# RETRIEVING HIDDEN FRIENDS A COLLUSION PRIVACY ATTACKS AGAINST ONLINE FRIEND SEARCH ENGINE

**TIKKA SIVA DURGA [#1], K.VENKATESH [#2]**

[#1] MSC Student, Master of Computer Science,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

[#2] Assistant Professor, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

**Abstract**

Online Social Networks (OSNs) are providing a variety of applications for human users to interact with families, friends and even strangers. One of such applications, friend search engine, allows the general public to query individual users' friend lists and has been gaining popularity recently. . In this paper, we propose an advanced collusion attack, where a victim user's friendship privacy can be compromised through a series of carefully designed queries coordinately launched by multiple malicious requestors. The effect of the proposed collusion attack is validated through synthetic and real-world social network data sets

## 1. INTRODUCTION

Online social networks (OSNs) have become very popular in recent years, such as Facebook and Twitter, which have been part of many people's daily life. The OSNs provide different applications for people to share their information and interact with each other. One of the most popular applications is the friend search engine, which allows users to query friend lists of other users. To increase their sociability and attract more users, OSNs tend to release users' friends as many as possible, as it is believed that the larger number of common friends are displayed, the more likely the requestor and the queried user would connect later.

However, this search engine may expose more friendship information than what a queried user is willing to share, which is considered as a privacy breach. A few researchers have observed such an issue by randomly crawling an OSN through the friend search API [1]. Also, they concluded that without appropriate defenses, one could discover all users' friendships in the OSN without using many queries [2], [3]. If such a privacy breach is not well dealt with, the OSN users may feel panic and hesitate to continue using the OSNs.

In our preliminary work, we designed a privacy-aware friend display scheme in [4], which cannot only successfully preserve users' friendship privacy but also boost the sociability of the OSN. This scheme is one of the most advanced researches on preserving user privacy for friend search engines, which has been verified to successfully prevent attacks from being launched by independent attackers. However, collusion attacks, where multiple malicious requestors share their knowledge and coordinately launch queries, may make the defense scheme ineffective. In this paper, we particularly focus on the design of collusion attacks against users' friendship privacy in OSNs. The major contributions of this paper are listed as follows.

First, to the best of our knowledge, we are the first researchers studying such advanced privacy attacks as collusion attacks against friend search engine in OSNs. Second, in-depth analysis has been provided on querying a small scale complete graph as well as a general network in various scenarios, which well explains the fundamental reasons of why and how the proposed attack is designed.

In particular, we observe the defense scheme's [4] asymmetric disclosure of users' symmetric friendships. By taking advantage of it, we design an advanced collusion attack, in which multiple malicious requestors closely coordinate with one another to launch their queries on different but related users in well designed orders. The design logic can be generally applied to launch attacks against any friendship privacy preserving solutions that disclose the symmetric friendship in an asymmetric way. Third, the proposed collusion attack is designed to carefully select which users to query, which can significantly reduce the total amount of query effort.

Fourth, to evaluate the effectiveness of our proposed attack strategy, we implement and run it on one synthetic data set and three large scale real-world data sets. Experiment results demonstrate that the proposed attack strategy works efficiently and effectively on large scale data sets. By comparing the proposed collusion attack with a naive direct attack, we find that our strategy performs better in terms of both the success rate and the required number of malicious requestors to compromise a user's friendship privacy. Last but not least, our research on this advanced collusion attack helps us better understand the attack design and shed lights on the design of a securer privacy preserving friend search engine in the future.

**PROBLEM STATEMENT**

First, to the best of our knowledge, we are the first researchers studying such advanced privacy attacks as collusion attacks against friend search engine in OSNs. Second, in-depth analysis has been provided on querying a small scale complete graph as well as a general network in various scenarios, which well explains the fundamental reasons of why and how the proposed attack is designed.

