fRiendTrust: A Privacy Preserving Reputation System for Online Social Networks

Iraklis Symeonidis, Filipe Beato, and Bart Preneel
ESAT/COSIC - KU Leuven and iMinds
Leuven, Belgium
{first.lastname}@esat.kuleuven.be

Abstract. Online social networks (OSNs) are currently very popular among Internet users, offering tools for sharing and submitting information, such as political opinions, photos and events. If users shares sensitive information with their online friends, those friends are in a position to collect, analyse and redistribute this information, which may result in a privacy issue. This would not be a problem if all friends would be trustworthy, but unfortunately not all friends can be trusted. Moreover, online friendship cannot be represented by a simple binary trust value. We propose a reputation system that measures and quantifies the trustworthiness of online friends based on the interactions in OSNs; our tool helps users to minimize the exposure of shared information. We analyse the sharing tools provided by the OSNs and discuss the privacy issues they introduce. Finally, we describe the implementation of a prototype that under Facebook limitations delivers a lower bound privacy score to each of the user friends.

1 Introduction

In the context of digital environment, Online Social Networks (OSNs) are very popular among Internet users: a 2010 survey reported nearly half of the online users are members of an OSN [49], with Facebook reaches more than 1 billion monthly active users and Twitter 140 million [11]. About 35% of active users spend more time online than face-to-face and 33% of them were more likely to speak to someone online than offline [4].

OSNs have altered the social ecosystem in a remarkable way by providing users’ with a plethora of easy-to-use tools [1], offering highly efficient means for establishing and maintaining social connections. Users are allowed to share information through text, pictures and videos, and build communities around shared interests. According to Facebook, a user with 130 friends and a membership in 80 groups and events submits, on average, 90 pieces of data per month [40]. The vast increase of users, sharing options and tools, leads to the dissemination of large amounts of information in OSNs [29]; users disclose information related to them or to their friends, as they share content. For instance, on Facebook users share personal opinions, pictures and submit their current activities [6]. This information is accessible by a diverse set of viewers, such as online friends,
service providers and third party organizations [16,31,43]. Henceforth, this has prompted serious privacy concerns among the users; as reported by the media and the research community [2,25,37–40,55]. These concerns increased after the US government mass surveillance program (including PRISM) became known to the public in 2013 [48].

Our paper presents a two-dimensional solution to tackle the privacy issues such as the exposure of user’s information through online friends. First, we have implemented a mechanism that collects user’s information via interaction tools and second we propose a reputation system (RepS) that intends to quantify user’s interactions and evaluate the trustworthiness of online friends. We have also developed a proof of concept to demonstrate the applicability of our claims (fRiendTrust).1 The application was developed on Facebook, as it is the most popular OSN. The main functionality of fRiendTrust is to aggregate and quantify users’ interactions (e.g., comments, photos, personal messages) in a centralized way and to assign an evaluation score for each user. The current release of the application only considers the personal messages that are exchanged between users; this will be extended in future work.

The remainder of this paper is organized as follows: Section 2 presents the theoretical background of the notions trust-trustworthiness and reputation systems while Section 3 briefly analyzes the privacy issues in OSNs. Section 4 describes the fRiendTrust application, a reputation system on Facebook and summarizes the limitations of the model. Section 5 overviews Facebook’s API and describes fRiendTrust technical specifications. Section 6 reviews related work and Section 7 draws a conclusion and proposes future topics of research.

2 Background

This section presents the definitions of trust and trustworthiness. Moreover, reputation systems and their components are introduced; we base the discussion on Schiffner [52] and Josang [35]. We analyze the schemes and highlight the most well-known reputations systems (RepSs) and reputation calculation algorithms (RepCAlgs). As most interactions are purely digital, the notions of trust-trustworthiness and RepS can be combined; they allow users to build trust relationships between entities, e.g., person to person or person to item. Finally, the difference between trust and reputation is discussed.

