury, and P.O. Boykin. Collaborative spam filtering using e-mail networks. *Computer*, 39(8):67 –73, aug. 2006.

[158] Nicolas Kourtellis, Joshua Finnis, Paul Anderson, Jeremy Blackburn, Cristian Borcea, and Adriana Iamnitchi. Prometheus: User-controlled p2p social data management for socially-aware applications. In *11th International Middleware Conference*, November 2010.

[159] Nicolas Kourtellis, Joshua Finnis, Paul Anderson, Jeremy Blackburn, Cristian Borcea, and Adriana Iamnitchi. Prometheus: User-controlled p2p social data management for socially-aware applications. In *11th International Middleware Conference*, November 2010.

[160] Christian Kreibich and Jon Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev.*, 34(1):51–56, January 2004.

[161] Christian Kreibich, Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M Voelker, Vern Paxson, and Stefan Savage. Spamcraft: An inside look at spam campaign orchestration. *Proc. of 2nd USENIX LEET*, 2009.

[162] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, pages 19–24, 2008.

[163] Balachander Krishnamurthy and Craig E. Wills. Characterizing privacy in online social networks. In *Proceedings of the first workshop on Online social networks*, pages 37–42, 2008.

[164] Bharadwaj Krishnamurthy. Privacy and online social networks: can colorless green ideas sleep furiously? *IEEE Security & Privacy*, 11(3):14–20, 2013.

[165] S.R. Kruk. Foaf-realm: control your friends access to the resource. In *In Proceedings of the 1st Workshop on Friend of a Friend*, 2004.

[166] S.R. Kruk, S. Grzonkowski, H.C. Choi, T. Woroniecki, and A. Gzella. D-foaf: Distributed identity management with access rights delegation. In *Proceedings of the 1st Asian Semantic Web Conference (ASWC 2006)*, page 140154, 2006.

[167] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, pages 591–600. ACM, 2010.

[168] Cliff Lampe, Nicole B. Ellison, and Charles Steinfield. Changes in use and perception of facebook. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 721–730, 2008.

[169] Amy N. Langville and Carl D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1:2004, 2004.

[170] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: social honeypots + machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 435–442, New York, NY, USA, 2010. ACM.

[171] Wesley Lefebvre. Best q&a search engines. http://www.listofsearchengines.org/qa-search-engines, 2014.

[172] J. Leskovec. Social computing data repository at ASU. http://snap.stanford.edu/data/, 2008.

[173] Robert Levine. *A Geography of Time: The Temporal Misadventures of a Social Psychologist or How Every Culture Keeps Time Just a Little Bit Differently*. University Press, 2006.

[174] Robert V Levine and Ara Norenzayan. The pace of life in 31 countries. *Journal of cross-cultural psychology*, 30(2):178–205, 1999.

[175] Theodore Levitt. The globalization of markets. *International Business: Strategic management of multinationals*, 3:18, 2002.

[176] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 3–12, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[177] Baichuan Li, Tan Jin, Michael R. Lyu, Irwin King, and Barley Mak. Analyzing and predicting question quality in community question answering services. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 775–782, New York, NY, USA, 2012. ACM.

[178] Yu-Ru Lin, Hari Sundaram, Yun Chi, Junichi Tatemura, and Belle L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 1–8, New York, NY, USA, 2007. ACM.

[179] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 1145–1146, New York, NY, USA, 2009. ACM.

[180] Heather Richter Lipford, Andrew Besmer, and Jason Watson. Understanding privacy settings in facebook with an audience view. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 2:1–2:8, 2008.

[181] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. ACM.

[182] Qiaoling Liu and Eugene Agichtein. Modeling answerer behavior in collaborative question answering systems. In *Advances in Information Retrieval*, pages 67–79. Springer, 2011.

[183] Qiaoling Liu, Eugene Agichtein, Gideon Dror, Yoelle Maarek, and Idan Szpektor. When web search fails, searchers become askers: Understanding the transition. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 801–810, New York, NY, USA, 2012. ACM.

[184] Yabing Liu, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 61–70, 2011.

[185] Gilad Lotan, Erhardt Graeff, Mike Ananny, Devin Gaffney, Ian Pearce, et al. The arab spring— the revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International Journal of Communication*, 5:31, 2011.

[186] Wentian Lu and Gerome Miklau. Exponential random graph estimation under differential privacy. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 921–930. ACM, 2014.

[187] Matthew M. Lucas and Nikita Borisov. Flybynight: mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, WPES '08, pages 1–8, New York, NY, USA, 2008. ACM.

[188] Wanying Luo, Qi Xie, and U. Hengartner. Facecloak: An architecture for user privacy on social networking sites. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 3, pages 26–33, 2009.

[189] Michelle Madejski, Maritza Lupe Johnson, and Steven Michael Bellovin. The failure of online social network privacy settings. *Department of Computer Science, Columbia University*, 2011.

