Measuring Privacy Risk in Online Social Networks

Justin Becker Hao Chen
jlbecker@ucdavis.edu, hchen@cs.ucdavis.edu
Computer Science Department, University of California, Davis

Abstract

Measuring privacy risk in online social networks is a challenging task. One of the fundamental difficulties is quantifying the amount of information revealed unintentionally. We present PrivAware, a tool to detect and report unintended information loss in online social networks. Our goal is to provide a rudimentary framework to identify privacy risk and provide solutions to reduce information loss. The first instance of the software is focused on information loss attributed to social circles. In subsequent releases we intend to incorporate additional capabilities to capture ancillary threat models. From our initial results, we quantify the privacy risk attributed to friend relationships in Facebook. We show that for each user in our study a majority of their personal attributes can be derived from social contacts. Moreover, we present results denoting the number of friends contributing to a correctly inferred attribute. We also provide similar results for different demographics of users. The intent of PrivAware is to not only report information loss but to recommend user actions to mitigate privacy risk. The actions provide users with the steps necessary to improve their overall privacy measurement. One obvious, but not ideal, solution is to remove risky friends. Another approach is to group risky friends and apply access controls to the group to limit visibility. In summary, our goal is to provide a unique tool to quantify information loss and provide features to reduce privacy risk.

1. Introduction

Online social networks have become highly popular in the past few years. As of this writing, Facebook has more than 200 million active users, and more than 100 million users log on to Facebook at least once each day [1]. Online social networks provide platforms for their users to publicize their personal information.

A basic principal in computer security is to prevent information from escaping its intended privacy boundaries. Information extending beyond defined partitions is commonly referred to as information leakage. In most environments measuring the amount of information lost is a difficult task. Moreover, associating lost information with a particular threat can be even more challenging. We aim to quantify the information revealed unintentionally in online social networks and provide solutions to reduce privacy risk.

The importance of quantifying privacy in online social networks is even more critical given the scale of the networks. Protecting the massive amount of corresponding personal information is a critical task. Recent examples [2], [3], suggest current mechanisms provide inadequate levels of protection. Moreover, a recent article [4] suggests users are unwilling to risk losing control of their personal information. In order to evaluate the privacy risks associated with social networks we first need a means to identify and quantify the different threats.

Our research is focused on quantifying privacy risks in online social networks and providing solutions to mitigate those risks. To help quantify privacy threats we introduce PrivAware, a tool to measure privacy risk in Facebook. PrivAware is designed to execute within a user's profile to provide reporting and a set of recommended actions to alleviate privacy threats. This current release quantifies the privacy risk attributed to friend relationships in Facebook. Additionally, the release provides simple solutions to reduce the privacy risk associated with this threat.

A total of 93 participants chose to install and execute PrivAware. On average, we were able to derive 59.5% of the personal attributes associated with our participants. For all demographics, men, women, married, and not married, we were able to derive the personal attributes associated with the users more than 50% of the time. In addition to the measurements we also supplied user-actions to our participants to help mitigate their risk. One action is to remove the risky friend relationships that lead to information loss. Another, more subtle approach, is to seperate the offending friends into groups and apply access control mechanisms to each group. For example, in Facebook, a user can apply access controls to limit the functionality associated with a particular group of users. PrivAware recommends *grouping* risky users rather than deleting them. To provide the lists of precarious friends we applied a series of heuristic approaches to limit the overall privacy risk while maximizing the number of friend relationships. The Heuristics were able to provide significant improvements over a baseline approach, randomly deleting users until the desired level of privacy is met. On average, the number of friends necessary to remove or group, using our common-friends heuristic, was 19 less than the baseline.

We intend to expand the current capabilities of PrivAware to include further mechanisms to measure and report additional threat models. The objective of these initial results is to call attention to the need to quantify privacy risk in social networks and encourage further research.

The remainder of this paper is structured as follows. Section 2 is the related works section. Section 3 provides the design details for PrivAware. Section 4 describes the experiment. Section 5 and 6 are future work and conclusion, respectively.