2.1 Trust

Trust is regarded as a central concept of economic, social and political behaviors; its absence thwarts many relations and can cause harm. It has been defined as the “mutual confidence that no party to an exchange will exploit another’s vulnerabilities” [5,51], i.e., the inclination of “a person A to believe that other persons B who are involved with a certain action will cooperate for A’s benefit

1 http://www.friendtrust.info
and will not take advantage of A if an opportunity to do so arises” [9]. Trust is referred to the relationship between two entities, A –the truster– who is willing to trust and B –the trustee– who is trying to gain confidence. As the definition implies, the relationship between two entities that trust each other is called trust path. For instance, “when parties to an exchange trust each other, they share a mutual confidence that others will not exploit any adverse selection, moral hazard, hold-up, or any other vulnerabilities that might exist in a particular exchange”. Trust can be constructed in a direct or indirect way. According to Jøsang et al. [35] direct trust is between two entities A and B who interact and trust each other whereas indirect trust can be constructed implicitly via recommendations.

Trust is an attribute of an established relationship. However, the process of building and maintaining trust requires an additional step for a user; to deduce the establishment of a trust path with the other entity (trustee). Trustworthiness is the complement key point to trust in terms of assisting a user (truster) to get insight in the other entity’s (trustee) behavior. The definition of trustworthiness implies from Sabel’s definition of trust as “an exchange partner is trustworthy when it is worthy of the trust of others” [5,51]. A worthy trust partner is someone who will not exploit the vulnerabilities of the other behaving trustworthy. It is useful to notice that trust is an attribute of a relationship between exchange partners, while trustworthiness is an attribute of individual exchange partners. Both notions complement each other and help the user to make decisions. Combining these two notions we can infer that trustworthiness could be the cause of trust. For instance, if someone is untrustworthy there is a high probability that the truster does not take the risk of trusting the other entity (trustee) whereas, conversely it is likely that a trustworthy entity will establish a trust relationship with a truster. Trust and trustworthiness could be applied as well in a group of entities. Each entity can be analyzed under the same scheme of truster and trustee; a group scheme could be analyzed through smaller mutual relationships.

Trust is deployed under a specific scope with an acceptable value of trustworthiness between entities; this can be translated as qualified and quantified trust [35,52]. Qualified trust is defined as the scope of interaction that is taking place between two entities in order to interact with trust. In particular, two entities trust each other if they have agreed to interact in a certain context and for a specific purpose, e.g., to privately share personal and group photos. Quantified trust is defined as the gradual value of trust; it typically takes more than two values (e.g., trusting someone or not). Mutual trust could exist under the same scope (qualified) and the same amount (quantified).

2.2 Reputation Systems

The purpose of using a reputation systems (RepS) is to help users get informed about the quality and reliability of an entity such as a seller, product or a service. Users decide if they want to interact with someone, or something in order to buy, use or lease an object. RepSs distinguish three major profiles: a person (user)
who gets feedback through previous evaluation scores, a person who submits
rates (rater), and an entity that receives ratings (ratee).

A RepS consists of three components [23, 28, 52]: (1) the rating process
(RatP), (2) the reputation calculation algorithm (RepCAIg) and the (3) query
process (QueP) as depicted in Fig. 1. It collects, measures, stores and refers back
to the users the evaluation scores. Moreover, a reputation database (RepDB) is
used to store the scores.

![Reputation System Diagram](image)

**Fig. 1. Reputation System**

RepSs use various reputation calculation algorithms (RepCAIg), to measure
and represent the trustworthiness of a ratee. Most RepCAIgs have been devel-
oped after solid academic and industrial research [28,35,52]; examples are Google
page rank and eBay RepS. The commonly used calculation schemes are: (1) the
additive reputation calculation algorithm takes into account positive and nega-
tive votes (if used) summing them up, resulting in a scalar value. (2) the weighted
averages reputation calculation algorithm, resulting in a percentage value for the
evaluation score of a ratee based on the average function. A rating could be a
positive vote, boosting the overall score of a rateec.

Alternative RepS schemes have been proposed; some of the are used in real
world applications while others are studied in academia. Vote-to-promote is one
of the top three RepSs according to ENISA [28]. A rater submits a vote with a
Boolean value of one or zero for every interaction. The collection of all the sub-
mitted votes are aggregated under the additive RepCAIg. To estimate a value
of trustworthiness fRiendTrust applies the vote-to-promote RepS by collecting
quantitative characteristics of the personal messages exchanged in order to estimate the trustworthiness as an evaluation score.

The network architecture of a RepS is either centralized or decentralized. In centralized architectures, the ratings are submitted to and the evaluation score is computed by a centralized service. In decentralized architectures there is no central location for submitting the ratings or obtaining the evaluation scores: every reputation component process and displays the ratings individually for each user.