[190] Daily Mail. Bank worker fired for facebook post comparing her 7-an-hour wage to lloyds boss's $4,000$-an-hour salary. http://dailym.ai/fjRTlC, April 2011.

[191] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2857–2866, New York, NY, USA, 2011. ACM.

[192] Amirreza Masoumzadeh and James Joshi. Ontology-based access control for social network systems. *IJIPSI*, 1(1):59–78, 2011.

[193] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *EEE Symposium on Security and Privacy*, pages 413–427. IEEE, 2012.

[194] Caroline McCarthy. Twitter crippled by denial-of-service attack. http://news.cnet.com/8301-13577_3-10304633-36.html, August 2009.

[195] Matt McGee. Yahoo answers hits one billion answers. http://searchengineland.com/yahoo-answers-hits-one-billion-answers-41167/, 2010.

[196] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[197] Bhaskar Mehta, Saurabh Nangia, Manish Gupta, and Wolfgang Nejdl. Detecting image spam using visual features and near duplicate detection. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 497–506, New York, NY, USA, 2008. ACM.

[198] Giovanna Menardi and Nicola Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122, 2014.

[199] Eduarda Mendes Rodrigues and Natasa Milic-Frayling. Socializing or knowledge sharing?: Characterizing social intent in community question answering. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1127–1136, New York, NY, USA, 2009. ACM.

[200] E. Mills. Facebook suspends app that permitted peephole. http://news.cnet.com/8301-10784_3-9977762-7.html, 2008.

[201] Tehila Minkus, Kelvin Liu, and Keith W. Ross. Children seen but not heard: When parents compromise children's online privacy. In *Proceedings of the 24th International Conference on World Wide Web*, pages 776–786. ACM, 2015.

[202] Jelena Mirkovic, Sven Dietrich, David Dittrich, and Peter Reiher. *Internet Denial of Service: Attack and Defense Mechanisms (Radia Perlman Computer Networking and Security)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[203] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[204] Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E. Tarjan. Clustering social networks. In *Proceedings of the 5th international conference on Algorithms and models for the web-graph*, WAW'07, pages 56–67, Berlin, Heidelberg, 2007. Springer-Verlag.

[205] Alan Mislove, Ansley Post, Peter Druschel, and Krishna P. Gummadi. Ostra: leveraging trust to thwart unwanted communication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 15–30, Berkeley, CA, USA, 2008. USENIX Association.

[206] Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. Measuring the mixing time of social graphs. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 383–389, New York, NY, USA, 2010. ACM.

[207] Mainack Mondal, Bimal Viswanath, Allen Clement, Peter Druschel, Krishna P. Gummadi, Alan Mislove, and Ansley Post. Defending against large-scale crawls in online social networks. In *Proceedings of the 8th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT'12)*, Nice, France, December 2012.

[208] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. Dirty jobs: the role of freelance labor in web service abuse. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 14–14, Berkeley, CA, USA, 2011. USENIX Association.

[209] M.Smith, C. Welty, and D. McGuinness. Web ontology language (owl) guide. http://www.w3.org/TR/owl-guide/, 2004.

[210] Makoto Nakatsuji, Yu Miyoshi, and Yoshihiro Otsuka. Innovation detection based on user-interest ontology of blog community. In *Proceedings of the 5th International Semantic Web Conference*, pages 515–528, 2006.

[211] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187, 2009.

[212] Atif Nazir, Saqib Raza, Chen-Nee Chuah, and Burkhard Schipper. Ghostbusting facebook: detecting and characterizing phantom profiles in online social gaming applications. In *Proceedings of the 3rd conference on Online social networks*, WOSN'10, pages 1–1, Berkeley, CA, USA, 2010. USENIX Association.

[213] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[214] Mark EJ Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[215] Dai Nishioka, Yuko Murayama, and Yasuhiro Fujihara. Producing a questionnaire for a user survey on anshin with information security for users without technical knowledge. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 454–463. IEEE, 2012.

[216] H. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004.

[217] H. Nissenbaum. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford Law Books, 2010.

[218] H. Nissenbaum. A contextual approach to privacy online. *Daedalus*, 140(4):32–48, 2011.

[219] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 83–92, New York, NY, USA, 2006. ACM.

[220] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. Use of temporal expressions in web search. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 580–584, Berlin, Heidelberg, 2008. Springer-Verlag.

[221] Shintaro Okazaki and Javier Alonso. Right messages for the right site: on-line creative strategies by japanese multinational corporations. *Journal of Marketing Communications*, 9(4):221–239, 2003.

[222] Katrina Panovich, Rob Miller, and David Karger. Tie strength in question & answer on social network sites. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1057–1066, New York, NY, USA, 2012. ACM.

[223] Hoon Park. Determinants of corruption: A cross-national analysis. *Multinational Business Review*, 11(2):29–48, 2003.

[224] Jaram Park, Young Min Baek, and Meeyoung Cha. Cross-cultural comparison of nonverbal cues in emoticons on twitter: Evidence from big data analysis. *Journal of Communication*, 64(2):333–354, 2014.