2. Related Work

The scale of online social networks coupled with increase scrutiny has pressured network operators (Facebook, MySpace, etc.) to provide increased levels of user privacy. Despite those efforts many privacy threats still exist. Specifically, recent studies have uncovered some challenging privacy concerns [5], [6], [7]. However, few efforts provide tangible feedback to the end user. Our research is concerned with identifying and measuring privacy threats to provide reporting so end users can make decisions depending on their desired level of privacy risk.

Currently, PrivAware is limited to measuring the privacy risk attributed to direct social contacts. Our research is not the first to explore this phenomenon. Analyzing social contacts to infer user attributes has been tested in previous research [8]. In this work, the authors leverage different statistical learning techniques to correctly infer the value for a particular attribute type. For example, in their study, the authors were able to correctly infer the attribute values for *political view* and *gender* in Facebook, on average, greater than 50% of the time. We view these results as complimentary to our work and provide additional data points to strength our position.

Similar work has also been conducted in the field of inference detection [9]. In this instance, researchers developed a semi-automated tool to infer private information from redacted data sources. Inference detection is accomplished using term frequency analysis coupled with online search results. From the tool, the authors were able to recover a large portion of the private information from redacted documents. The authors also introduced novel formalisms to quantifying the information revealed unintentionally. This work influenced our decision to pursue measuring privacy risk in online social networks. In [9] the researchers experienced difficulties in validating their results. Few data sets are available that contain both the redacted document and the original. In the domain of social networks we are not limited in the same way. We are able to verify our inference results directly using the target nodes original profile information.

Our research benefits from these works and many others. However, we are of the opinion that our work provides novel contributions to the field. First, we consider a wider array of attributes. We do not limit our research to a set of structured data types. For example, we derive inferences based on semi-structured values such as employer and educational institution. Consequently, we are forced to address challenging problems such as data disambiguation [10] and named entity recognition [11]. Second, we present solutions to help mitigate the results from our findings. We consider different algorithms to reduce information loss, explore their corresponding runtime complexities, and suggest actions to users to reduce their privacy risk. Finally, and most important, we suggest a framework encouraging future research to extend our work to include additional implementations for various threat models.

3. Design

To quantify and reduce privacy risks attributed to friends in online social networks, we propose PrivAware. PrivAware provides two functions: First, it infers the attributes of a user based on those of his friends. Second, it suggests how to change the members of the user's friends to reduce the number of inferrable attributes to an acceptable level.

3.1. Inference detection

We formally define the inference problem as follows. Let User t be the inference target. Let F be the set of direct friends of t. The inference problem is: given all the attributes of all the users in F, infer the attributes of t.

3.1.1. Inference calculation.

This problem represents several concrete problems in real social networks. For example, a privacy-conscious user t sets some of their private attributes to be only group-accessible. A non-group member may still be able to infer the values of the group-accessible attributes based on the values of t's friends'. As another example, a privacy-paranoid user has installed a social network application. Since the application can access all his attributes, they purposely leaves certain attributes blank. However, since the application may access the attributes of the privacy-paranoid user's friends, it may be able to infer the omitted attributes of the user.

Based on the observation that social circles tend to leak information we infer a user's attributes based on those of their friends. Some may recall the old adage, birds of a feather flock together. We use a simple, intuitive algorithm to demonstrate the power of this technique and to serve as a baseline for comparison. For each attribute, the algorithm selects the most popular value of this attribute among the user's friends. If the number of friends who share this value exceeds a threshold, the algorithm assign this value as the inferred attribute of the user. Currently, PrivAware derives inferences for the following attributes: age, country, state, zip, high school name, high school grad year, university, degree, employer, affiliation, relationship status, and political view.