2.3 Difference between Trust and Reputation

RepSs are widely used in online markets and communities, such as eBay, Amazon and Slashdot [28]. Trust corresponds to an individual relation while RepS to public opinion. A good evaluation score of an entity does not automatically lead to a trust relationships. A truster could potentially trust a trustee despite a bad reputation score and the converse also holds. As a result, the RepS focuses on the trustworthiness of an entity and it measures the tendency of an entity to honor a relationship and behave honestly.

3 Model

One of the primary goals of OSNs is to encourage users to disclose personal information via interaction tools, such as personal messages, events and groups. While users enjoy the use of the interaction tools, online friends may not always be trustworthy leading to possible privacy issues, as they can observe, accumulate and redistribute high volumes of shared information. In this section we analyze the privacy risks that OSNs introduce through their interaction tools.

Likes: tend to be simple evaluation mechanisms, quite similar to reputation logic. The study of Kosinski et al. [36] showed that a wide variety of user’s personal attributes, ranging from sexual orientation to intelligence, can be automatically and accurately inferred using their Facebook Likes.

User’s Profile Information: consists of a user’s basic information (e.g., full name, birthday, email, home town) required to become a member of a service. In particular, on Facebook the required profile information consists of the user name, the subscription date and the home address [34]. Even if the required amount of information for a Facebook account is minimal, the total amount of information that a user may submit can be quite large. From a privacy point of view, profile information corresponds to personal (e.g., date of birth, location) and sensitive (e.g., health condition, political opinions) data; this can be harmful if it would be abused. The study of Amanda et al. [47] showed that on Facebook, users who indicated their relationship status as either single or in a relationship disclosed significantly more information than others. Furthermore, users choose on average approximately 25% of possible profile information: gender, relationship status and age, are the profile features that are most disclosed.
**Online friends:** of a user potentially reveal the personality and behavior of a user. The research of Jernigan et al. [32] demonstrates a method for accurately predicting the sexual orientation of Facebook users by analyzing friendship associations. Moreover, Ediger et al. [20,22] showed that a massive analysis of OSN-graphs is feasible via a tool entitled GraphCT; it can process a graph of 2.14 billion edges in approximately 15 minutes [21]. This speed-up process potentially provides the capability of massive user relations analysis.

**Comments and messages:** consist of an enriched informative interaction tool, as users share their opinions with their online friends; implying qualitative and quantitative values of user’s information. With the use of manual or semantic analysis, a malicious stakeholder can deduce information about a user. Carvalho et al. [13] presented a set of possibilities for detecting ironic comments in the online news communities. Moreover, in case of Facebook the operation of deleting personal messages corresponds to hiding rather than erasing. For instance, if users erase a conversation with a friend and send a new message later, they will be surprised to notice that the conversation re-appears, including the older messages.

**Photos and videos:** can contain an important amount of information, that can threaten the privacy of a user. A user, who likes a landscape, a food item, an event or a car, and who shares related multimedia data, is also sharing personal and even sensitive information. For instance, a user’s photo in a hospital can easily lead to the conclusion that the user is sick (sensitive data) or is an employee (personal data). Henceforth, from the privacy point of view, a malicious stakeholder can make use of a user’s multimedia information by profiling a user for various purposes like advertisement.

**Events and groups:** can reveal a user’s behavioral information. The fact that a user attends an event or is a member of a group can be sensitive behavioral information. For instance, a user with preference for particular political events and groups can reveal trends of political preference.

**Applications:** The popularity of third party Apps has prompted several privacy concerns, as they are hosted on external servers and are out of the OSN’s control. In particular, several researchers have demonstrated that some apps collect more information than required [17]. Moreover, they have revealed the existence of malicious applications that deviate from the generic permissions pattern established by Facebook in order to acquire more information about the users [24]. Subsequently, Chaabane et al. demonstrated that apps gain tracking capabilities [16], and can later disseminate the collected information to “fourth party” organizations [15]. Similarly, Huber et al. [31] developed an automated evaluation tool (AppInspect) demonstrating security and privacy weaknesses of a large set of Facebook Apps.
4 fRiendTrust

In this section, we demonstrate our proof of concept implementation that is called fRiendTrust. Based on the interaction tools, it provides an overall evaluation score to each of the online friends through a reputation system (RepS) applied on Facebook. As a first approach, fRiendTrust measures and quantifies the user’s behavior (trustworthiness) based on the exchanged messages among them. Our system is centralized. We now describe the scheme and list the most important parameters.

