Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Sh@re: Negotiated Audit in Social Networks

Alejandro Gutierrez, Apeksha Godiyal, Matt Stockton, Michael LeMay,
Carl A. Gunter and Roy H. Campbell
Department of Computer Science
University of Illinois at Urbana Champaign
Emails: {agutier3, godiyal2, mstockto, mdlemay2, cgunter, rhc}@illinois.edu

***Abstract*—With the growth in the popularity of social networking sites like Facebook and MySpace, there is an increasing concern about privacy of content posted by users. Many users enter personal details about themselves but have poor understanding of theats such as identity theft and stalking. There is a need to educate and assist users in understanding how their personal data is exposed to other users. In this paper, we introduce the concept of *negotiated audit* which gives users of social networks valuable feedback about how their data is being used. Our design has three levels of auditing for both sharing and browsing data: *no audit*, *complete audit* and *anonymous audit*. Users can classify their data as requiring some level of auditing and can also set their browsing preference to one of the auditing levels. Users can only see some data if their browsing preference is compatible with the data's audit level thus giving rise to negotiation of how much users are willing to reveal about their activities and how much data they will be able to access. We provide a mathematical model and describe a simple social networking prototype called *Sh@re* that implements negotiated audit.**

***Index Terms*—Privacy, Social Networks, Negotiated Audit.**

I. Introduction

Online social networks have rapidly gained popularity in recent years. Sites such as MySpace, Facebook and Friendster have become established forums for keeping in contact with old acquaintances, meeting new ones and sharing different types of media. For example, the web traffic data for Facebook shows 51.6 million unique US visitors a month [12]. MySpace ranks eighth in overall web traffic, with over 68.4 million unique US visitors each month [13].

While these web sites are useful tools for exchanging information, there has been growing concern over breaches in privacy caused by poor understanding by users of the potential consequences of adding sensitive content. Many users are finding how the information they shared with their friends can easily find its way into the hands of strangers, authorities, and the public at large. The personal data of users is being circulated far more widely than they would like.

Many social network users enter a significant amount of personal details about themselves such as: where they went to school, their favorite movie titles as well as their relationship status. To control the use and dissemination of this information, users can configure the privacy setting options offered by the web sites. These privacy configurations determine who sees what information about the user. However, this method has at least three limitations. First, there is little privacy awareness among users [6], [4] due to limited feedback informing them by whom and how their data is being viewed or used. Moreover, users are not aware of the possible consequences of providing personally identifiable information. Second, more often than not, the privacy settings offered by the websites are too granular and overly complex. The more complicated and granular the privacy settings are, the less likely the user is going to configure anything. It is difficult and time-consuming to fine tune the privacy settings such that users can maximize their social benefits of sharing different types of media with trusted friends, and minimize their privacy risks of leaking data to untrusted parties. Third, the least obvious privacy violations in social networks can be found in how the information flows through weak ties, defined as the links among people that are not closely bonded [5]. Researchers have shown how loosely connected individuals are interconnected on social networks through relatively few random ties [14]. Users are often not aware of how their data is accessed by weak ties. Thus a significant threat to privacy in online social networks arises when users have no feedback about how their data is being viewed or used by others. Design of online social networks can therefore benefit from making users aware of the usage pattern of their data, enabling them to fully understand the privacy implications of their privacy settings and spotting potential privacy violations.

We propose a way to partially address these concers by offering easily configurable privacy settings with feedback that helps users understand the implications of their privacy settings and tune them to better suit their needs. Our system *audits* the access to content and reports this access to users. We present three levels of auditing to choose from for both sharing and browsing data: *No Audit*, *Anonymous Audit*, and *Complete Audit*. Users can access a particular datum only if their browsing preference is compatible with its audit level. This represents a kind of negotiation between users regarding how much they are willing to disclose about their activities and how much data they will be able to access. We call this strategy *negotiated audit* and the aim of this paper is to describe one approach to it both mathematically and with a prototype implementation. The discussion is organized as follows. Section II overviews some previous work on online social networks. Section III discusses the concept of negotiated audit. Section IV describes a mathematical foundation for our approach and Section V describes the format of the generated audit reports. Section VI has details of the *Sh@re* prototype implementation. Sections VII and VIII discuss possible extensions and conclusions respectively.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

II. Previous Work

There has been huge interest revolving around threats to privacy posed by social networking sites. The issue has been explored not only in academia but has also garnered attention in online articles, blogs and newspaper columns. However, to the best of our knowledge, there has been no prior work on incorporating a privacy feedback feature to address the problem of educating users about the implications of their privacy settings.