3.1.2. Disambiguation.

The simple algorithm described above faces a challenge: how to determine if two attribute values are *equal*, when the same conceptual value may be represented differently? For example, "UC Berkeley", "Berkeley", and "Cal" are all variants of the University of California, Berkeley. This is commonly referred to as the data disambiguation problem, usually related to semi-structured and unstructured text. To mitigate this problem, we employ a number of approaches. First, we developed a dictionary of common variations for universities, political parties, degrees, and employers. PrivAware uses the dictionaries as a lookup table to transform values into their canonical forms. For example, PrivAware transforms the attribute value "Doctor of Philosophy" to "Ph.D", and "Cal" to "UC Berkeley". Second, we process each value using the Levenshtein Algorithm [12]. This approach unifies terms containing simply misspellings and punctuation differences.

3.1.3. Verification.

To evaluate the power of our simple inference algorithm, we use the following metrics:

- *Inferred attributes*: the attributes that PrivAware can infer. PrivAware cannot infer an attribute when the number of friends sharing the most common same attribute value falls below a threshold.
- *Verifiable inferences*: the attributes that PrivAware can infer and that are also present in the target user's profile. This way, we may verify the correctness of the inferred attributes.
- *Correct inferences*: the attributes that PrivAware can infer and that match the value in the target user's profile.

3.2. Inference reduction

Once PrivAware shows that certain private or absent attributes of a user may be inferred, the user may wish to change his set of friends to avoid the inference. To defeat the inference algorithm of PrivAware, the user could adopt two strategies. First, the user could remove friends. When the number of friends who share the same attribute value falls under a threshold, PrivAware's inference algorithm fails. In realty, the user has two choices. He can simply remove the risky friends as indicated by PrivAware (to be discussed later). Alternatively, if his social network platform supports access control on groups (such as Facebook), he can partition his friends into safe and unsafe groups, set his unsafe groups to be invisible, and move all his risky friends as indicated by PrivAware into the unsafe group (i.e., hiding these friends from the public view). The latter approach is often more desirable, because users are unlikely willing to remove friends, especially those with similar attribute values. Instead of deleting or hiding friends, the user could also pollute his network of friends by adding fake friends. For instance, if the user has added enough fake friends such that the most common value of an attribute are among fake friends, our inference algorithm would output this attribute, which mismatches the user's true attribute. However, this approach has drawbacks. First, the user might be unwilling to add fake friends, as this might confuse his real friends and distort his social networks. Second, the fake friends might be unwilling to accept the add requests, which might prevent the user from adding these fake friends on certain social network platforms. We will only explore the first approach, i.e., identifying risky friends to remove or hide.

3.2.1. Problem definition.

Formally, we define our problem as follows. We represent the friends associated with user t as a set of tuples $(f_v, \{(type, value, weight)\})$. The term f_v represents the value assigned to the friend by user t. This term is useful because it allows us to optimize the results in favor of friends who have higher social value. For

Type	Value	Weight
Age	27	1
Employer	Google	1
University	Stanford University	.8
Relationship Status	Single	1

Table 1. Example attributes of a friend

example, a user might assign a higher social value to friends who are family members than friends who are colleges. The next value in the tuple is a set corresponding to the attributes associated with the friend. We represent each attribute in the set with a triple. The term *type* is the attribute type associated with the value, e.g., university, age, zip code. Similarly, the term *value*, in the triple, corresponds to the actual value for the particular attribute, e.g., UC Berkeley, 25, 95812. The final term *weight* is a value in the range from zero to one assigned based on the confidence of the disambiguation process. For example, the disambiguation service might assign a score of 0.8 for the attribute value Cal and a score of 1 for UC Berkeley. Our inference algorithm uses this weight to compute the (weighted) frequency of attribute values. For example, if the university attribute is Cal for one friend and UC Berkeley for another friend, they contribute $0.8 + 1.0 = 1.8$ to the frequency count of the canonical attribute value UC Berkeley. Table 1 shows the attributes of an example friend.

Given the above description of friends, our goal is to reduce the number of inferred attributes by removing or hiding friends. Apparently, the more friends we remove, the fewer number of attributes we may infer. But on the other hand, the user wishes to keep as many friends as possible under a privacy requirement. Therefore, we define the inference reduction problem as follows:

> Given a privacy requirement, represented as the maximum allowed number of inferrable attributes, and a set of friends, find the subset of friends that maximize the total values of friends and that satisfy the privacy requirement.