Reputation System. fRiendTrust is based on the vote–to–promote RepS, in which all the ratings are aggregated directly. The RepS is responsible for collecting the ratings, calculating an overall score and displaying this information to the users. An evaluation score is calculated and assigned to each user every time fRiendTrust is requested.

Reputation Values. As a first approach, our tool concentrates on the messages exchanged between the users. We introduce two classes of metrics, the general behavior metrics (Gbm) and the individual behavior metrics (Ibm). The first class considers the general message characteristics. The second takes into account the actions of each user (see Fig. 2). Both Gbm and Ibm have three different types of metrics: the amount (e.g., message_count, sum_response_time), the average value (e.g., avg_response_time, avg_message_length) and the rate of change (e.g., rateofchange_message_length) of the exchanged messages.

Fig. 2. fRiendTrust domain metrics. (a) Ibm, (b) Gbm

- Individual Behavior Metrics (Ibm): aggregate, measure and store individual features of the messages exchanged by each user. For instance, the overall time that a user is active, or the number of messages (Table 1).

2 http://www.friendtrust.info
– General Behavior Metrics (Gbm): aggregate, measure and store the general characteristics of a messages. For instance, the overall time that a communication interaction is taking place, measuring the time of the first and last submitted message (Table 2).

<table>
<thead>
<tr>
<th>Fields</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thread_id</td>
<td>integer</td>
<td>The id of the message.</td>
</tr>
<tr>
<td>viewer_id</td>
<td>integer</td>
<td>The id of the user participating in a conversation.</td>
</tr>
<tr>
<td>originator</td>
<td>Boolean</td>
<td>Indicates the user that has sent first a message.</td>
</tr>
<tr>
<td>author</td>
<td>Boolean</td>
<td>Indicates if the user that has respond to the message.</td>
</tr>
<tr>
<td>recipient</td>
<td>Boolean</td>
<td>Indicates if the user has the permissions to read a message.</td>
</tr>
<tr>
<td>usr_sum_response_time</td>
<td>integer</td>
<td>The overall time that a user is active in a conversation, taking into account the time of the first and the last sent message.</td>
</tr>
<tr>
<td>usr_avg_response_time</td>
<td>integer</td>
<td>The average time that a user is active in a conversation, taking to account the time of the first and the last sent message, divided by the overall conversation time.</td>
</tr>
<tr>
<td>usr_message_count</td>
<td>integer</td>
<td>The number of messages that a user has sent.</td>
</tr>
<tr>
<td>usr_avg_message_length</td>
<td>integer</td>
<td>The average length of the sent messages, taking into account the sent messages by the user, divided by the overall messages length.</td>
</tr>
<tr>
<td>usr_msg_body_semantanal</td>
<td>integer</td>
<td>The quality of the submitted messages by the specific user.</td>
</tr>
<tr>
<td>usr_rateofchg_message_length</td>
<td>integer</td>
<td>The ratio of message length progression, that a user submits.</td>
</tr>
<tr>
<td>unseen</td>
<td>Boolean</td>
<td>A per-message flag indicating whether the user has seen the new incoming message.</td>
</tr>
<tr>
<td>unread</td>
<td>Boolean</td>
<td>A per-message flag indicating whether the user has read the new incoming message.</td>
</tr>
</tbody>
</table>

**Reputation System Architecture:** forFriendTrust is centralized (Fig. 3) and consists of three components:

– Rating Process (RatP): is responsible for collecting the necessary information, sending calls to the Facebook API. forFriendTrust uses both Facebook Query Language (FQL) and Graph API tools to collect data from the Facebook server; and stores the data in a database for further processing.
Table 2. General Behavior Metrics