**PURPOSE**

Collusion attacks can be defined as attacks that involve multiple malicious entities aiming at obtaining greater gain than what the entities benefit from individually launched attacks. The multiple entities can be fake accounts created by a single attacker or by different real attackers. Compared to individual

attacks, collusion attacks can use more complicated attack strategies and often exploit system vulnerabilities that cannot be discovered by individual attacks.

**OBJECTIVE**

In the proposed system, First, to the best of our knowledge, the system is the first researchers studying such advanced privacy attacks as collusion attacks against friend search engine in OSNs.Second, in-depth analysis has been provided on querying a small scale complete graph as well as a general network in various scenarios, which well explains the fundamental reasons of why and how the proposed attack is designed.

## 2. LITERATURE SURVEY

### 2.1 INRODUCTION

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need lot of external support. This support obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into for developing the proposed system.

### 2.2 RELATED WORK

**1. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels**

**AUTHORS:** R. Canetti and H. Krawczyk

We present a formalism for the analysis of key exchange protocols that combines previous definitional approaches and results in a definition of security that enjoys some important analytical benefits: (i) any key-exchange protocol that satisfies the security definition can be composed with symmetric encryption and authentication functions to provide provably secure communication channels (as defined here); and (ii) the definition allows for simple modular proofs of security: one can design and prove security of key-exchange protocols in an idealized model where the communication links are perfectly authenticated, and then translate them using general tools to obtain security in the realistic setting of adversary-controlled links. We exemplify the usability of our results by applying them to obtain the proof of two classes of key-exchange protocols, Diffie-Hellman and key-transport, authenticated via symmetric or asymmetric techniques.

2. Map Reduce: Simplified Data Processing On Large Clusters

**AUTHORS: J. Dean and S. Ghemawat**

Map Reduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across largescale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct Map Reduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand Map Reduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.

3. Scalable Security for Petascale Parallel File Systems

**AUTHORS: A.W. Leung, E.L. Miller, and S. Jones**

Petascale, high-performance file systems often hold sensitive data and thus require security, but authentication and authorization can dramatically reduce performance. Existing security solutions perform poorly in these environments because they cannot scale with the number of nodes, highly distributed data, and demanding workloads. To address these issues, we developed Maat, a security protocol designed to provide strong, scalable security to these systems. Maat introduces three new techniques. Extended capabilities limit the number of capabilities needed by allowing a capability to authorize I/O for any number of client-file pairs. Automatic Revocation uses short capability lifetimes to allow capability expiration to act as global revocation, while supporting non-revoked capability renewal. Secure Delegation allows clients to securely act on behalf of a group to open files and distribute access, facilitating secure joint computations. Experiments on the Maat prototype in the Ceph petascale file system show an overhead as little as 6--7%.

4. Scalable Performance Of The Panasas Parallel File System

**AUTHORS: B. Welch, M. Unangst, and B. Zhou**

The Panasas file system uses parallel and redundant access to object storage devices (OSDs), per-file RAID, distributed metadata management, consistent client caching, file locking services, and internal cluster management to provide a scalable, fault tolerant, high performance distributed file system. The clustered design of the storage system and the use of client-driven RAID provide scalable performance to many concurrent file system clients through parallel access to file data that is striped across OSD storage nodes. RAID recovery is performed in parallel by the cluster of metadata managers, and declustered data placement

yields scalable RAID rebuild rates as the storage system grows larger. This paper presents performance measures of I/O, metadata, and recovery operations for storage clusters that range in size from 10 to 120 storage nodes, 1 to 12 metadata nodes, and with file system client counts ranging from 1 to 100 compute nodes. Production installations are as large as 500 storage nodes, 50 metadata managers, and 5000 clients.

## 3. EXISTING SYSTEM

Collusion attacks can be defined as attacks that involve multiple malicious entities aiming at obtaining greater gain than what the entities benefit from individually launched attacks. The multiple entities can be fake accounts created by a single attacker or by different real attackers. Compared to individual attacks, collusion attacks can use more complicated attack strategies and often exploit system vulnerabilities that cannot be discovered by individual attacks.

