2008 International Conference on Information Security and Assurance

Privacy Preserving Collaborative Social Network

Gary Blosser Justin Zhan
Carnegie Mellon University
gblosser, justinzh@andrew.cmu.edu

Abstract

There are many kinds of social networks in existence. To our best knowledge, there is no effort on how to construct a social network jointly from different parties. Thus, there is a need for a proper protocol to both make a collaborative social network feasible between different parties and ensure privacy. We propose a series of protocols to create and interact with a privacy preserving collaborative social networks and evaluate their potential. The protocols are implemented, tested and evaluated.

1. Introduction

At its core, a social network is a graph of connections (or edges) between points of data (commonly called nodes). Depending on the purpose of the network the edges may have direction and weight, while other social networks may be simple undirected graphs [9]. A commonly used social networking medium is email communication [1, 2, 3, 4, 5]. In an email-based social network the nodes are generally individuals while the edges represent communication between the individuals. In more complete email social networks the edges show direction and strength of the communication by tracking the direction of the emails and how often the individuals communicate. While email is the most prevalent method used to create 'new' social networks there are also a myriad of other methods which have been used. MySpace and Facebook are examples of social networking web sites that can be used to create a social network based on the 'friends' listed in each profile [7, 8]. Another method is to use phone records to build a social network. While each social network creation method is useful, most people have networks spanning more than one medium. While the email social network would show some edges, the phone and website network could reveal other important edges that do not use email communication [3].

Social networks have always existed in society in varying forms, but with the record keeping power of computers and the reach of the Internet both the interactions within and scale of these social networks are becoming apparent. Although the examination of individual social networks has proven fruitful and been the topic of much research, we feel the ability to use multiple social networks collaboratively, while retaining individual privacy, will allow the creation of an integrated 'complete' social network for research. This network could then be used in multiple fields to learn about human interactions, the distribution of information, and other application around the world.

Privacy preserving social networking is a relatively unexplored research area. While some research has been done into the privacy aspects of social network in the realms of structure [14] and attacks [12,13], there is a lack of information on collaborating multiple social networks together while maintaining privacy.

2. The Problems

To build a collaborative social network there are a number of hurdles that must be overcome. The first hurdle is how to actually combine the social network data of multiple, potentially competing, providers into a single network. This must be done in a way to prevent a provider from acquiring the data of other providers and without the central server knowing the contents of the social network. The next hurdle is keeping the collaborative social network updated so edges that have been removed in sub-networks are removed in the collaboration. Last hurdle is handling user interaction within the collaborative social network.

2.1 Combining Social Networks

To combine networks is merely an extension of the existing algorithm for creating the network in the first place. The problem comes about in preservation of both user privacy and company confidentiality. While both user privacy and corporate secrecy are common, and well explored, concerns when dealing with personal information, implementing them in a collaborative, but hostile, environment causes unique

978-0-7695-3126-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ISA.2008.33

problems. For the successful creation of the collaborative social network the following criteria will have to be met:

- Given the collaborative social networks S which contains n social networks, n_1 cannot determine the contents of n_2 when $n_1 \neq n_2$.

2.2 Updating Combined Social Network

While it would be simple to just rebuild the collaborative social network from time to time so it is kept up to date, using updates can minimize bandwidth considerations. We recommend using a pull architecture where the centralized server will periodically pull updates from the involved social networks. The problems with performing updates are the same as for the creation of the network and do not add any perceived security or privacy problems.

2.3 User Interaction

Interactions with the collaborative social network may include various types of searching and even direct browsing of the network structure. Both interactions are essentially the same and have a common result of showing a resulting social network to the user in the form of a list of nodes and their connecting edges. The most important factors in this transaction is to ensure query, network, and result privacy. In essence:

- Given a users query Q, the collaborative social network set S, cannot determine the original attribute values within Q.
- Given a users query Q, social network S_n within S cannot determine the values of any non-matching attribute value(s) in Q.
- Given a query result set R, a user cannot determine which social network S_n within S matched Q to generate R.

3. Physical Architecture

The main considerations to take into account for the physical architecture of the collaborative social networking environment is the minimization of data leakage. In particular users querying the system cannot know what social networks provided the results and social networks being used cannot obtain information about other social networks. While other methods may work, we propose a client-server architecture as the technological base of this system. This allows the users and other social networks to be clients of a server that would keep the data and communications partitioned. The channels used to communicate between the client and server would be strongly encrypted in one of the many well-known communication encryption schemes to limit data alterations, replay attacks, and the like. It is assumed that a third-party would host the server that would communicate with users wishing to query the collaborative social network and with the social networks that are being integrated together. As we defined our criteria above, the data passing through the server will not reveal details of the query or results in a usable format to this server, limiting the damage a malicious server could exert. The worst privacy violations from this server being compromised is disclosure of users interacting with the collaborative social network. The actual query and results will not be disclosed.