D. Hiltz et al. [3] relate how in a comparison with MySpace, Facebook users were more trusting of their site and its members, and they were more willing to include identifying data in their profile. However, the MySpace users were more active in the forming of fresh relationships than Facebook users. H. Jones et al. [9] examined how the Facebook design is seriously flawed and how the design affects privacy of Facebook users. R. Gross et al. [6] studied the patterns of information revelation in online social networks and their privacy implications. They evaluated the amount of information users disclose and highlighted the potential attacks on various aspects of their privacy. T. Govani et al. [4] studied privacy issues awareness amongst students and the available privacy protection provided by Facebook. They found that most students are aware of the possible consequences of providing personally identifiable information, but nevertheless feel comfortable providing it and did not take the initiative to protect their information.

Articles [10] and [7] discuss at length the privacy risks that could result from not using encryption for social information. They explain some of the fears that may cause the behavior of sharing less information on social networks. [8] shows how context rich content can lead to successful phishing attacks.

III. Negotiated Audit

In a social networking site with negotiated audit, users are allowed to post personal information, pictures, videos, comments and other resources like any other social networking site. In addition, users have the option of monitoring their data by adjusting a degree of auditing. Enabling auditing on any resource implies that the user who uploaded the resource will be able to see details of accesses to it. Three auditing levels are available for any data:
(1) *No Audit*: No report will be generated when manipulating the resource.
(2) *Anonymous Audit*: The details of the user accessing the resource are anonymized in the audit report, but the access to the resource is logged. This audit level does not enable the owner to uniquely identify the user making the access, though the owner will know the relationship it has with that particular user.
(3) *Complete Audit*: This audit report enables the owner to uniquely identify the user who accessed the resource.

Users can configure the audit level they wish to select for each of the resources they upload/own. They can also set their browsing preference to access only *Non Audited*, *Anonymously Audited* or *Complete Audited* resources. Users can set this preference for every user they are related to. A user will only be able to see the data that is compatible with their browsing preferences and by doing so we ensure that an audit entry about a user's access is not generated without their knowledge. For example a user with *Non Audited* browsing preference will not be presented with any *Anonymously Audited* or *Complete Audited* data.

Our approach ensures privacy preserving interactions among users. Users will be able to see all the data only if they are willing to reveal everything about their activities. For example, 'User A' has her data as *Complete Audit*. 'User B' whose browsing preference is *Non Audited* wants to access all the data belonging to 'User A'. Then 'User B' will not be able to do so without changing his browsing preference to *Anonymous Audit* or *Complete Audit*. If users are not willing to reveal anything about their activities, they will be able to access only data that is marked as *Non Audited* by other users in the social network.

Our system presents novel degree of negotiations between users by not only preserving their privacy but also by providing them useful feedback. It also presents a right balance between how much a user is willing to reveal about their activities and how much feedback the other users are seeking.

IV. Mathematical Model of Negotiated Audit

The proposed auditing and access control scheme can be represented with an *Attribute Based Access Control (ABAC)* [15] model. The model involves several attributes which can be assigned to principals and objects in the system. Additionally, the model involves several policy rules. One policy rule determines whether or not to grant access to an object to a principal on the social network. Several other policy rules determine when attributes can be updated on principals and objects. *Obligations* determine if any necessary auditing actions should be taken once a principal has accessed an object. This section defines the key attributes, the key policy rules, obligations and actions necessary to implement the privacy auditing model.

A. Definitions for Attribute Based Access Control Model

- *Principals(P)*: Set P is the set of all users in the social networking application.
- *Objects(O)*: Set O is the set of all pieces of information created by principals in the social networking application.
- *Attributes(A)*: Set A is the set of all attributes that are assignable to principals and objects on the social network. Combined with functions, these determine the overall access and auditing policies of the social network.

B. Assignable Attributes

The following are the assignable attributes for the active items of the social network described above.

1) Principal-Assignable Attributes:

- *Name(p)*: This attribute is simply the name of the principal $p \in P$ on the social networking site. This attribute is set upon the initial creation by the principal, and *cannot* be modified.

- *AllowableAuditLevelForAccessing(p)*: This attribute represents the maximum allowable audit level for the principal $p \in P$ when the principal accesses objects on the social network. This means that for any object of the system, if the audit level of the object is greater than the audit level of the user, the system should deny the user access to that object.
- *DefaultAuditLevelForCreation(p)*: This attribute represents the default audit level for all objects that are created by the principal $p \in P$ on the social network site.