**LIMITATION OF EXISTING SYSTEM**

The following are the main limitations of the existing system. They are as follows:

1. There is no filtering system to find Privacy Attack.
2. Less security due No URL Based attack Detection.

## 4. PROPOSED SYSTEM

In the proposed system, First, to the best of our knowledge, the system is the first researchers studying such advanced privacy attacks as collusion attacks against friend search engine in OSNs.Second, in-depth analysis has been provided on querying a small scale complete graph as well as a general network in various scenarios, which well explains the fundamental reasons of why and how the proposed attack is designed.

**ADVANTAGES OF THE PROPOSED SYSTEM**

The following are the advantages of the proposed system, they are as follows:

1. The system provides the flexibility for individual users to determine the number of friends, say k, to display in response to friend queries.
2. Particularly focus on the design of collusion attacks against users' friendship privacy in OSNs

## 5. SOFTWARE PROJECT MODULES

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. The front end of the application takes JSP,HTML and Java Beans and as a Back-End Data base we took My

SQL data base. The application is divided mainly into following 2 modules and inside these modules there are several other sub modules present. They are as follows:

### 5.1 OSN Server Module

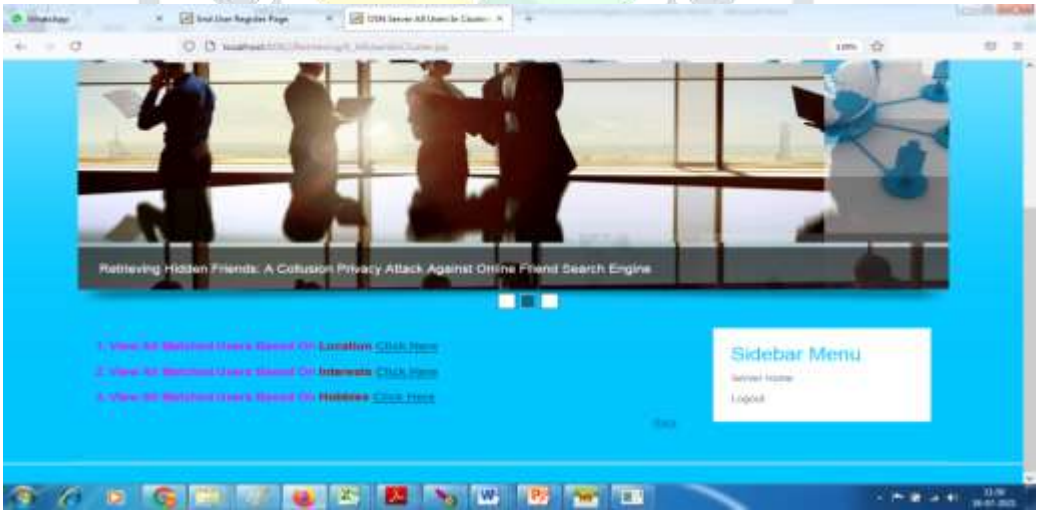
In this module, the Admin has to login by using valid user name and password. After login successful he can perform some operations such as View All Users And Authorize,View Friend Request and Response,View All Matched Users,View All User Post Posts,View All Posts Recommended Details,View All Friend Recommended Details,View All Collusion Attacker Details,View Posts Scores Results ,View Collusion Attacker Results