<table>
<thead>
<tr>
<th>Fields</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thread_id</td>
<td>integer</td>
<td>The id of the message.</td>
</tr>
<tr>
<td>avg_response_time</td>
<td>integer</td>
<td>The average time that the interaction is taking place, measuring the first and the last message that have been sent.</td>
</tr>
<tr>
<td>sum_conversation_time</td>
<td>integer</td>
<td>The overall conversation time, taking into account the first and the last sent message.</td>
</tr>
<tr>
<td>avg_message_length</td>
<td>integer</td>
<td>The average length of the exchanged messages.</td>
</tr>
<tr>
<td>msg_body_semantanal</td>
<td>integer</td>
<td>The quality of the submitted messages by all users.</td>
</tr>
<tr>
<td>msg_body_lth_rateofchge</td>
<td>integer</td>
<td>The progression of message lengths that users exchange.</td>
</tr>
<tr>
<td>count_recipients</td>
<td>integer</td>
<td>The number of recipients in a message. A number strictly larger than two indicates a group message.</td>
</tr>
<tr>
<td>count_messages</td>
<td>integer</td>
<td>The number of messages that the users have exchanged.</td>
</tr>
<tr>
<td>count_recent_authors</td>
<td>integer</td>
<td>The number of authors. An author is someone who has permission to read but also has posted a message.</td>
</tr>
</tbody>
</table>

Fig. 3. fRiendTrust network architecture

- **Reputation Calculation Algorithm (RepCAlg):** aggregates the collected data from the database and calculates an evaluation score. The calculation is based on the *addition calculation*. The current scheme of fRiendTrust, takes into account both the Ibm and Gbm parameters in order to calculate the
evaluation score. In particular, it uses the following parameters: author, originator from Ibm and the count_recipients, count_messages from the Gbm.

– **Query Process (QueP):** is triggered when a user visits the application, requesting data from the database; the queries are SQL-based.

– **Calculation equation:** Let $\Psi$ be the evaluation_score, $\Gamma$ the author, $\Delta$ the originator, $\Theta$ the count_recipients and $\Lambda$ the count_messages. Also let $z_1$ be the weight of author, $z_2$ be the weight of originator, $z_3$ be the weight of count_recipients, $z_4$ be the weight of count_messages. In this paper we make an assumption related to the weights; there is a high subjectivity for each person with regard to their perception of why someone is trustworthy. We assign the following values to the weights $z_1 = z_3 = z_4 = 1/2$ and $z_2 = 1$. Thus, the calculation of the evaluation_score is represented by the Eqn (1).

$$\Psi = z_1 \cdot \Gamma + z_2 \cdot \Delta + z_3 \cdot \Theta + z_4 \cdot \Lambda \quad (1)$$

Given an example, a user could exchange messages with a friend or a group of friends (Fig. 6):

**Case 1 – Group Message:** a user who has not responded to a message ($\Gamma = 0$), is not the originator of the message ($\Delta = 0$), the recipients are a group of 13 friends ($\Theta = 13$) and the number of exchanged messages is 41 ($\Lambda = 41$), thus the evaluation_score $\Psi \approx 26.9$.

**Case 2 – Between two friends:** a user who has responded to a message ($\Gamma = 1$), is the originator of the message ($\Delta = 1$), the recipients are the 2 friends ($\Theta = 2$) and the number of exchanged messages is 11 ($\Lambda = 11$), thus the evaluation_score $\Psi \approx 7.9$.

### 4.1 Limitations

In this paper we propose fRiendTrust a RepS that dynamically measures the quantitative characteristics available in OSNs. However, we can identify the following limitations in our study:

– **Equation Weights:** need to be validated with social studies such as online surveys, experimental sessions, of trust games. Trust Games [12,30,33,52,57] is a widely used experimental design and measuring tool for interpersonal trust [10] in social science and economics [26]. Previous work of Ben-Ne et al. [9] showed that there exist no metric by which the evaluation of the trust and trustworthiness measures can be objectively defined, studying a sample of 200 university students.

– **Reputation scores:** are calculated for every online friend independently and can only be accessed by the user. This secures the model from privacy leaks eliminating the scenario of users having access to other (users) reputation results. However, it limits the measurement results to the particular user and only.
Centralized reputation architecture: should be trusted by the application users as it places trust in the aggregation server. fRiendTrust collects the necessary amount of information for the reputation calculation algorithm on an external server and thus this server should be trusted by the users of the application.

5 Implementation

In this section we describe the tools provided by Facebook to developers. Moreover, we describe the technical specifications of fRiendTrust.