4. Collaborative Social Network Creation

In order to link multiple social networks together with creating a collaborative social network, matching attribute(s) must be chosen and agreed upon by both social network providers. This attribute could be an email address, name or other item that could be used to determine that nodes within different social networks are the same so the edges in both networks should be combined at that point. As the information required to join the networks is identifiable information that should be kept private due to its unique nature, steps must be taken to ensure the privacy of this information and to prevent data leakage between social networks. In order to limit the distribution of this information the comparisons will be made against hashed values. This prevents the central server from knowing the contents of the social network and still allows matches between social network nodes to allowing creation of the collaborative social network model.

4.1 Collaborative Social Network Addition Protocol

Protocol Definitions:

Objects

- *S* – The collaborative social network, which is a set of social networks
  - *S(o, e)* –
    - *o* – A node within the social network n, note that o is already hashed with function h( ) as it is created from hashed data.
      - o_j – The set of social networks that have provided these attributes, this numbering is held at the individual attributes level.
    - *e* – A set of edges related to node o
      - e_i – The set of social networks that have provided this edge
  - S_r – The resulting social network from a user query
- *N* – A social network being added to S
  - *N(d, g)* –
    - *d* – A node within the social network N
    - *g* – A set of edges related to node o

- R – A revocation social network being removed from S, note that R == N for social network identification purposes
  - *R(d, g)* –
    - *d* – A node within the social network R
    - *g* – A set of edges related to node o
- U – A user of S
  - U_q – A query containing attributes to look for within S

Functions

- *a(o)* – An attribute or set of attributes of a node which can be used to uniquely identify the node
- h( ) - A secure hash function. For example, SHA 512

Protocol:

For N(d=[1..n],g)
  N(h(d),g) is sent to S
  If any a(S(o)) == a(N(h(d))) then
    For each attribute within h(d) that matches in o
      o_j += h(N)
    Else
      Add new attribute from h(d) to o with a o_j=h(N)
    End
    For any edge within e that matches an edge within g
      e_i += h(N)
    Else
      Add non-matching edges within set g to set e with e_i=h(N)
    End
    Else
      Add new node and edge set N(h(d),g) to S
    End
Next

5. Collaborative Social Network Updates

While the adding and merging of social networks is fairly simple to perform, the removals and updates that happen periodically to the social network are more complicated to handle. This is where the attribute and edge numbering systems are invaluable. A member social network can choose to update in one of two ways. The first method is to revoke their entire social network then re-add it to the collaborative social network. While easy to implement, this method is fairly wasteful on bandwidth and processing time. The second, recommended, method is to send a subset social network that contains a revocation list for removing changed or deleted data and a addition list with the new information.

Collaborative Social Network Revocation Protocol

Protocol:

For R(d=[1..n],g)
  R(h(d),g) is sent to S
  If any a(S(o)) == a(R(h(d))) then
    For each attribute within h(d) that matches in o
      o_j -= h(R)
      If o_j empty then remove o
    End
    For any edge within e that matches an edge within g
      e_i -= h(R)
      If e_i empty then remove e
    End
  End
Next

6.Collaborative Social Network Interaction

While there will undoubtedly be many forms and variations in interacting with the collaborative social network, the common thread will be a basic comparison of user-provided attributes and the contents of nodes in the network. For this collaborative social network we assume the user knows what they are searching for and expect to receive a social network matching their search criteria while maintaining query privacy.

Collaborative Social Network Searching Protocol

Protocol:

U generates query U_q which contains attributes of interest
  $h(U_q)$ is sent to S
  For S(o) with attributes matching $h(U_q)$
    S_r += S(o,e) not including o_j or e_i
  Next
  For each e within S_r
    If both nodes of e are not within S_r then
      Remove e from S_r
    End
  Next
  S_r is sent to U
U uses hash values from U_q to reveal matching attributes in S_r

7. Experiment

The aforementioned protocols for social network interaction within a client-server architecture were implemented using the freely available Enron email corpus [4,11]. The first decision made was to use the individual email addresses as the unique identifier, or a(o), of the node. Next, in order to develop a comparison base, the entire corpus was processed into a single weighted social network using the following rules:

1. Ignore any messages with more than 10 recipients. This rule is used to reduce spam and ignore

corporate mass mailings that could cause false edges [1].