3.2.2. Heuristic solutions. We proceed with heuristic-based approaches to reduce privacy risk. We also consider a more specific instance of the problem. We equalize the value of all friends. we also assume perfect disambiguation, so the weight for each attribute value is constant. By making these two changes, we are able to provide tractable solutions to minimizing privacy risk.

Removing random friends. . This algorithm is straightforward and serves as a baseline for comparison. It keeps removing friends randomly until it satisfies the privacy requirement (the maximum allowed number of inferred attributes).

Removing friends with most attributes. Randomly removing friends fails to consider the difference between friends in their contributions to inference. Intuitively, we wish to remove friends who contribute the most to the inference. Our first heuristic approach is, during each iteration, remove the friend with the largest number of visible attributes. This is based on the intuition that friends with a larger number of attributes contributes a greater amount to inference detection.

Removing friends with most common friends. Our second heuristic algorithm considers the number of common friends between the target user and each of his friends. During each iteration, the algorithm removes the friend that shares the most common friends with the target user. The intuition is that people who share more friends also share more common attributes.

3.2.3. Design considerations. PrivAware was designed to execute in both OpenSocial networks and Facebook. However, we selected Facebook to gather our initial results for two central reasons. The first and most obvious is the amount of available data. 200 million users are currently registered with Facebook. This provides increase opportunity for PrivAware to spread virally (not in a malicious manner) through the network and gives us the opportunity to leverage the massive amount of user generated content. The other major advantage is a subtle difference in policy between Facebook and OpenSocial. With OpenSocial, a third-party application can only query a user's friend data if both parties (user and friend) have consented and installed the application [13]. Said in another way, if user A installs and executes PrivAware in an OpenSocial network, the application can only query friend information for user A's friends who have also installed PrivAware. In contrast, Facebook does not impose this restriction. If user A executes PrivAware in Facebook it may query the data associated with user A's friends. This difference allows us to collect and examine the friend information for a user who executes PrivAware.

4. Experiments

4.1. Data collection

We initially developed PrivAware to measure the privacy risk for a known privacy threat concerning third

Question	Options	Results
How familiar are you with Facebook's privacy policy concerning third party applications?	Not Familiar Slightly Familiar Very Familiar	33.3% 51.4% 15.3%
How would you score the privacy settings associated with your profile information?	1, 2, 3, 4, 5	3.057
Have you used any of the privacy mechanisms provided by Facebook?	Yes, No	0%, 100%
Given your grade, how would you score the privacy settings associated with your profile information?	1, 2, 3, 4, 5	1.047
Will you change any of your privacy settings?	Yes, No	64.7%, 35.3%

Table 2. Reactions to privacy threat

party applications [14]. The goal of PrivAware was to determine whether users would take action if presented with an unfavorable privacy measurement. To solicit participants we placed an advertisement on Facebook with the following text.

> **Privacy concerns?** PrivAware is a tool to measure the profile information accessible to applications. Determine how much information you're revealing.

If a user clicked on the advertisement they were directed to the homepage for PrivAware which contained the following description.

> PrivAware is a simple Facebook application designed to score privacy settings conerning third-party applications. It will query your profile to determine what information is available to Facebook applications. Based on the amount of available information PrivAware will assign a corresponding grade. In addition to the score, users will be prompted with a series of questions concerning privacy. After completing the survey, users are encouraged to reconfigure their privacy settings and recompute their privacy score. Questions will not be prompted to the user on subsequent visits.

We received 105 individuals willing to participate in the study. We asked each participant in our study to answer a series of questions before and after their privacy score was revealed. The intent was to capture the participants sentiment when exposed to their privacy risk. The results of the survey are listed in Table 2. Results are given in percentages with the exception of user-privacy scores, those are given as averages.