2) Object-Assignable Attributes:

- *Owner(o)*: This attribute represents which principal is the owner of the object $o \in O$ on the social networking site. This attribute is automatically assigned when the object is created and attributed to the principal that created the object and *cannot* be modified.
- *ConfiguredAuditLevel(o)*: This attribute represents the auditing level that is assigned to a particular object $o \in O$. This level determines what type of auditing will be done upon non-owner access to the object on the social network.

C. Auditing Levels

These attributes are assignable to different active items on the social network. However, their valid values are the same. The following are the valid values.

1) No Audit:

- *ConfiguredAuditLevel*: The value of this attribute specifies that an object $o \in O$ can be accessed by any principal $p \in P$ on the social network with no resulting audit information. The object access is anonymous in that the owning principal (e.g. Owner(o)) will have no evidence that other principals in P have accessed the object o.
- *AllowableAuditLevelForAccessing*: The value of this attribute specifies that the principal is willing to access objects in the social networking site only if the object's ConfiguredAuditLevel is equal to *No Audit*.
- *DefaultAuditLevelForCreation*: The value of this attribute value specifies that any object created by the principal should have a ConfiguredAuditLevel of *No Audit*. As we will describe later in the paper a function allows the user to override this default.

2) Anonymous Audit:

- *ConfiguredAuditLevel*: The value of this attribute specifies that when an access occurs to object $o \in O$, an anonymous audit log entry will be generated. The log entry will be available for reading only by the principal owner of the object. Anonymous log entries do not allow the owner of the object to uniquely identify the principal making the access, but will allow them to limit the subset of possible principals that accessed their data. For information on the contents of these types of log entries, see Section V.B - Anonymous Audit.
- *AllowableAuditLevelForAccessing*: The value of this attribute specifies that the principal is willing to access objects in the social networking site only if the objects ConfiguredAuditLevel is equal to or less restrictive than *'Anonymous Audit'*. The comparative restriction is defined at the end of this section.
- *DefaultAuditLevelForCreation*: The value of this attribute specifies that any object created by the principal should have a ConfiguredAuditLevel of *'Anonymous Audit'*.

3) Complete Audit:

- *ConfiguredAuditLevel*: The value of this attribute specifies that when an access occurs to object $o \in O$, a completely/uniquely identifiable log entry will be generated. This log entry will be available for reading only by the principal owner of the object. The entry itself will at minimum contain the username, the object in question, and a timestamp of the access.
- *AllowableAuditLevelForAccessing*: The value of this attribute specifies that the principal is willing to access any objects on the social networking site. *'Complete Audit'* is the most restrictive audit level for objects in the social network.
- *DefaultAuditLevelForCreation*: The value of this attribute specifies that any object created by the principal should have a ConfiguredAuditLevel of *'Complete Audit'*.

There is a restriction ordering of auditing levels for objects. *Complete Audit* is more strict $(>)$ than *Anonymous Audit*, and *Anonymous Audit* is more strict $(>)$ than *No Audit*.

D. Policy Rules

The following are the policy rules for the system, along with information about when and how the rules are triggered.

- **CanAccess(p,o)**: This rule defines what object $o \in O$ a principal $p \in P$ can access. This rule is executed any time a principal attempts to access an object in the social network. The rule definition is: $((Owner(o) = Name(p)) \vee (AllowableAuditLevelForAccessing(p) \geq ConfiguredAuditLevel(o)))$.
- **CanChangeObjectAuditLevel(p,o)**: This rule defines when a principal $p \in P$ can change the ConfiguredAuditLevel of an object $o \in O$. This rule is executed any time a principal attempts to set a new ConfiguredAuditLevel on an object in the social network. The rule definition is: $Owner(o) = Name(p)$.

E. Obligations

Obligations are mandatory requirements that a principal has to perform after obtaining rights on an object [11].

- **GenerateCompleteAudit(p,o)**: This defines when the social network should generate a complete audit entry. The obligation is executed any time a principal $p \in P$ actually accesses an object $o \in O$ in the social network. The obligation definition is: $((Owner(o) \neq Name(p)) \wedge (ConfiguredAuditLevel(o) = CompleteAudit))$.
- **GenerateAnonymousAudit(p,o)**: This defines when the social network should generate an anonymous audit entry. The obligation is executed any time a principal actually

makes access to an object in the social network. The obligation definition is: $((Owner(o) \neq Name(p)) \wedge (ConfiguredAuditLevel(o) = AnonymousAudit)$.