[225] Thomas Paul, Martin Stopczynski, Daniel Puscher, Melanie Volkamer, and Thorsten Strufe. C4ps - helping facebookers manage their privacy settings. In *Social Informatics*, pages 188–201, 2012.

[226] Dan Pelleg, Elad Yom-Tov, and Yoelle Maarek. Can you believe an anonymous contributor? on truthfulness in yahoo! answers. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 411–420. IEEE, 2012.

[227] David J Phillips. Defending the boundaries: Identifying and countering threats in a usenet newsgroup. *The information society*, 12(1):39–62, 1996.

[228] Ansley Post, Vijit Shah, and Alan Mislove. Bazaar: strengthening user reputations in online marketplaces. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 14–14, Berkeley, CA, USA, 2011. USENIX Association.

[229] Matthew Prince, Benjamin Dahl, Lee Holloway, Arthur Keller, and Eric Langheinrich. Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot. In *Second Conference on Email and Anti-Spam*, 2005.

[230] Emil Protalinski. 56% of employers check applicants' facebook, linkedin, twitters. http://www.zdnet.com/article/56-of-employers-check-applicants-facebook-linkedin-twitter/, January 2012.

[231] Krishna P. N. Puttaswamy and Ben Y. Zhao. Preserving privacy in location-based mobile social applications. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, pages 1–6, 2010.

[232] Mingcheng Qu, Guang Qiu, Xiaofei He, Cheng Zhang, Hao Wu, Jiajun Bu, and Chun Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1229–1230, New York, NY, USA, 2009. ACM.

[233] D. Quercia and S. Hailes. Sybil attacks against mobile users: Friends and foes to the rescue. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5, 2010.

[234] Daniele Quercia. Don't worry, be happy: The geography of happiness on facebook. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 316–325, 2013.

[235] L. Rainie, A. Lenhart, and A. Smith. The tone of life on social networking sites. http://www.pewinternet.org/2012/02/09/the-tone-of-life-on-social-networking-sites/, 2012.

[236] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Detecting and tracking political abuse in social media. In *Proc. of ICWSM*, 2011.

[237] Katherine Sarah Raynes-Goldie. *Privacy in the Age of Facebook: Discourse, Architecture, Consequences*. Curtin University., 2012.

[238] Priscilla M. Regan. *Legislating Privacy: Technology, Social Values, and Public Policy*. University of North Carolina Press, 1995.

[239] Katharina Reinecke, Minh Khoa Nguyen, Abraham Bernstein, Michael Näf, and Krzysztof Z Gajos. Doodle around the world: Online scheduling behavior reflects cultural differences in time perception and group decision-making. In *Proceedings of the conference on Computer supported cooperative work*, pages 45–54, 2013.

[240] T. Reynaert, W. De Groef, D. Devriese, L. Desmet, and F. Piessens. Pesap: A privacy enhanced social application platform. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 827–833, 2012.

[241] Matthew Richardson and Ryen W. White. Supporting synchronous social q&a throughout the question lifecycle. In *Proceedings of the 20th International Conference on World Wide Web*, pages 755–764. ACM, 2011.

[242] Duncan Riley. Stat gaming services come to youtube. http://www.bbc.co.uk/news/technology-18813237, Aug 2007.

[243] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, 1995.

[244] J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[245] Chirag Shah and Jefferey Pomerantz. Evaluating and predicting answer quality in community qa. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 411–418, New York, NY, USA, 2010. ACM.

[246] A. Shakimov, H. Lim, R. Caceres, L.P. Cox, K. Li, Dongtao Liu, and A. Varshavsky. Vis- a-vis: Privacy-preserving online social networking via virtual individual servers. In *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, pages 1–10, 2011.

[247] Amre Shakimov and Landon P. Cox. Mutt: A watchdog for osn applications. In *Proceedings of the First ACM SIGOPS Conference on Timely Results in Operating Systems*, pages 6:1–6:14. ACM, 2013.

[248] M. Shehab, G. Cheek, H. Touati, A.C. Squicciarini, and Pau-Chen Cheng. User centric policy management in online social networks. In *IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*, pages 9 –13, 2010.

[249] Anna Shtok, Gideon Dror, Yoelle Maarek, and Idan Szpektor. Learning from the past: Answering new questions with past answers. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 759–768, New York, NY, USA, 2012. ACM.

[250] Thiago H Silva, Pedro OS Vaz de Melo, Jussara M Almeida, Mirco Musolesi, and Antonio AF Loureiro. You are what you eat (and drink): Identifying cultural boundaries by analyzing food and drink habits in foursquare. In *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, 2014.

[251] Andrew Simpson. On the need for user-defined fine-grained access control policies for social networking applications. In *Proceedings of the workshop on Security in Opportunistic and SOCial networks*, SOSOC '08, pages 1:1–1:8, New York, NY, USA, 2008. ACM.

[252] Kapil Singh, Sumeer Bhola, and Wenke Lee. xbook: redesigning privacy control in social networking platforms. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 249–266, Berkeley, CA, USA, 2009. USENIX Association.