To compute the privacy risk we simply divided the total number of attributes visible to third party applications by the total number of attributes per participant. The computed result is the percentage of attributes revealed to third party applications. For simplicity when presented to the end user, we translated the percentages to a letter grade ranging from *A* to *F*. An *F* score corresponds to a large number of attributes being revealed and an *A* score represents very few attributes being revealed. The distribution of scores for the 105 participants is the following, 64% scored an *F*, 36% scored a *D*, and no participants scored an *A*, *B*, or *C*. These results suggest the privacy risk attributed to the threat of a malicious third party application is high.

To answer our initial question, would users take action to mitigate a high level of privacy risk, we examined the survey results in the context of the privacy scores. Before a participant was issued a grade they were asked three questions. The first question was to determine their familiarity with Facebook's privacy policy concerning third party applications. From the results, 51.4% felt they were ”slightly familiar” with the policy. This suggests our participants had some idea of what to expect in terms of privacy risk. Next, the participants were asked how they would score their privacy settings, on a scale of 1 to 5. 1 representing no privacy and 5 representing complete privacy. The average score was 3.057. This implies users would expect half of their information to be accessible to third party applications, given their current privacy settings. Finally, participants were asked if they have used any of the privacy mechanism provided by Facebook. Surprisingly, 100% responded they haven't used any mechanisms. Based on these results we can characterize our participants as moderate to weak privacy advocates. Once the participant had answered the first series of questions they were issued their privacy risk score. To reiterate, 64% scored an *F*. Subsequently, the participants were presented with two final questions. The first question asked again for the privacy score, only this time the user has been issued a grade. As expected, the results where much lower, the average score was 1.047. The second question asked if the participant would take action to change their privacy settings. From the results, 64.7% responded they would change their settings.

Among the 105 individuals who took our survye, 93 agreed to participate in our inference detection and reduction research. The demographics of the latter participants follows. The average age was 23.89 with

	All Users	Men	Women	Married	Not Married	< 25	≥ 25
Total people	93	47	24	24	40	25	24
Total friends contacts	12,523	6,201	5,133	2,394	6,153	5,049	2,750
Average friends	134	131	183	99	153	201	114

Table 3. Total and average number of friends

	All Users	Men	Women	Married	Not Married	< 25	≥ 25
Total people	93	47	24	24	40	25	24
Total social contacts	12,523	6,201	5,133	2,394	6,153	5,049	2,750
Average social contacts	134	131	183	99	153	201	114
Total attributes inferred	1,673	933	508	472	726	515	436
Total verifiable inferences	918	508	280	265	402	283	250
Total attributes correctly inferred	546	329	157	141	238	182	131
Percent correctly inferred	59.5	64.8	56.1	53.2	59.2	64.3	52.4

Table 4. Total inferred attributes

Attribute	Correct inference
Affiliation	63.1
Age	72.3
Country	96.0
Degree	38.9
Employer	18.8
High School Name	74.1
High School Grad Year	82.7
Political View	36.7
Relationship Status	69.4
State	87.0
University	51.0
Zip	20.0

Table 5. Correct inferences by attribute

a standard deviation of 6.1 and a range of 14-44. The pool included 47 men and 24 women. The group included 12 different hometown countries: Canada, China, Ecuador, Egypt, Iran, Malaysia, New Zealand, Pakistan, Singapore, South Africa, United Kingdom, and United States. Some participants chose not to state their age, gender, or origin. To derive our privacy scores we examined a total of 12,523 direct friend relationships. Table 3 shows the total and average number of friends.

4.2. Inference detection

To evaluate the effectiveness of our inference algorithm, we use the metrics defined in section 3.1.3 : *Inferred attributes*, *Verifiable inferences*, and *Correct inferences*. Table 5 shows the percentage of correct inferences (over all verifiable inferences) by each attribute. It shows that structured attributes — such as age, country, state, high school grad year — tend to be correctly inferred a higher percentage of the time. The one exception is zip code. Conversely, semi-structure and unstructured attributes tend to be more difficult to infer correctly. With improved data disambiguation, we conjecture the percentage of correct inferences would increase. Our expectation is based on a sampling of data from our results. Analyzing the data manually, we found many instances where terms were in fact *equal* but were not identified as such by our data disambiguation techniques.