Facebook Application Programming Interface (API): is a development tool, providing access to Facebook data. The API offers two ways of retrieving content, the Graph API and the Facebook Query Language (FQL): (1) The Graph API contains vertices (users) and edges (relationships) between vertices [54]. Graph API is a HTTP tool based on GET command (Fig. 4 (A)), forwarded through the URL [19]. (2) FQL: is a SQL-style interface based on the Social Graph, offering extended features which are not available on the Graph API, such as embedded queries (Fig. 4 (B)) [19].
The network architecture of a Facebook application is centralized, as illustrated in Fig. 5. The protocol of the Facebook API consists of three groups of messages. First, when a user accesses an application, the user is authenticated based on messages exchanged between the Application and Facebook (Fig. 5 (B.a)). Second, after the acceptance of access, the user requests data from the Application Server (Fig. 5 (B.b)). The Application Server replies with the requested data, or sends a request to the Facebook Server in order to gather the necessary information. Facebook returns the requested data to the application Server (Fig. 5 (B.c)) through the API (e.g., Graph API and FQL). A Facebook application consists of two main areas (iframes); the main part which indicates that you are browsing the Facebook website (Fig. 5 (A.i)) and the application area, where the application content is displayed (Fig. 5 (A.ii)).

![Fig. 5. A. Content structure and B. Facebook applications network architecture](image)

**fRiendTrust interface:** relies on the friend list information page of Facebook. Although it doesn’t have the same structure and design, it has the main parameters of the Facebook friend list page. For instance, as depicted in Fig. 6, the items that are displayed in the Facebook friend list are name, photo, the number of friends and the condition of the friendship related to the user. The interface of fRiendTrust displays a list of friends that a user has together with their names, photos and the evaluation score for each user (Fig. 6).

**Technical specifications:** fRiendTrust was developed with the PHP v.5 programming language [50]. The requests are processed and served by an Apache server [3] and the evaluation scores-votes are stored in a MySQL database [46] (Fig. 7). The interactions among the application and the fRiendTrust server are formulated with the SQL programming language [53]. The application is installed and operated under the CentOS v.6 operating system in a Cloud as a platform virtual machine [14].
6 Related Work

During the last few years a substantial amount of research has been performed to identify and minimize the exposure of users’ information in OSNs [1]. Most of this work, focused either on the analysis of particular behavioral patterns [32] or on the protection of submitted content [7, 41, 42].

FSEO [6], FaceCloak [42], FaceVPSN [18] and NOYB [27] are privacy schemes providing obfuscation techniques for OSNs. Scramble! [6, 7] and flyByNight [41] are privacy solutions that use access control and symmetric key cryptography. Minkus et al. [45] propose a metric approach for the level of the privacy state.
of a user, determined by the privacy settings. PrivAware, is a tool aiming to detect and quantify the amount of information revealed in OSNs focused on social circles [8].

The solutions are complementary to our work as they don’t take into account the dynamics of users in OSNs [40, 44, 56]. Our research goes beyond these models, aiming to quantify the amount of information revealed in OSNs via the interaction tools and it proposes a reputation system (RepS), that measures and evaluates the trustworthiness of a user.

7 Conclusion

Online social networks (OSNs) have acquired enormous popularity and have become an essential part of many people’s daily lives. Along with the convenient ways of socializing OSNs can become a threat to users’ privacy and dignity. This paper proposes a new approach for the design of a RepS that preserves privacy in OSNs and particularly on Facebook. In particular, our contribution delivers a solution that estimates the trustworthiness of a user by measuring his behavior. In order to demonstrate the practicality of our approach, a proof of concept was developed entitled fRiendTrust; this tool offers an automatic mechanism that gives each user a perspective on the estimated trustworthiness of their online friends by assigning an evaluation score to each of them. As a first approach, our research focused on personal messages, as those contain a lot of quantitative information. Our implementation represents a first step; our model could be expanded to other interaction tools of Facebook such as events, comments, likes, applications and extending fRiendTrust to other OSNs.

References


17. Pern Hui Chia, Yusuke Yamamoto, and N Asokan. Is this app safe? a large scale study on application permissions and risk signals. In *WWW*, Lyon, France, 04/2012 2012. ACM.


45. Tehila Minkus and Nasir Memon. On a scale from 1 to 10, how private are you? scoring facebook privacy settings. 2014.