F. Additional Actions

In addition to the access control and auditing triggers described above, there are several other actions available in the context of the access control and auditing system. Principals can modify their own AllowableAuditLevelForAccessing. Principals can modify their own DefaultAuditLevelForCreation. These changes can be made at any time.

It will be interesting to have an action where a principal $p \in P$ is denied access to an object $o \in O$ and is able to temporarily override this denial and access object o. This will still cause all necessary log entries to be generated, even if they violate the AllowableAuditLevelForAccessing of p. Though this action is not allowed in our current simple model, but we want to incorporate it in future.

G. Mapping the ABAC model to a standard ACM

For a more detailed understanding, it is interesting to look at examples where the Attribute-Based Access Contol (ABAC) model has been mapped to a standard Access Control Matrix (ACM) model. This will show us on a per-object basis how the objects are protected, and how object accesses are being audited. It will also help understand the flexibility provided by the ABAC system for configuration on a per-object basis.

Table 1 is an example of a possible ACM where there are three principals, and a total of 6 pieces of data. In each cell, the audit level is listed for non-owner access to that object.

Each principal owns exactly two pieces of data. Note that it appears that the DefaultAuditLevelForCreation for principal 2 is 'No Audit'. This is noted since both principal 1 and principal 3 have 'No Audit' access to all data owned by principal 2. Similarly, the DefaultAuditLevelForCreation for principal 3 seems to be 'Complete Audit'. Principal 1 has changed the auditing level at a more granular level. Note that for object 1, other principals have 'Complete Audit' access, whereas for object 2, other users have 'Anonymous Audit' access. One possible scenario is that principal 1 has a DefaultAuditLevelForCreation of 'Anonymous Audit' but has specifically changed the ConfiguredAuditLevel for object 1 to 'Complete Audit.

Given the mapping to an ACM in Table1, Table 2. is an example of the actual Read / Write rights that principals could have on the objects.

In this example, principal 1 appears to have an AllowableAuditLevelForAccessing of either 'No Audit' or 'Anonymous Audit'. Note that the principal has read access to object 3 and object 4 ('No Audit'), but not object 5 and object 6 ('Complete Audit'). Principal 2 seems to have an AllowableAuditLevelForAccessing of 'Anonymous Audit'. Note that the only data he or she can read besides his or her own data (object 2) has an auditing level of 'Anonymous Audit'. Principal 3 has read access to everything. From this, we can conclude that either principal 3's AllowableAuditLevelForAccessing is 'Complete Audit' or the level is lower, but principal 3 has made temporary override decisions to allow the reading of all the data in the system.

V. AUDITING

Auditing seems to be a good way to at the very least expose principals to their overall privacy implications on social networks. In the proposed auditing system, there are two levels that produce audit entries (*'No Audit'* level produces no auditing information).

A. Complete Audit

In complete audit mode, a data access to an object $o \in O$ by principal $p \in P$ will generate a fully identifiable audit entry. The audit entry contains information like user name of principal performing the data access, time/date of the data access and object that was accessed. Here are a few possible audit entries generated by data accesses to objects that are marked for complete audit. Note that audit logs are only available to the owning principal p' (p' = owner(o)) at any time:

Username	Time / Date	Accessed Object
Mstockton	11/19/08 2:45pm	Picture #127
escotofio	11/01/08 7:17 pm	Blog Entry #21
Agodiyal	11/27/08 9:14pm	Status Update #32

B. Anonymous Auditing

In anonymous auditing mode, a data access to an object $o \in O$ by principal $p \in P$ will generate a partially identifiable audit entry. The goal of the entry is to allow the owning principal to understand the context of the data access without fully identifying the principal making the data access. This accomplishes two things. First the owning principal will have a better insight into their privacy implications of sharing data. Knowing something as opposed to nothing (*No Audit*) about the data access will give the principal a picture of how private their information is on the social network. Second the accessing principals privacy will be somewhat protected. Users have the opportunity to opt out of any activity that will generate audit logs. Users ultimately control how much their activity is being audited on the social network.

We believe that this anonymous auditing strategy is a key element that could make users understand the privacy implications of sharing data on social networks. At the same time, it does not compromise one of the key benefits of social networking - being able to build your network of contacts, and turn casual / weak-tie contacts into stronger personal contacts. We have discussed several possible ways to anonymously audit data access in social networks. It is unclear what strategy would provide the most benefit to the data owner while still preserving the privacy of the principal accessing the data.