[253] Aaron Smith. 6 new facts about facebook. http://www.pewresearch.org/fact-tank/2014/02/03/6-new-facts-about-facebook/, February 2014.

[254] John J Sosik and Dong I Jung. Work-group characteristics and performance in collectivistic and individualistic cultures. *The Journal of social psychology*, 142(1):5–23, 2002.

[255] L. Spitzner. The honeynet project: trapping the hackers. *Security Privacy, IEEE*, 1(2):15–23, 2003.

[256] Lance Spitzner. *Honeypots tracking hackers*. Addison-Wesley, 1 edition, 2002.

[257] Anna Squicciarini, Federica Paci, and Smitha Sundareswaran. Prima: an effective privacy protection mechanism for social networks. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 320–323, 2010.

[258] Anna C Squicciarini, Federica Paci, and Smitha Sundareswaran. Prima: a comprehensive approach to privacy protection in social network sites. *annals of telecommunications-annales des télécommunications*, 69(1-2):21–36, 2014.

[259] Jessica Staddon, David Huffaker, Larkin Brown, and Aaron Sedley. Are privacy concerns a turn-off?: Engagement and privacy in social networks. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS '12, pages 10:1–10:13, New York, NY, USA, 2012. ACM.

[260] E. Steel and G. A. Fowler. Facebook in privacy breach. http://online.wsj.com/article/SB10001424052702304772804575558484075236968.html, 2010.

[261] Tao Stein, Erdong Chen, and Karan Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, SNS '11, pages 8:1–8:8, New York, NY, USA, 2011. ACM.

[262] Tziporah Stern and Nanda Kumar. Improving privacy settings control in online social networks with a wheel interface. *Journal of the Association for Information Science and Technology*, 65(3):524–538, 2014.

[263] Katherine Strater and Heather Richter Lipford. Strategies and struggles with privacy in an online social networking community. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction*, pages 111–119, 2008.

[264] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM.

[265] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. Follow the green: Growth and dynamics in twitter follower markets. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 163–176, New York, NY, USA, 2013. ACM.

[266] Thorsten Strufe. Profile popularity in a business-oriented online social network. In *Proceedings of the 3rd Workshop on Social Network Systems*, SNS '10, pages 2:1–2:6, New York, NY, USA, 2010. ACM.

[267] Fred Stutzman and Jacob Kramer-Duffield. Friends only: examining a privacy-enhancing behavior in facebook. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1553–1562, 2010.

[268] John R Suler and Wende L Phillips. The bad boys of cyberspace: Deviant behavior in a multimedia chat community. *CyberPsychology & Behavior*, 1(3):275–294, 1998.

[269] Ke Sun, Yunbo Cao, Xinying Song, Young-In Song, Xiaolong Wang, and Chin-Yew Lin. Learning to recommend questions based on user ratings. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 751–758, New York, NY, USA, 2009. ACM.

[270] Latanya Sweeney. Uniqueness of simple demographics in the us population. *Carnegie Mellon University Laboratory for International Data Privacy*, 2000.

[271] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[272] Idan Szpektor, Yoelle Maarek, and Dan Pelleg. When relevance is not enough: Promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1249–1260, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[273] Yuta Takano, Satoshi Ohta, Tatsuro Takahashi, Ryosuke Ando, and Takeru Inoue. Mindyourprivacy: Design and implementation of a visualization system for third-party web tracking. In *12th International Conference on Privacy, Security and Trust*, pages 48–56. IEEE, 2014.

[274] Alessandra Toninelli, Animesh Pathak, Amir Seyedi, Roberto Speicys Cardoso, and Valérie Issarny. Middleware support for mobile social ecosystems. In *COMPSAC Workshops*, pages 293–298, 2010.

[275] Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *In Proceedings of the 6th Symposium on Networked System Design and Implementation (NSDI*, 2009.

[276] Harry C Triandis, P Carnevale, M Gelfand, Christopher Robert, Arzu Wasti, T Probst, E Kashima, T Dragonas, D Chan, XP Chen, et al. Culture, personality and deception: A multilevel approach. *International Journal of Cross-Cultural Management*, 1:73–90, 2001.

[277] Alexia Tsotsis. Quora grew more than 3x across all metrics in the past year. http://tcrn.ch/1Oidkem, 2013.

[278] P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. *SIGCOMM Comput. Commun. Rev.*, 18(4):35–42, August 1988.

[279] Zeynep Tufekci. Can you see me now? audience and disclosure regulation in online social network sites. *Bulletin of Science, Technology & Society*, 28(1):20–36, 2008.

[280] Twitter. Twitter usage/company facts. https://about.twitter.com/company, September 2015.

[281] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[282] Blase E Ur and Vinod Ganapathy. Evaluating attack amplification in online social networks. In *Proceedings of the 2009 Web 2.0 Security and Privacy Workshop*. Citeseer, 2009.

