CERIAS

The Center for Education and Research in Information Assurance and Security

PURDUE UNIVERSITY

Privacy-Preserving and Efficient Friend Recommendation in Social Networks

Bharath K. Samanthula[1] , Lei Cen[2], Wei Jiang[3], Luo Si[4]

[1,3]Dept. of Computer Science, Missouri S&T, {bspq8, wjiang}@mst.edu

[2,4]Dept. of Computer Science, Purdue University, {lcen, lsi}@cs.purdue.edu

[4]CERIAS

Introduction

Friend recommendation is a well-known application in many social networks and has been studied extensively in the recent past. However, with the growing concerns about users' privacy, there is a strong need to develop privacy preserving friend recommendation methods for social networks. In this paper, we propose two novel methods **of Privacy-Preserving Friend Recommendation**(**FFPR**) based on common neighbors proximity.

The first method is based on the properties of **additive homomorphic encryption** scheme and also utilizes a universal hash function for efficiency purpose. Nevertheless, this efficiency comes at the expense of degraded accuracy due to the involved hash collisions. Whereas, the second method utilizes the concept of **protecting the source privacy** through randomizing the message passing path and recommends friends accurately.

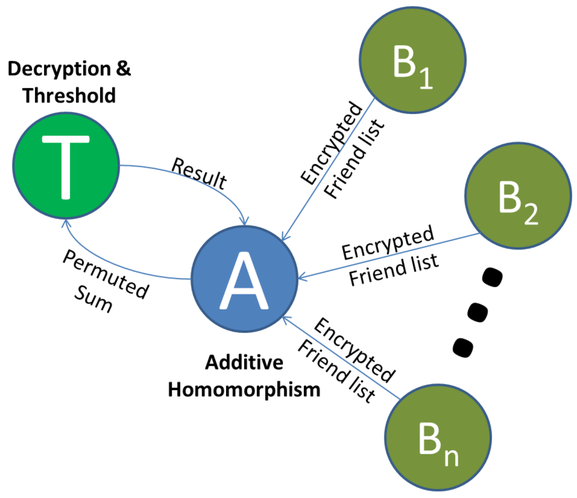
Two Protocols

The protocol $PPFR_h$ utilizes the, **additive homomorphic encryption scheme[1]** to sum up the friend list of all A's friends under encryption.

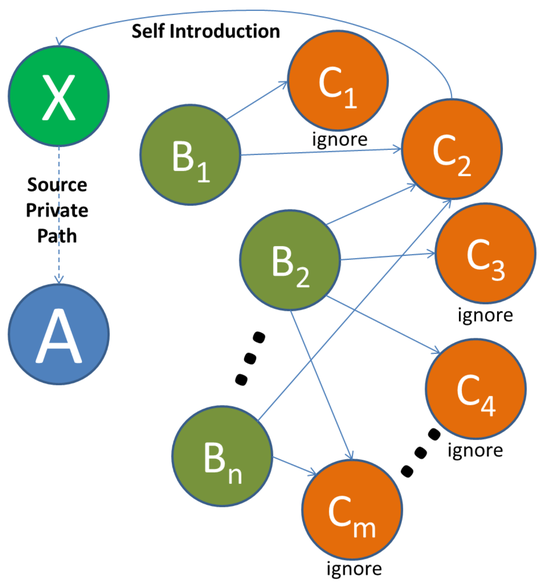
A trusted server T is used to public the encryption key and do the decryption. Only T knows the private key for decryption.

Each B_i encrypts his/her friend list as a vector and send it to A. A sums up all the list, permute the list and send it to T. T decrypts only the numbers with high frequency, and send it back to A. A undo the permutation and get the common neighbor recommendation.

A hash function is used to represent the friend list, resulting in approximate recommendation.



Protocol: $PPFR_h$



The protocol $PPFR_{sp}$ utilizes **protecting the source privacy[2]** technique to let the candidates introduce themselves.

All A's friends B_i will send announcement to their friends C_j together with an randomly chosen message passing path.

C_j can choose different policy to decide whether send an self introduction along the given path.

A will get multiple messages from anonymous users. A can read the self introduction only when he/she get enough number of messages from the same C_j. This is ensured by use **secrete sharing technique[3]**.

Public key system(RSA) is used to ensure source authentication.

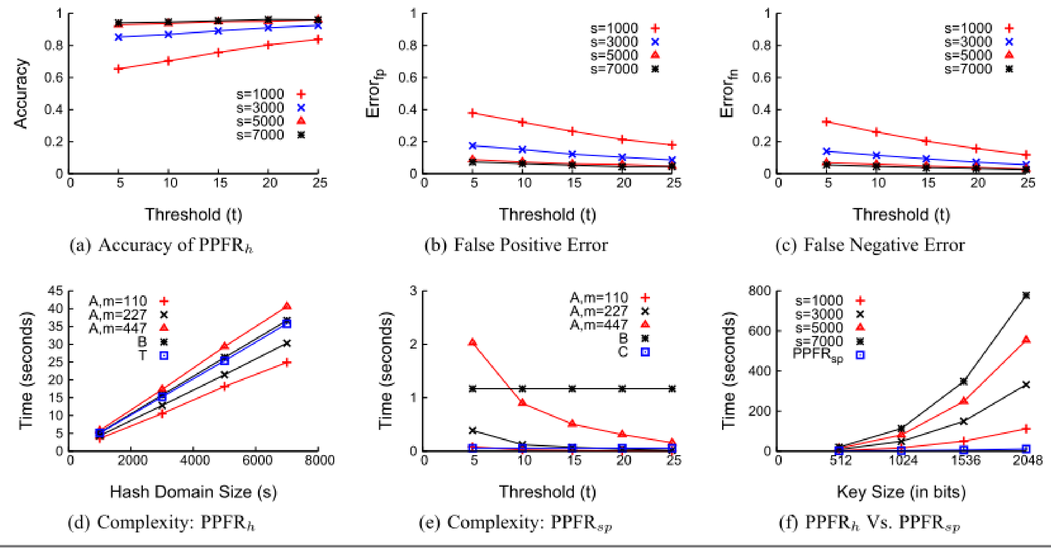
Protocol: $PPFR_{sp}$

Comparison

	$PPFR_h$	$PPFR_{sp}$
Computation Cost	$O(s \cdot \lvert Fr(A)\rvert)$ s is the hash space size.	$O(\lvert Fr(B)\rvert)$
Communication Cost	$O(K \cdot s \cdot \lvert Fr(A)\rvert)$ K is the encryption key size.	$O(k \cdot l)$ k is the size of path. l is the number of (B_i, C_j) pairs.
Security (Semi-honest assumption)	More secure.	Less secure. A gets the exact number of common friends

In all, $PPFR_h$ is more secure under assumption and $PPFR_{sp}$ is more efficient and accurate. Our protocols act as a trade-off among security, efficiency, and accuracy thereby providing more flexibility to the users.

Empirical Results



(a) Accuracy of $PPFR_h$ (b) False Positive Error (c) False Negative Error

(d) Complexity: $PPFR_h$ (e) Complexity: $PPFR_{sp}$ (f) $PPFR_h$ Vs. $PPFR_{sp}$

[1] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th international conference on Theory and application of cryptographic techniques,* Springer-Verlag, 1999.

[2] W. Jiang, L. Si, and J. Li. Protecting source privacy in federated search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[3] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

CERIAS

Discovery Park e-Enterprise Center