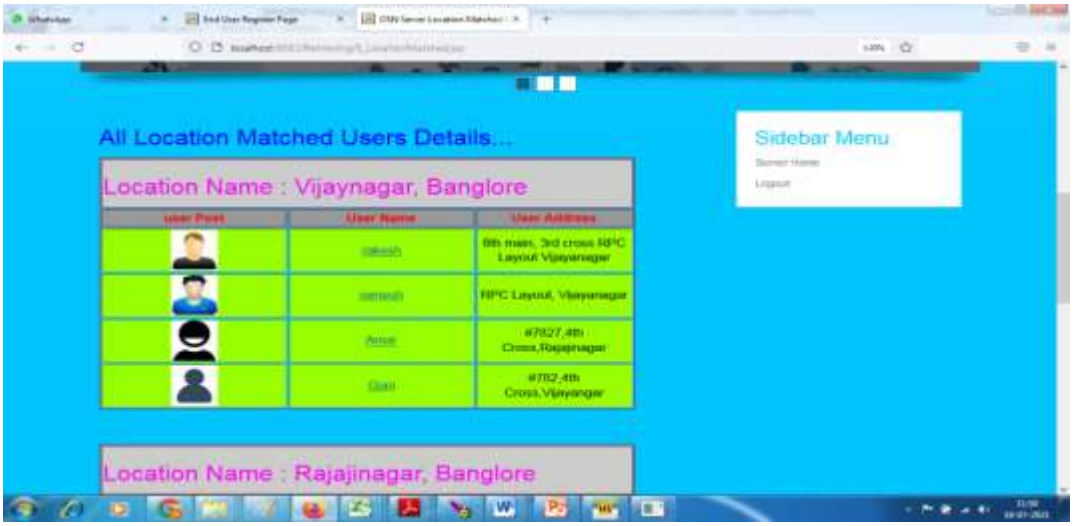
### 5.2 User Module

In this module, the Admin has to login by using valid user name and password. After login successful he can perform some operations such as View All Users And Authorize,View Friend Request and Response,View All Matched Users,View All User Post Posts,View All Posts Recommended Details,View All Friend Recommended Details,View All Collusion Attacker Details,View Posts Scores Results ,View Collusion Attacker Results