[283] Maja van der Velden and Margaret Machniak. Colourful privacy: designing visible privacy settings with teenage hospital patients. In *Proceedings of the 6th International Conference on Information, Process, and Knowledge Management*, 2014.

[284] B. Viswanath, M. Mondal, A. Clement, P. Druschel, K.P. Gummadi, A. Mislove, and A. Post. Exploring the design space of social network-based sybil defenses. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–8, 2012.

[285] Bimal Viswanath, M Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *Proceedings of the 23rd USENIX Security Symposium*, 2014.

[286] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[287] Bimal Viswanath, Mainack Mondal, Krishna P. Gummadi, Alan Mislove, and Ansley Post. Canal: scaling social network-based sybil tolerance schemes. In *Proceedings of the 7th ACM european conference on Computer Systems*, EuroSys '12, pages 309–322, New York, NY, USA, 2012. ACM.

[288] Bimal Viswanath, Ansley Post, Krishna P. Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. In *Proceedings of the ACM SIGCOMM 2010 conference*, SIGCOMM '10, pages 363–374, New York, NY, USA, 2010. ACM.

[289] Gang Wang, Konark Gill, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Wisdom in the social crowd: An analysis of quora. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1341–1352, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[290] Gang Wang, Manish Mohanlal, Christo Wilson, Miriam Metzger Xiao Wang, Haitao Zheng, and Ben Y. Zhao. Social turing tests: Crowdsourcing sybil detection. In *In Proceedings of The 20th Annual Network & Distributed System Security Symposium (NDSS)*, 2013.

[291] Yang Wang, Gregory Norice, and Lorrie Faith Cranor. Who is concerned about what? a study of american, chinese and indian users privacy concerns on social network sites. In *4th International Trust Conference*, pages 146–153. Springer, 2011.

[292] Steve Webb, James Caverlee, and Calton Pu. Social honeypots: Making friends with a spammer near you. In *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS 2008), Mountain View, CA*, 2008.

[293] Ingmar Weber, Antti Ukkonen, and Aris Gionis. Answers, not links: Extracting tips from yahoo! answers to address how-to web queries. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 613–622, New York, NY, USA, 2012. ACM.

[294] Jackie M Wellen and Matthew Neale. Deviance, self-typicality, and group cohesion the corrosive effects of the bad apples on the barrel. *Small Group Research*, 37(2):165–186, 2006.

[295] Christo Wilson, Alessandra Sala, Joseph Bonneau, Robert Zablit, and Ben Y. Zhao. Don't tread on me: moderating access to osn data with spikestrip. In *Proceedings of the 3rd conference on Online social networks*, WOSN'10, pages 5–5, Berkeley, CA, USA, 2010. USENIX Association.

[296] Xintao Wu, Xiaowei Ying, Kun Liu, and Lei Chen. *A survey of privacy-preservation of graphs and social networks*, pages 421–453. Springer, 2010.

[297] Qian Xiao, Rui Chen, and Kian-Lee Tan. Differentially private network data release via structural inference. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 911–920. ACM, 2014.

[298] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 229–240. ACM, 2006.

[299] Ling Xu, Satayapiwat Chainan, Hiroyuki Takizawa, and Hiroaki Kobayashi. Resisting sybil attack by social network and network clustering. In *Proceedings of the 2010 10th IEEE/IPSJ International Symposium on Applications and the Internet*, SAINT '10, pages 15–21, Washington, DC, USA, 2010. IEEE Computer Society.

[300] Wei Xu, Fangfang Zhang, and Sencun Zhu. Toward worm detection in online social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 11–20, New York, NY, USA, 2010. ACM.

[301] Jilong Xue, Zhi Yang, Xiaoyong Yang, Xiao Wang, Lijiang Chen, and Yafei Dai. Votetrust: Leveraging friend invitation graph to defend against social network sybils. PP(99), 2015.

[302] Chao Yang, Jialong Zhang, and Guofei Gu. A taste of tweets: Reverse engineering twitter spammers. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 86–95. ACM, 2014.

[303] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. Uncovering social network sybils in the wild. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 259–268, New York, NY, USA, 2011. ACM.

[304] Alyson L. Young and Anabel Quan-Haase. Information revelation and internet privacy concerns on social network sites: A case study of facebook. In *Proceedings of the Fourth International Conference on Communities and Technologies*, pages 265–274, New York, NY, USA, 2009. ACM.

[305] Haifeng Yu. Sybil defenses via social networks: A tutorial and survey. *SIGACT News*, 42(3):80–101, October 2011.

[306] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: a near-optimal social network defense against sybil attacks. *IEEE/ACM Trans. Netw.*, 18(3):885–898, June 2010.

[307] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 267–278, New York, NY, USA, 2006. ACM.

[308] R. Zafarani and H. Liu. Stanford large dataset collection. http://socialcomputing.asu.edu, 2009.

[309] Eva Zangerle and Günther Specht. Sorry, i was hacked: a classification of compromised twitter accounts. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 587–593. ACM, 2014.