A full analysis of all the potential options is outside the scope of this paper. We have chosen two possible promising anonymous auditing techniques:

TABLE I
ACM WITH AUDIT LEVELS FOR NON-OWNER ACCESS TO OBJECTS.

	Object 1	Object2	Object3	Object4	Object5	Object 6
Principal 1	Own	Own	No Audit	No Audit	Complete Audit	Complete Audit
Principal 2	Complete Audit	Anonymous Audit	Own	Own	Complete Audit	Complete Audit
Principal 3	Complete Audit	Anonymous Audit	No Audit	No Audit	Own	Own

TABLE II
READ/WRITE RIGHTS THAT PRINCIPALS HAVE ON OBJECTS.

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6
Principal 1	Write, Read	Write, Read	Read	Read		
Principal 2		Read	Write, Read	Write, Read		
Principal 3	Read	Read	Read	Read	Write, Read	Write, Read

1) Anonymous Auditing by Number of Connections: A feature of all social networking sites is the ability to friend someone else. It is a logical connection between two users on a social networking site. For two users a common measure to identify how closely tied they are is to see how many friends they share in common. If most of your friends are shared, then presumably your connection is strong. If very few of your friends are shared, then presumably your connection is weak. Providing an anonymous audit entry that discloses the number of connections you share in common with the user accessing your data is potentially a good method for anonymous auditing. Without identifying the individual accessing your data, it provides a sense of how far your data is propagating to weaker ties in your network.

2) Anonymous Auditing by Membership in a Network: In some social networking sites (e.g. Facebook), users are categorized into networks. A network is generally a high-level grouping of users. The common network categories are based on: location, school/university and company/employer. Often times users belong to multiple networks having some overlap of shared connections. This interaction could potentially exposes the users' data belonging to one network to unauthorized members of other networks (this phenomenon is commonly referred to as the *network collision* problem [2]). The anonymous audit entry could identify the user's actions by the network they belong to, thus raising user awareness about the privacy implications of sharing information and indicating potential instances of unwanted *network collisions*.

VI. PROTOTYPE IMPLEMENTATION

Sh@re is our simple social network prototype that we implemented using PHP version 5.2.6, MySQL 5.0.67, phpMyAdmin 2.11.9.2 and XAMPP v.1.6.8 [1].

The users of Sh@re are given a username and password allowing them to login to the prototype. Such usernames and passwords were created before hand to allow us to experimenting with different types of scenarios. Initially we also assigned specific profiles to each user by simply setting different audit levels (complete audit, anonymous audit, no audit) for uploading and browsing the resources in our simple social network. Once the user has logged successfully into the social network the user is presented with the following modules:

- uplo@d: Enables the user to upload resources into the social network. The default audit levels for each resource the user wishes to upload are defined by the values the user has defined in their profile. We also present the users the option to modify such values before the resources are uploaded.
- g@llery: Presents through a preview thumbnail all the resources uploaded by a particular user. By pressing the button containing the name of the resource, the user can modify the audit levels for each that specific resource. This allows the user to have more control over their resources.
- @bout: This module presents to the user their profile. They can modify not only their password, but also modify the audit levels associated initially with the resources they wish to browse and upload.
- @migo: The user is able to try and browse friends' resources. Depending on the audit level of browsing for each individual user, they will or will not be allowed to browse their friends' data. For example: if a particular user selects to browse no audit resources because they have their profile audit level to none; when they press the button of a friend that has uploaded resources that are audited or anonymously audited, the user will not be presented with those resources because it is not allowed to view them. Depending if the user can browse (according to their profile settings for audit level browsing) audited or anonymously audited resources, upon accessing the resources they generate data that goes into a log file. These log files contain the name of the resource being accessed, number of accessed times and a timestamp. For completely audited Sh@re generates and logs the usernames of the persons accessing the resource. In the case of a resource being anonymously audited Sh@re does not generate nor logs any usernames, but reports if the user accessing the resource belongs to the set of friends of the owner of that particular resource.
- logout: Logs the user off our Sh@re simple social network

Sh@re can be ported to run as an application within Face-

book by modifying each of the applications listed above to use Facebook's markup language and Facebook's user identification scheme. Facebook does not currently permit applications to extend its built-in services such as photo albums, notes, and built-in profile fields, so Sh@re will be forced to exist as a parallel infrastructure to Facebook's core. However, many other popular applications also operate in such a way. A port of Sh@re to Facebook is underway but was not operational at the time of writing. Once it is fully functional, we will be able to reach out to a significant number of users and evaluate their response to negotiated auditing.