2. Assign varying weights to the edges depending on how many recipients there are and if the email is 'To', 'CC', or 'BCC' the recipient:
    1. To – 100 weight.
    2. CC – 25 weight. From our experience the CC has little to contribute to social interactions in business and is usually used to inform superiors of information.
    3. BCC – 50 weight. The BCC serves the same purpose as a CC usually, but other recipients cannot see the communication, this means the recipients will not have a weight to add to the total for this edge, thus the higher value.
    4. Variance – The assigned weight is then divided by the total number of recipients. The recipients of most messages will also have a copy of the message in their email, so this measure prevents messages to many people from being unfairly weighted, i.e., a message to 5 people that is not divided would end up giving a total weight of 500 to each person involved, divided each person only receives 100.
3. Drop any edges with a total weight less than 500. This removes most spam and attempts to further limit the results to social communication between individuals [1]. This rule reduces the edge list to 14,914 from the original 83,724.

Finally, we hash all of the email addresses, for this experiment we chose to use SHA512 as our function h(). The resulting 14,914 unique edges and weights of this complete social network will be used for comparison.

7.1 Collaborative Social Network Creation Experiment

For testing the network creation algorithm, the corpus was split, preprocessed, run through the creation protocol, post-processed, then validated against the complete social network detailed above. The splitting of the original 83,724 edges between the sub-social networks was done using a random number generator, while the preprocessing was done by using rules 1 and 2 from section VIII above. While unmeasured, the transmission of the hashed edge records from the sub-social networks to the central server would be dependent upon the speed of the link and number of records. There would only be a single transfer of data from the sub-social network to the central server. The central server would use this transferred data to build the collaborative social network. The creation protocol was both timed and validated for each set of data.

7.2 Results

Subsets	Average Time	Validation
2	22.5841	14914/14914
5	22.8626	14914/14914
10	22.6705	14914/14914
100	22.0946	14914/14914
1000	23.0157	14914/14914

7.3 Evaluation

As can be seen from the data above, there is little to no change in time taken by increasing the number of submitting social networks. The process is linear with the only significant change due to the size of the dataset. The data was validated by confirming that the identifying hashes and generated weights match the comparison base. The only downside in the creation protocol is that the entirety of each participating social network must stored within the central server to perform merging and post-processing.

7.4 Collaborative Social Network Revocation Experiment

The revocation process consisted of removing the revoked attributes, subtracting the removed weight from the collaborative social network edges, and post-processing those edges to ensure their validity. The process was first tested with a small subset of records to ensure reliable results before being done on the larger dataset for time complexity testing.

7.5 Results

Removed	Average Time
15 (0.1%)	0.0644
149 (1%)	0.3342
1491 (10%)	1.2822
2982 (20%)	2.1909
7557 (50%)	4.9034

7.6 Evaluation

Searching of the data was fast due to indexed storage and the actual removal was also quite simple to accomplish. Furthermore, by applying the post-processing to only the affected records time complexity was further reduced. Overall the process is linear and dependent solely upon the number of edges in the social network.

7.7 Collaborative Social Network Searching Protocol Experiment

Due to the customizable nature of the searching protocol many different tests were performed. The first series of tests were with a single email address and simply returned the node if the email address existed. The second series of tests used multiple email addresses to find if there existed any edges between the nodes. The time taken for searching was found to be based upon the number of criteria queried for and the total size of the collaborative social network.

8. Criteria Evaluation

8.1 Criteria 1 – Network Provider Secrecy

Given the collaborative social networks S which contains n social networks, n_1 cannot determine the contents of n_2 when $n_1 \neq n_2$.

Because all data stored within S is hashed social network N_1 that cannot determine the exact attribute contents of N_2 through queries. On the other hand, when the number of social networks contributing to S is only 2, some social network structure can be found from N_2. If N_1 used a query to browse the entire contents of S and removed their N_1 social network data from the results a partial or complete social network of N_2 would be revealed. If there was no overlap between N_1 and N_2 it would be a complete structural map of N_2. However, if there was overlap then some portions that are shared between N_1 and N_2 would be removed by removing N_1 from S. Another method that could be performed is for N_1 to revoke their entire social network from S, retrieve the structure of N_2 from S, then add N_1 back into S. While these methods provide a minor threat they still do not reveal the identity of N_2 or any attribute data from N_2. Furthermore, when there are more than 2 social networks within S these attacks are not longer effective without collaboration of x-1 social network providers where x is the number of social networks contained in S.

Another danger in determining the data belonging to a specific network is capturing the traffic between a social network and the server during network creation or update. To minimize this threat we have recommended the use of communications encryption like SSL.

The final danger is S being malicious itself. In this situation the entire structure of each N is available due to the edges and attributes being tagged with their source social network. However, the attribute values are all in hashed format making the structure the only visible item to a malicious S.

8.2 Criteria 2 – Query Privacy

Given a users query Q, the collaborative social network set S, cannot determine the original attribute values within Q.