[310] Chi Zhang, Jinyuan Sun, Xiaoyan Zhu, and Yuguang Fang. Privacy and security for online social networks: challenges and opportunities. *Network, IEEE*, 24(4):13–18, 2010.

[311] Elena Zheleva and Lise Getoor. Preserving the privacy of sensitive relationships in graph data. In *Privacy, security, and trust in KDD*, pages 153–171. Springer, 2008.

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

[313] Elena Zheleva and Lise Getoor. *Privacy in social networks: A survey*, pages 277–306. Springer, 2011.

[314] Bin Zhou and Jian Pei. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge and Information Systems*, 28(1):47–77, 2011.

[315] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10(2):12–22, December 2008.

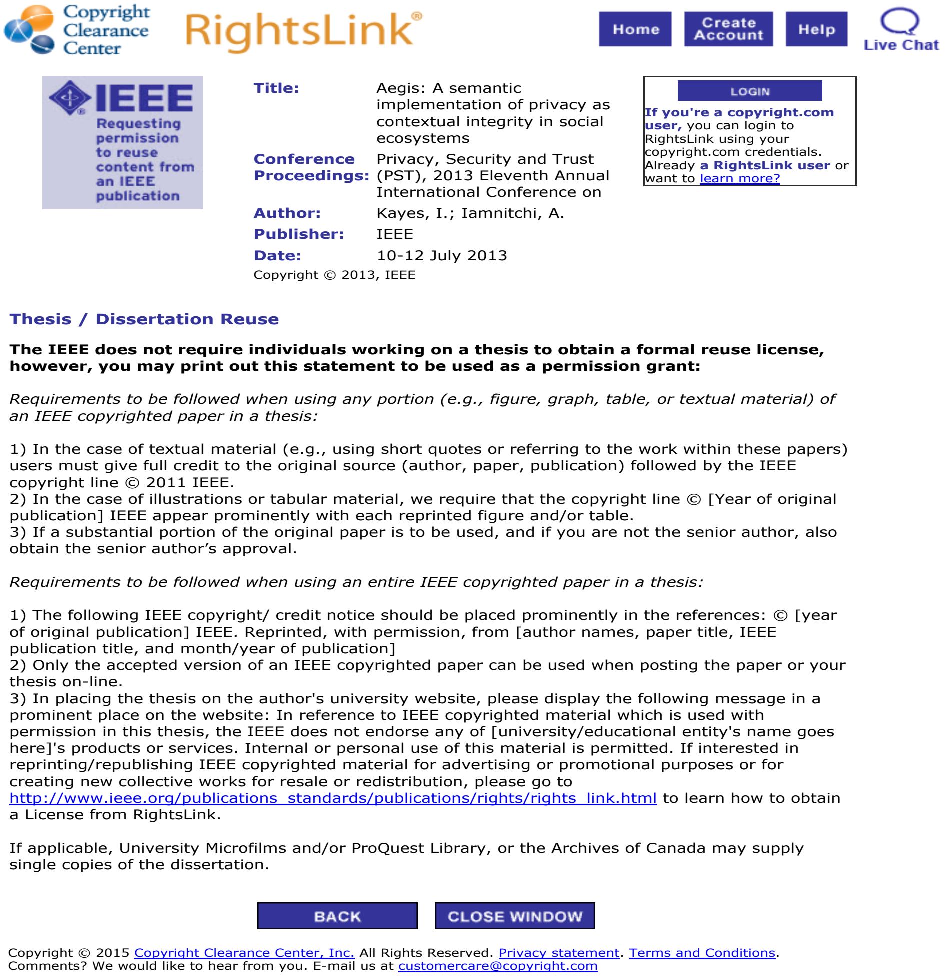
[316] Zhichao Zhu and Guohong Cao. Applaus: A privacy-preserving location proof updating system for location-based services. In *INFOCOM, 2011 Proceedings IEEE*, pages 1889–1897. IEEE, 2011.

[317] Donald W Zimmerman. Teachers corner: A note on interpretation of the paired-samples t test. *Journal of Educational and Behavioral Statistics*, 22(3):349–360, 1997.

[318] Aaron Zinman and Judith Donath. Is britney spears spam. In *Fourth Conference on Email and Anti-Spam, Mountain View, CA*, 2007.

APPENDIX A: COPYRIGHT PERMISSIONS

Below is permission for the use of materials in Chapter 7.


