International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Volume.06, Issue.01,
January-June, 2018,
Pages:0111-0113

G-Match Secure and Privacy Preserving Group Matching in Social Networks

M. SUNEETHA[1], UCHV PRASAD[2]

[1]PG Scholar, Dept of CSE, DJR College of Engineering & Technology, AP, India, E-mail: sunithamaridu@gmail.com.
[2]Associate Professor, Dept of CSE, DJR College of Engineering & Technology, AP, India, E-mail: chanuuppalapati09@gmail.com.

Abstract: Groups are becoming one of the most compelling features in both online social networks and Twitter-like micro blogging services. A stranger outside of an existing group may have the need to find out more information about attributes of current members in the group, in order to make a decision to join. However, in many cases, attributes of both group members and the stranger need to be kept private and should not be revealed to others, as they may contain sensitive and personal information. How can we find out matching information exists between the stranger and members of the group, based on their attributes that are not to be disclosed? In this paper, we Fesent a new group matching mechanism, by taking advantage private set intersection and ring signatures. With our scheme, a stranger is able to collect correct group matching information while sensitive information of the stranger and group members are not disclosed. Finally, we propose to use batch verification to significantly improve the performance of the matching process.

Keywords: Groups, Social, Blogging, Matching, Mechanism.

I. INTRODUCTION

As online social networks and Twitter-like micro-blogging services redefine our lifestyle, groups are becoming one of the most frequently used features. Groups are, in general, formed with common attributes, such as geographic locations and hobbies. However, the features of a group are generally described by only a few keywords or a short description, which sometimes is not enough for users to make decisions when choosing an appropriate group for themselves. Especially, when several groups have similar (or even the same) keywords and descriptions, it is very inconvenient for users to choose the most suitable one among these groups. In order to make a better decision when choosing a group to join, a stranger with a profile of his own attributes — who is still an outsider of the group — needs to collect detail matching information from all the group members' profiles. Such a problem is referred as to group matching. In most situations, attributes of users are sensitive, such as personal health records and religious preferences. It is typical for a user to store these attributes privately, so that only his friends or members in the same group are able to reveal these attributes, but strangers or any third party cannot learn these sensitive information. Unfortunately, collecting group matching information using these sensitive attributes may introduce a number of privacy problems. On one hand, since the stranger is not familiar with the group, the stranger does not want to reveal his sensitive attributes to any group member during the matching process.

On the other hand, because the stranger is an outside and untrusted user to the group, each group member is reluctant to reveal his own attributes and the exact matching results between two entities to the stranger. To make matters more challenging, each group member needs to generate a signature on his matching response, which contains matching information between the stranger and himself, and sends the signature and the matching response together to the stranger, so that the stranger is convinced the matching response is reliable and correct. Unfortunately, due to the unforgeability of signatures (only the entity with the knowledge of the private key can create valid signatures), the stranger is able to learn the identity of the signer on each matching response, and reveal exact matching information between himself and each group member.

II. EXISTING SYSTEM

Beyond asking for explicit user input, earlier work by Li and Croft focused on handling re-cency queries, which are queries that are after recent events or breaking news. Li and Croft's time sensitive approach processes a re-cency query by computing traditional topic similarity scores for each document, and then "boosts" the scores of the most recent documents, to privilege recent articles over older ones. In contrast to traditional models, which assume a uniform prior probability of relevance p(d) for each document d in a collection, Li and Croft define the prior p(d) to be a function of document d's creation date. The prior probability p(d) decreases exponentially with time, and hence recent documents are ranked higher than older documents. Li and Croft 's strategy is designed for queries that are after recent documents, but it does not handle other types of time-sensitive queries, such as [Madrid bombing), (Google IPO], or even (Sarkozy French elections) (in May 2008), that implicitly target one or more past time periods.

Disadvantages:

- Previous system difficult to search and joint the group.
- There is no group verification and batch verification to search.

III. PROPOSEDSYSTEM

We propose a more general framework for efficient way to client can joint the group of social network. We propose a more general framework for handling batch and During the group matching, our scheme should be able to provide the following desirable privacy properties.

Advantages:

- Security purpose user detail encrypted and decrypted.
- Implemented the group verification to fast search best group in Social network and user can joint in best group
- Batch verification increase efficient way to validate the group
- Group adding also secure way to share the key to add groups,
- Admin have control to add group and delete group.

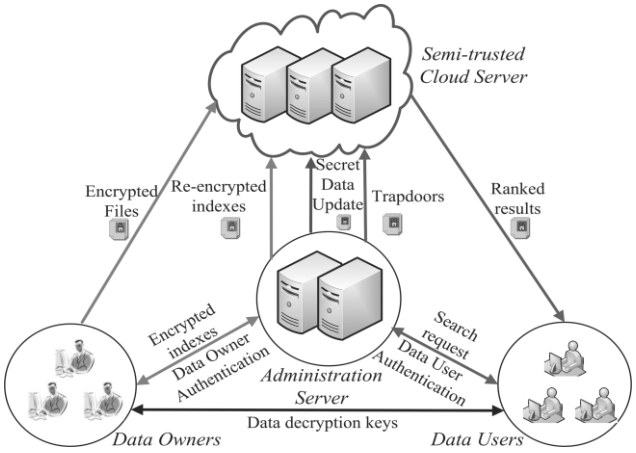

Fig.1. System Architecture.