Table 4 enumerates the number of inferred attributes, verifiable inferences, and correct inferences. Additionally, we include results for different demographics to identify trends in the data.

Table 6 shows the number of contributors for derived inferences. We define a contributor as a friend who provides at least one contribution in the collection of derived values. For example, if our inference algorithm infers the value Stanford University for university, all friends with Stanford University listed in their profiles will have contributed to that inference. Table 7 provides the average number of contributors per inference, verifiable inferences, and correct inferences.

The results above suggest, for the participants, that Facebook is at best providing privacy less than fifty percent of the time when faced with the threat of attribute frequency count.

4.3. Inference reduction

We ran the three algorithms for removing friends in Section 3.2.2. First, we executed the algorithm that removes random friends over all participants setting the desired level of privacy to zero, representing complete privacy. The average number of friends necessary to remove to meet this level was 145. Upon first consideration, this result seems unlikely given the average number of friends was 134. However, analyzing the

	All Users	Men	Women	Married	Not Married	< 25	≥ 25
Total contributors for inferences	11,972	5,951	4,924	2,266	5,867	4,948	2,647
Total contributors for verifiable inferences	11,805	5,908	4,847	2,213	5,799	4,923	2,619
Total contributors for correct inferences	11,324	5,775	4,642	2,115	5,581	4,797	2,508

Table 6. Total contributors to inferences

	All Users	Men	Women	Married	Not Married	< 25	≥ 25
Average contributors per inference	7	6	9	4	8	9	6
Average contributors per verifiable inferences	12	11	17	8	14	17	10
Average contributors per correct inference	20	17	29	15	23	26	19

Table 7. Average contributors to inferences

	Random	Visible attr.	Common friends
<50 Friends	25	18	14
<100 Friends	40	31	26
<200 Friends	78	69	54
<500 Friends	112	101	97
All	145	134	111

Table 8. Friends necessary to remove

data reveals a bias introduced by participants with a large number of friends. For example, participants with a number of friends greater than five hundred required a much greater number of friends be removed than those participants with fewer friends. To reduce this bias, we partitioned the set of data into groups of participants with increasing numbers of friends. The sets considered in the results are divided into groups with 50, 100, 200, and 500 friends. We also included an all-participants result for completeness. Table 8 enumerates the results.

Table 8 also shows the results from the other two algorithms, i.e., removing the friends with the most attributes, and removing the friends with the most common friends. The results show that the last algorithm (removing the friends with the most common friends) works the best (resulting in removing the fewest number of friends), while the first algorithm (removing random friends) is the worst. In fact, on average, by removing commons friends, the difference in the number of friends necessary to remove or group, in contrast to the random approach, was 19 over all five groups.

Both of these algorithms provide improvements over randomly removing friends to limit privacy loss. However, neither approach takes into consideration order. As mentioned previously the optimal solution, one that considers each permutation, would take into consideration how order effects the privacy score. For example, removing a friend earlier in the process might result in a higher level of privacy than removing them later in the process. Moreover, none of these approaches attempts to the solve the more general problem which considers social value and imperfect disambiguation results.

5. Future Work

Currently, PrivAware quantifies privacy risk for a single threat model in Facebook. In future releases, our intent is to incorporate many different threats models to capture a more complete assessment of the privacy risk in online social networks. We also intend to implement a version of PrivAware that executes in OpenSocial networks. With multiple variants running in different networks we with have the capability to compare and contrast the privacy risk associated with each network. Then, users will have the information to gauge which social network provides the adequate level of privacy for their risk tolerance. Additionally, we are interested in quantifying the risk associated with common user actions in online social networks. For example, measuring the risk associated with friends tagging pictures online or users cross commenting.

Improving the algorithms used to measure privacy risk is also a major area of research. The approaches presented here only serve as a baseline to highlight the risk concerning friend relationships in Facebook. Our aim is to provide more complex algorithms to better quantify the level of privacy for multiple threat models. We are encouraged by research in natural language processing and data mining to further our development of PrivAware. Specifically, data cluster techniques and more advanced named entity recognition algorithms may provides improved privacy measurements.