As S contains only hashed attribute values and Q is also hashed for comparison and security purposes there is no way for a malicious S to determine the contents of S. However, if a malicious S and malicious participating social network N are working together then any matching attributes in Q that are a part of N can be revealed. If malicious N happens to match all attributes within Q then the entirety of Q can be revealed to S. However, in normal situations participating social networks of S will never see any queries preventing the contents of Q from being revealed.

8.3 Criteria 3 – User Query Data Leak Minimization

Given a users query Q, social network S_n within S cannot determine the values of any non-matching attribute value(s) in Q.

As S contains the complete hashed set of attributes and edges for all Sn no queries ever pass from S to S_n. This prevents any knowledge of Q. However, if S is malicious then Q may be passed to S_n. In this case matching hashed attributes within Q can be determined and reversed into the actual value. However, non-matching attribute original values still cannot be determined due to hashing.

8.4 Criteria 4 – Source Social Network Privacy

Given a query result set R, a user cannot determine which social network S_n within S matched Q to generate R.

As o_j and e_i are explicitly removed from the results generated by S there are no obvious ways to determine S_n from R. However, the attacks outlined in VIII.A above are still threats when the number of social networks participating in S are small. Otherwise there are no problems with this criteria.

8.5 Overall Evaluation

Though the protocols do meet the criteria and segregate the data sources, they also provide a central point of failure in S. The dangers of S being malicious are quite large and would make it the logical target of any malicious user or malicious social network. This would be minimized if the user could directly generate S and query against their own data store. However, this opens the problem of allowing network identification and user identification. This is a trade-off that deserves future consideration, especially since a local S prevents any possibility of queries from being identified.

9. Conclusion and Future Work

While a client-server architecture is useful, it also provides a central point of failure and an obvious target for malicious activity. Furthermore, the central server must take care of all processing and query handling putting the resource burden in one place. It would be more efficient to have the user directly build and process the social network on their own system to distribute the processing load to each user instead of a centralized server. However, this will bring many problems of social network and user identification into play and requires more thought and examination.

The sheer possible size of the collaborative social network requires highly optimized searching methods to quickly and reliably retrieve complete data from the network. In a giant social network consisting of millions to billions of edges it may be more useful for the search to build a customized sub-social networks with certain properties and edges on the fly than to browse randomly or use simple keywords.

In conclusion, we have seen the usefulness of many small social networks in various fields and have proposed a method to combine these networks together while retaining the sanctity of data for the owners, privacy for the users, and completeness for research purposes. We feel this will allow for a broader understanding of social networks by using a more complete network.

REFERENCES

[1] Adamic, L. and Adar, E. *How to search a social network.* Social Networks 27 (3). 187-203. Elsevier, 2005.

[2] Culotta, A. and Bekkerman, R. et al. *Extracting Social Networks and Contact Information From Email and the Web.* Defense Technical Information Center, 2005.

[3] Grippa, F. and Zilli, A. and Laubacher, R. and Gloor, P. *E-mail May Not Reflect The Social Network.* International Sunbelt Social Network Conference, 2006.

[4] Klimt, B. and Yang, Y. *The Enron Corpus: ANew Dataset for Email Classification Research.* Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004: Proceedings. Springer, 2004.

[5] Bird, C. and Gourley, A. and Devanbu, P. and Gertz, M. and Swaminathan, A. *Mining email social networks.* Proceedings of the 2006 international workshop on Mining software repositories, 137-143. ACM Press New York, NY, US, 2006.

[6] Csányi, G. and Szendrői, B. *Structure of a large social network.* Physical Review E 69 (3), 36131. APS, 2004.

[7] *MySpace.com – a place for friends.* http://us.myspace.com/. 2008.

[8] *Facebook*. http://www.facebook.com/. 2008.

[9] *Social Network – Wikipedia, the free encyclopedia.* http://en.wikipedia.org/wiki/Social_network. 2008.

[10] Zhan, Z. *Privacy-Preserving Collaborative Data Mining.* Ph.D thesis. University of Ottawa, 2006.

[11] *Enron Email Dataset.* http://www.cs.cmu.edu/~enron/. January, 2008.

[12] Wang, D. and Liau, C. and Hsu, T. *Privacy Protection in Social Network Data Disclosure Based On Granular Computing.* Proceedings of IEEE International Conference on Fuzzy Systems, Vancouver, BC, Canada. 2006.

[13] Zhou, B. and Pei, J. *Preserving Privacy in Social Networks Against Neighborhood Attacks.* Proceedings of the 24th International Conference on Data Engineering (ICDE '08), Cancun, Mexico. 2008.

[14] Singh, L. and Zhan, J. *Measuring Topological Anonymity in Social Networks.* Proceedings of IEEE International Conference on Granular Computing, Silicon Valley, USA. 2007.