Copyright Clearance Center
RightsLink®
Home
Create Account
Help
Live Chat
IEEE
Requesting permission to reuse content from an IEEE publication
LOGIN

Title:	Aegis: A semantic implementation of privacy as contextual integrity in social ecosystems
Conference Proceedings:	Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on
Author:	Kayes, I.; Iamnitchi, A.
Publisher:	IEEE
Date:	10-12 July 2013

Copyright © 2013, IEEE

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

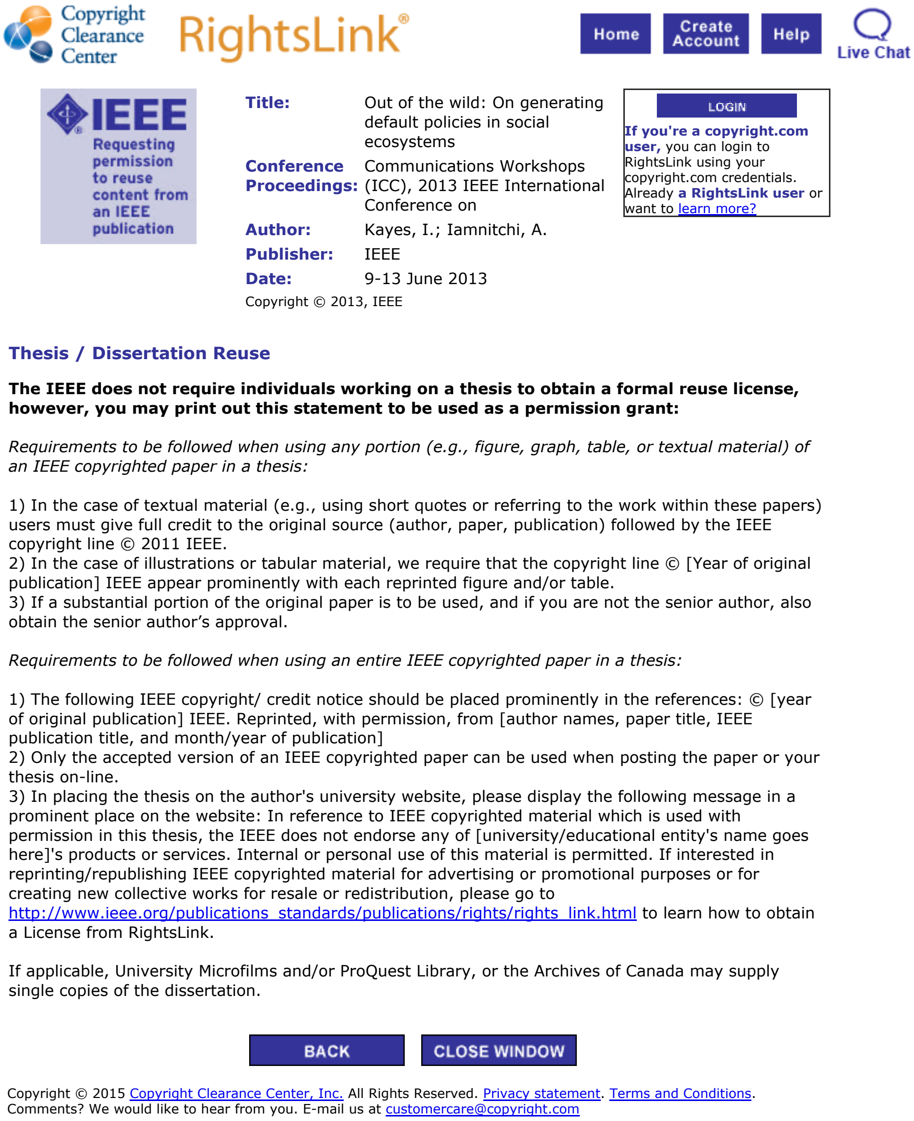
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Below is permission for the use of materials in Chapter 7.


Copyright Clearance Center
RightsLink®
Home
Create Account
Help
Live Chat
IEEE
Requesting permission to reuse content from an IEEE publication
LOGIN
BACK
CLOSE WINDOW

Title:	Out of the wild: On generating default policies in social ecosystems
Conference Proceedings:	Communications Workshops (ICC), 2013 IEEE International Conference on
Author:	Kayes, I.; Iamnitchi, A.
Publisher:	IEEE
Date:	9-13 June 2013

Copyright © 2013, IEEE

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

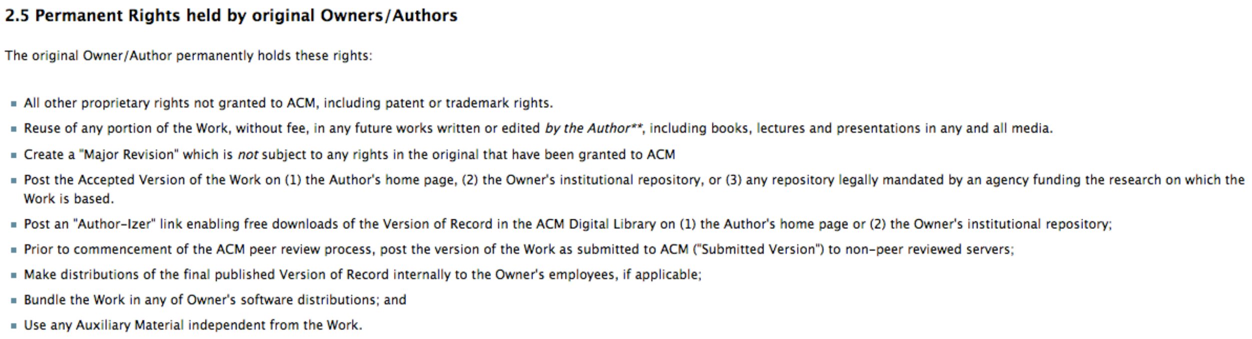
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Below is permission for the use of materials in Chapter 4, 5, 6.