## 6. OUTPUT RESULTS

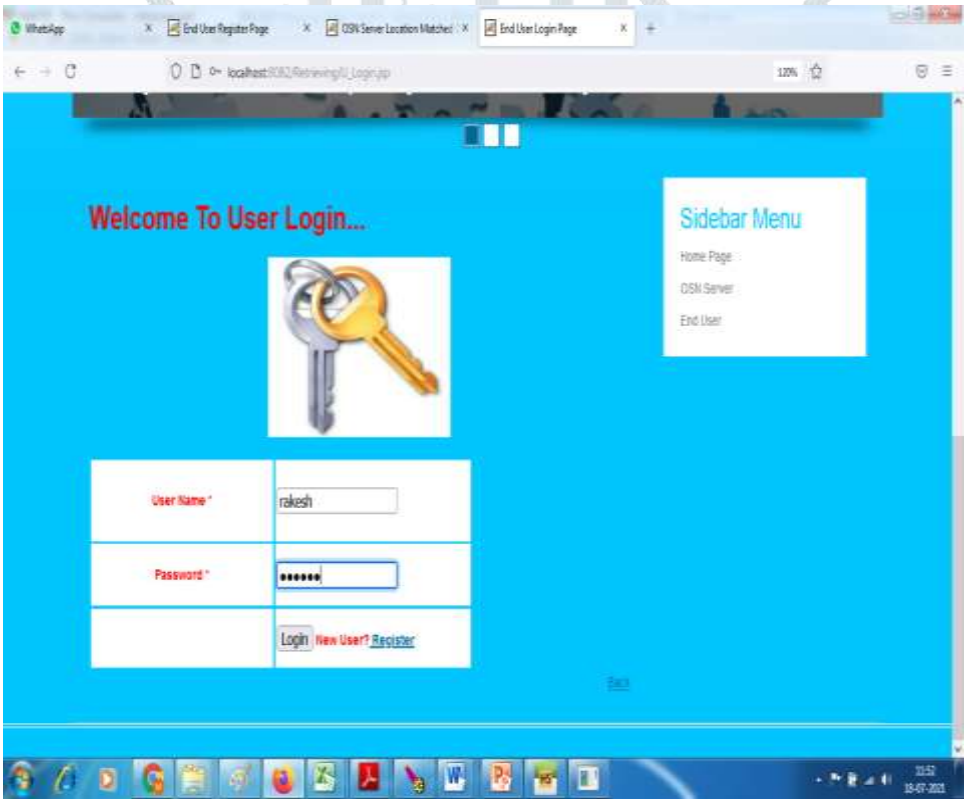
**Admin Views the Matched users**






**Represents the Matched Users**

**User Login**




**Represents the User Login and Home Page**

**View All Friends based on profile keyword matching**


**Represents the All Friends and recommendations**

**User can View His Post Details**


**Represents the user own post and his/her Friends post**

## 7. CONCLUSION

In this proposed work, we have proposed an advanced collusion attack strategy where multiple attackers with very limited initial knowledge (i.e. only the victim node) can successfully penetrate the defense and violate victim node's privacy settings on friend search engine. In particular, we start this study with a simple and small social clique model, aiming to deeply understand users' friendship types and reveal the fundamental reasons why collusion attacks can be done successfully. Based on observations made from this model, we further propose to classify social network users into non-popular users and popular users; develop different attack strategies against them; and illustrate the attack effectiveness in a general social network through different scenarios. Experiment results show that our proposed collusion attack strategy has achieved high success rate by using limited number of malicious requestors.

## 8. REFERENCES

[1] "Friendlist api." [Online]. Available: https://developers.facebook.com/docs/reference/fql/friendlist

[2] J. Bonneau, J. Anderson, F. Stajano, and R. Anderson, "Eight friends are enough: social graph approximation via public listings," in Proceedings of ACM SNS'09, 2009, pp. 13–18.

[3] A. Yamada, T. H.-J. Kim, , and A. Perrig, "Exploiting privacy policy conflicts in online social networks," in Technical report, 2012.

[4] N. Li, "Privacy-aware display strategy in friend search," in Proceedings of IEEE International Conference on Communications (ICC), Communication and Information Systems Security Symposium, 2014, pp. 951–956.

[5] R. Fogues, J. M. Such, A. Espinosa, and A. Garcia-Fornes, "Open challenges in relationship-based privacy mechanisms for social network services," International Journal of Human-Computer Interaction, vol. 31, no. 5, pp. 350–370, 2015.

[6] L. Guo, C. Zhang, and Y. Fang, "A trust-based privacy-preserving friend recommendation scheme for online social networks," IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 4, pp. 413–427, 2015.

[7] Z. Feng, H. Tan, and H. Shen, "Relationship privacy protection for mobile social network," in Advanced Cloud and Big Data (CBD), 2016 International Conference on. IEEE, 2016, pp. 215–220.

[8] N. Li, N. Zhang, and S. Das, "Relationship privacy preservation in publishing online social networks," in Proceedings of the Third IEEE International Conference on Social Computing (SocialCom'11), MIT, Boston, US, 2011.

[9] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in Proceedings of the 24th IEEE International Conference on Data Engineering(ICDE'08), 2008, pp. 506–515.

[10] J. Cheng, A. W. Fu, and J. Liu, "K-isomorphism: Privacy preservation in network publication against structural attack," in SIGMOD'10, 2010.

[11] S. Das, O. Egecioglu, and A. E. Abbadi, "Anonymizing edge-weighted social network graphs," in Technical Report CS-2009-03, Computer Science, The University of California, Santa Barbara, 2009.

[12] M. Hay, G. Miklau, D. Jensen, D.Towsley, and P. Weis, "Resisting structural reidentification in anonymized social networks," in PVLDB'08, 2008.

[13] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," SIGKDD Explorations, vol. 10, no. 2, pp. 12–22, December 2008.

[14] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in Proceedings of the 1st ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD, PinKDD2007, In Conjunction with the 13th ACM SIGKDD International Conference in Knowledge Discovery and Data Mining, KDD,(PinKDD'07), San Jose, CA, USA, 2007.

[15] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preserving in social networks against sensitive edge disclosure," in Technical Report CMIDA-HiPSCCS 006-08, Department of Computer Science, University of Kentucky, KY, 2008.