IV. RELATED WORK

A. Two-Party Private Matching

In this paper proposed a private matching scheme, which allows a client and a server compute the set intersection with their own private sets. During private matching, the client only obtains the set intersection while the server does not know any matching result. Agrawalel al. Introduced a private matching scheme between two databases using commutative encryptions. Hazay and Lindell exploited pseudo random functions to evaluate set intersection. In Dachman-Soledet al. exploited polynomial evaluations to compute the set intersection between two parties, and also leveraged Shamir secret sharing and cut-and-choose protocol to improve efficiency. Recent work in introduced an authorized private set intersection (APSI) based on blind AES signatures. In APSI, each element in the client's set must be authorized by some mutually trusted authority.

B. Multi-Party Private Matching

In this paper proposed a multi -party private matching scheme to compute the union, intersection and element reduction operations for multiple sets. However, this scheme requires a group decryption among multiple entities, which is impractical between the stranger and group members in social networks. Ye el al extended previous scheme to a distributed scenario with multiple servers. The dataset of the original server is shared by several sub-servers using Shamir secret sharing. Proposed a private multi - party set intersection scheme based on the two-dimensional verifiable secret sharing scheme.

C. Private Matching In Social Networks

In this paper focuses on finding the best matched user from the group in mobile social networks. Yang et al. introduced E-Small Talker, which allows users to privately match other people in mobile social networks using the iterative bloom filter (IBP) protocol.

V. CONCLUSION

In this paper, we proposed Gmatch, a secure and privacy-preserving group matching in social networks. With Gmatch, the stranger can successfully collect group matching information while the private information of group members is preserved. Our experimental results show that Gmatch can efficiently compute correct group matching information with batch verification.

VI. RESULTS

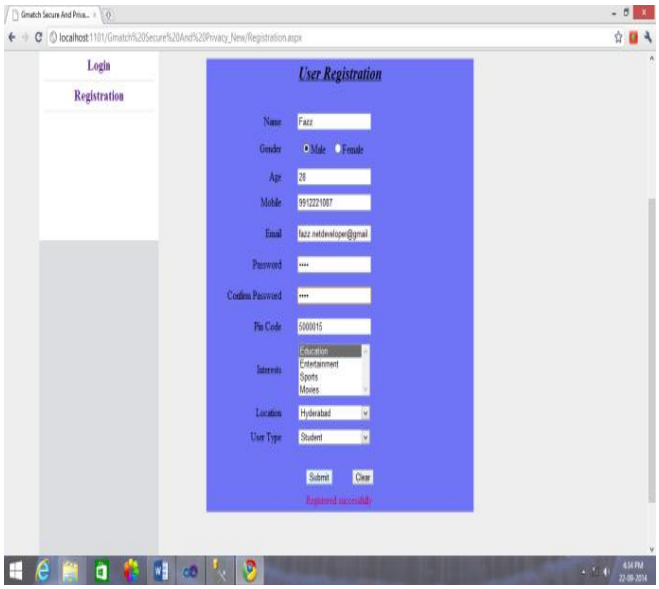

Fig.2.

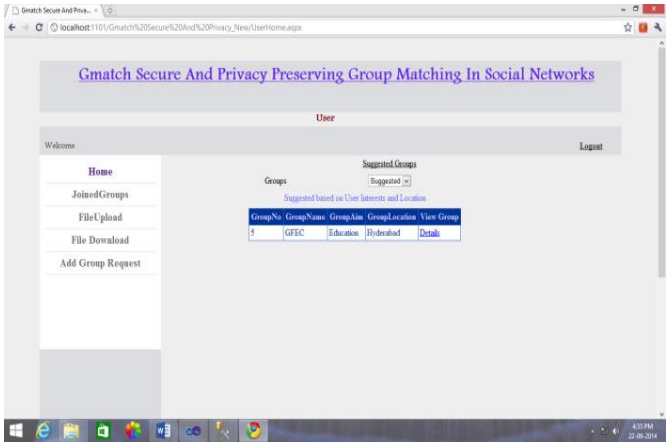

Fig.3.

VI. REFERENCES

[1]Beginning ASP.NET 4: in C# and VBby ImarSpaanjaars.
[2]Programming ASP.NET 3.5by Jesse Liberty, Dan Maharry, Dan Hurwitz.
[3]Beginning ASP.NET 3.5 in C# 2008: From Novice to Professional, Second Editionby Matthew MacDonald.
[4]Data Communications and Networking, by Behrouz A Forouzan.
[5]Computer Networking: A Top-Down Approach, by James F. Kurose.
[6]Operating System Concepts, by Abraham Silberschatz.
[7]M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. USB-EECS-2009-28, Feb 2009
[8]Amazon Web Services (AWS), online at http://aws. amazon.com.
[9]Google App Engine, Online at http://code.google.com/ appengine/.
[10]Microsoft Azure, http://www.microsoft.com/azure/.

Author's Profile:

M. Suneetha, received her B.Tech degree in computer science and engineering and pursuing M.Tech degree in computer science and engineering from , DJR College of Engineering & Technology.

U CH V Prasad M.Tech received his M.Tech degree and B.Tech degree in computer science and engineering .He is currently working as an Assoc Professor in DJR College of Engineering & Technology**.**