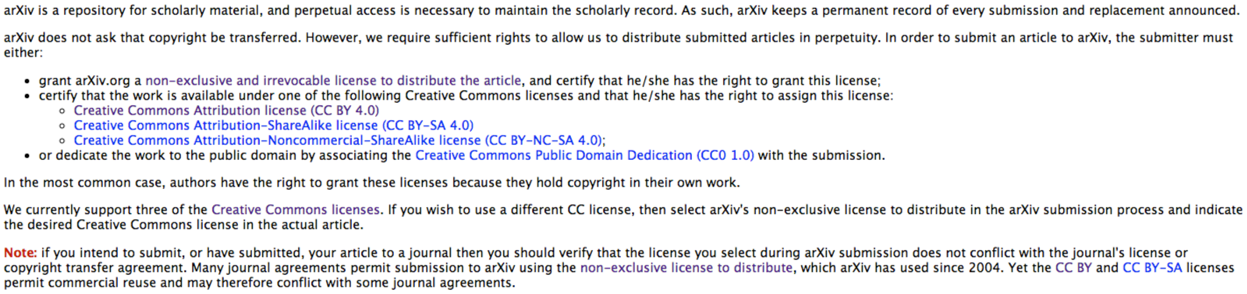


2.5 Permanent Rights held by original Owners/Authors

The original Owner/Author permanently holds these rights:

- All other proprietary rights not granted to ACM, including patent or trademark rights.
- Reuse of any portion of the Work, without fee, in any future works written or edited *by the Author***, including books, lectures and presentations in any and all media.
- Create a "Major Revision" which is *not* subject to any rights in the original that have been granted to ACM
- Post the Accepted Version of the Work on (1) the Author's home page, (2) the Owner's institutional repository, or (3) any repository legally mandated by an agency funding the research on which the Work is based.
- Post an "Author-Izer" link enabling free downloads of the Version of Record in the ACM Digital Library on (1) the Author's home page or (2) the Owner's institutional repository;
- Prior to commencement of the ACM peer review process, post the version of the Work as submitted to ACM ("Submitted Version") to non-peer reviewed servers;
- Make distributions of the final published Version of Record internally to the Owner's employees, if applicable;
- Bundle the Work in any of Owner's software distributions; and
- Use any Auxiliary Material independent from the Work.

Below is permission for the use of materials in Chapter 8.



arXiv is a repository for scholarly material, and perpetual access is necessary to maintain the scholarly record. As such, arXiv keeps a permanent record of every submission and replacement announced.

arXiv does not ask that copyright be transferred. However, we require sufficient rights to allow us to distribute submitted articles in perpetuity. In order to submit an article to arXiv, the submitter must either:

- grant arXiv.org a non-exclusive and irrevocable license to distribute the article, and certify that he/she has the right to grant this license;
- certify that the work is available under one of the following Creative Commons licenses and that he/she has the right to assign this license:
  - Creative Commons Attribution license (CC BY 4.0)
  - Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0)
  - Creative Commons Attribution-Noncommercial-ShareAlike license (CC BY-NC-SA 4.0);
- or dedicate the work to the public domain by associating the Creative Commons Public Domain Dedication (CC0 1.0) with the submission.

In the most common case, authors have the right to grant these licenses because they hold copyright in their own work.

We currently support three of the Creative Commons licenses. If you wish to use a different CC license, then select arXiv's non-exclusive license to distribute in the arXiv submission process and indicate the desired Creative Commons license in the actual article.

Note: if you intend to submit, or have submitted, your article to a journal then you should verify that the license you select during arXiv submission does not conflict with the journal's license or copyright transfer agreement. Many journal agreements permit submission to arXiv using the non-exclusive license to distribute, which arXiv has used since 2004. Yet the CC BY and CC BY-SA licenses permit commercial reuse and may therefore conflict with some journal agreements.

ABOUT THE AUTHOR

Md Imrul Kayes is a PhD candidate in the department of computer science and engineering at the University of South Florida, USA. He received his MS degree in Computer Science from the University of South Florida in 2014. He got the B.Sc. Engg. degree in Computer Science and Engineering from the Bangladesh University of Engineering and Technology (BUET) in 2009. His research interests broadly lie in the area of networked systems. At a high level, his current work focuses on privacy, abuse, user influence, and retention in large, complex networks such as online social, community Q/A and blogging networks. Before joining the PhD program, he worked more than three years in software development industry. He has worked as software engineers for SoftwarePeople, Delta life Insurance, and Binary solutions. He spent the fall of 2013 as an intern at Yahoo Research Labs, Barcelona, Spain.