VII. Possible Extensions

There are several possible extensions to negotiated audit that can build on what we have described and implemented so far. These include the use of contexts, broader ownership, and thresholds.

Context-Aware Auditing: The current audits contain minimal information (accessing user, time, and object) and only provide a small glimpse to users on how their data is being viewed by other users. We are currently incorporating in our prototype a 'context-aware' view of the data - providing a 'summary' of the interaction that occurred may be more valuable (e.g. a human-readable audit message stating 'user X was browsing your photos from your vacation for 20 minutes, and was specifically interested in photo B').

Social Network Data-Ownership: One of the key benefits of social networks is that people can share information about themselves with friends and other people. This information can come in many forms, including pictures of other people, tagged data (e.g. pictures or stories) involving other people, and comments about other people. One may question, who is the owner of the data when the context of the data involves several people? Currently in our model, the owner of the data is the creator of the data. However, one could argue that other people 'affected' by the data should have some decision-making power of the use of that data. Further research could include an attempt to model multi-ownership of data and the resulting audit information in a privacy-preserving way.

Threshold-based Auditing: One common form of privacy violations on social networks is online stalking. Often times, individual users will access specific user's data repetitively, to the point where the owner of the data may consider the collection of accesses to be a privacy violation simply because of the volume of data accesses. We are in the process of developing a module that adjusts the auditing strategy based on the volume of access by individual users.

VIII. Conclusion

The attribute based access control and auditing model presented in the paper provides the user with flexible, yet not overbearing configuration options. A principal can use the system to configure a global auditing policy where any object which they create, thus own, will automatically be assigned to one of the three auditing levels we implemented. With this policy, the configuration takes little effort. At the same time, such a generic configuration may not provide a sufficient auditing policy for the data owner. For example, a principal may want to allow anonymous access to most of his or her data, but may also want to specify that a certain set of data should generate complete-audit entries. Similarly, a user may desire to 'opt out' of accessing any objects that create log entries. This flexibility and ease of configuration is a novel way for users to understand their privacy implications on a social network.

Negotiated audit based social networks is the first step towards raising awareness to users about the consequences of their activities on online social networks. It addresses the weak-tie information flow problem by making users aware of who is viewing their data and thus enabling them to understand the privacy implications of uploading their data on these sites.

This work was supported in part by NSF CNS 07-16626, NSF CNS 07-16421, NSF CNS 05-24695, ONR N00014-08-1-0248, NSF CNS 05-24516, DHS 2006-CS-001-000001, and grants from the MacArthur Foundation and Boeing Corporation. We appreciated input from Sonia Jahid. The views expressed are those of the authors only.

References

[1] Apache. Xampp, http://www.apachefriends.org/en/xampp.html, May 2002.

[2] InformationWeek Cory Doctorow. How your creepy ex-co-workers will kill facebook, November 2007.

[3] Catherine Dwyer, Starr R. Hiltz, and Katia Passerini. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *Proceedings of the Thirteenth Americas Conference on Information Systems*, August 2007.

[4] Tabreez Govani and Harriet Pashley. Student awareness of the privacy implications when using facebook. none, 2005.

[5] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, Jan 1973.

[6] Ralph Gross, Alessandro Acquisti, and John H. Heinz. Information revelation and privacy in online social networks. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, New York, NY, USA, 2005. ACM Press.

[7] Saikat Guha, Kevin Tang, and Paul Francis. Noyb: Privacy in online social networks. In *Proceedings of the First ACM SIGCOMM Workshop on Online Social Networks (WOSN)*, Seattle, WA, USA, August 2008.

[8] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, 2007.

[9] Harvey Jones and José H. Soltren. Facebook: Threats to privacy, December 2005.

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

[11] Jaehong Park and Ravi Sandhu. Towards usage control models: beyond traditional access control. In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 57–64, New York, NY, USA, 2002. ACM.

[12] http://www.quantcast.com/facebook.com Quantcast. Profile for facebook.com, December 2008.

[13] http://www.quantcast.com/myspace.com Quantcast. Profile for myspace.com, December 2008.

[14] D. Watts. *Six Degrees: The Science of a Connected Age*. W.W.Norton Company, 2003.

[15] Eric Yuan and Jin Tong. Attributed based access control (abac) for web services. In *ICWS '05: Proceedings of the IEEE International Conference on Web Services*, pages 561–569, Washington, DC, USA, 2005. IEEE Computer Society.