Another important aspect of our research is to encourage a common framework to measure privacy risk in online applications. An implicit goal of our research is to provide an example of deriving privacy in online social networks. However, this work is not limited to social networks, but has applications in other domains. For example, measuring the privacy risk associated with online email or micro-blogging applications. In the future we intent to expand our research into these areas and provide a common measurement of privacy across these different domains.

6. Conclusion

Measuring privacy risk in online social networks is an important task. Millions of users are contributing large amounts of information to their social graphs. Information exposed unintentionally can have serious consequences. To complicate matters, many users are unfamiliar with the underlying privacy risks associated with social networks. Common user actions such as adding a friend can increase the level of information revealed unintentionally.

PrivAware aims to quantifies the amount of information revealed in online social networks and provide means to reduce those risks. In this current release, we measure the information loss associated with friend relationship in Facebook. From our results, we were able to correctly infer, 59.5% of the time, the attributes of a user based on their social contacts. We also provide results for different demographics of users suggesting attributes can be inferred with a probability greater than 50% of the time. In addition to reporting privacy risk, we were able to supply user actions to participant to help mitigate their privacy risk. On average, the number of friends necessary to remove or group for complete privacy, using our common-friends heuristic, was 19 less than the baseline. These results are encouraging and provide the catalyst for future research. Our long term goal is to provide a system that measures multiple threat models and provides users with the guidance to reduce those privacy risks.

References

[1] (2009) Facebook statictics. [Online]. Available: http://www.facebook.com/press/info.php?statistics

[2] B. M. Rubin. Social-networking sites viewed by admissions officers. [Online]. Available: http://archives.chicagotribune.com/2008/sep/20/local/chi-facebook-college-20-sep20

[3] W. Du. Job candidates getting tripped up by facebook. [Online]. Available: http://www.msnbc.msn.com/id/20202935/

[4] (2009, February) Facebook backs down, reverses on user information policy. [Online]. Available: http://www.cnn.com/2009/TECH/02/18/facebook.reversal/index.html

[5] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu, "Link privacy in social networks," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2008, pp. 289–298.

[6] M. M. Lucas and N. Borisov, "Flybynight: mitigating the privacy risks of social networking," in *WPES '08: Proceedings of the 7th ACM workshop on Privacy in the electronic society*. New York, NY, USA: ACM, 2008, pp. 1–8.

[7] A. Simpson, "On the need for user-defined fine-grained access control policies for social networking applications," in *SOSOC '08: Proceedings of the workshop on Security in Opportunistic and SOCial networks*. New York, NY, USA: ACM, 2008, pp. 1–8.

[8] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," University of Maryland, College Park, Tech. Rep. CS-TR-4926, 2008, an earlier version appears as CS-TR-4922, July 2008.

[9] J. Staddon, P. Golle, and B. Zimny, "Web-based inference detection," in *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2007, pp. 1–16.

[10] H. T. Nguyen and T. H. Cao, "Named entity disambiguation: A hybrid statistical and rule-based incremental approach," in *ASWC '08: Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 420–433.

[11] A. Mikheev, M. Moens, and C. Grover, "Named entity recognition without gazetteers," in *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1999, pp. 1–8.

[12] Z. Su, B.-R. Ahn, K.-Y. Eom, M.-K. Kang, J.-P. Kim, and M.-K. Kim, "Plagiarism detection using the levenshtein distance and smith-waterman algorithm," in *ICICIC '08: Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control*. Washington, DC, USA: IEEE Computer Society, 2008, p. 569.

[13] "Opensocial api developer's guide," March 2009. [Online]. Available: http://code.google.com/apis/opensocial/docs/0.8/devguide.html

[14] A. Felt and D. Evans, "Privacy protection for social networking apis," in *Web 2.0 Security and Privacy 2008*, May 2008. [Online]. Available: http://www.cs.virginia.edu/felt/privacybyproxy.pdf