Privacy-Preserving P2P Information Sharing Protocol for Mobile Social Networks

Eric Ke Wang, *Member, IACSIT,* Yunming Ye, S. M. Yiu, and L. C. K. Hui

***Abstract*—The wide use of mobile social networks has made it easier for people to share and exchange information with others anywhere, anytime. However, sharing operations of personal data on mobile social networks has put individual privacy under risk in unprecedented ways. We propose a protocol for privacy-preserving P2P information sharing for mobile social networking; because of the limitation of computation and communication for mobile devices, traditional privacy-preserving information sharing for private data sources cannot be applied in mobile social networks. Therefore, we designed a lightweight obfuscation and encryption model to defend against honest-but-curious behavior attack. The protocol is practical for mobile devices and the simulation performance is encouraging.**

***Index Terms*—Privacy-preserve, mobile social network, obfuscation.**

I. Introduction

Mobile social networking is social networking where one or more individuals of similar interests or commonalities or same location, conversing and connecting with one another using the mobile phone. It allows users to share videos, photos, ideas, activities, events, and interests within their moving individual networks.

A recent survey shows that nearly a quarter of mobile users in UK visited a social network via mobile devices; Facebook, the social network with the largest number of users recently announced that a quarter of their users visit their social network sites via a mobile device every month. Another survey [1] reports that traffic on the mobile Web doubled in 2009. In China, the active mobile QQ users have been 16 million [2]. The spread of mobile social networks has made it easier for people to share and exchange information anywhere, anytime. However, the sharing operations of personal data have put individual privacy under risk in unprecedented ways too, many unnoticed privacy leakage issues occur on the information sharing of mobile social networking, for example, popular atomic actions such as 'checking in' at a location reveals a lot of information about the user: their presence, their location, their friends and the current timestamp. Actually people prefer to share information to enjoy easy and funny life without compromising their privacy.

Currently, mobile social networking can be roughly divided into two types: the first is web based social networks being extended for mobile access through mobile browsers and smart phone applications, such as Facebook, twitter and Myspace; the second type is native mobile social networks with dedicated focus on mobile use like mobile communication, location based services, augmented reality requiring mobile devices and technology. These technologies include WIFI, Bluetooth, SMS, WAP, Java, BREW and i-mode. Examples are Cyworld (South Korea, web+mobile) and Tencent QQ (China, web+mobile). In this paper, we mainly target on the second type, especially for peer-to-peer context-aware mobile social networking systems such as SocialAware and WhozThat which can exchange data between devices using short-range wireless technology such as Bluetooth.

Manuscript received November 10, 2012; revised January 24, 2013.

Eric Ke Wang and Yuming Ye are with the Computer Science Department, Harbin Institute of Technology, and Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen, P.R.C. (e-mail: wk_hit@ hitsz.edu.cn, yym@ hitsz.edu.cn).

Xiaofei Xu, is with the Computer Science Departement, Harbin Insitute of Technology, P.R.C. (email: xiaofei@hit.edu.cn).

S. M. Yiu and Lucas C. K. Hui are with the Computer Science Department, the University of Hong Kong, HongKong, China (e-mail:smyiu@cs.hku.hk, hui@cs.hku.hk).

II. Motivation Applications

A. Application I

Phone books intersection sharing. Almost every mobile phone has a phone book inside, but commonly the content of some entries of the phone books are not complete. For example, Tom has Jack's contact information including his mobile phone number, email address. Alice also has Jack's contact information but only including his home address, home telephone number and so on. So if Alice and Tom can join their phone books and share the intersection entries with each other such as Jack's information, it would be more convenient for both of them to get the complete contact information of Jack. However, when joining the phone books, Tom does not want Alice to know or obtain his other contactors' information besides the common entries inside his phonebook, so does Alice.

B. Application II

Media Sharing. People are creating great media everyday: photos, blogs, wish lists, playlists, and more, they would like to share their media contents with others who have the same interests or activities, such as they could share their photos on some place like a famous viewpoint with others who have also some pictures on that place. Or they could share their videos with others on the same topics. When people enjoy information sharing by mobile social networking, however, the privacy concern has become a major obstacle to information sharing. As we all know, when someone exchanges data with others, he does not want his other private information leakage. For instant, suppose he would like to exchange the phone book items which have the same entries (e.g. names) as the other individual; he does not want the rest

DOI: 10.7763/IJCCE.2013.V2.200

items of his phonebook leakage which has not the same entries with the other.

From the applications on mobile social networking, we have found that Privacy-preserving peer-to-peer information sharing protocol is significant for mobile social networking users. Our aim of the protocol is to allow two individuals with mobile devices, each holding a set of inputs, to jointly share the intersection of their inputs without leaking extra private information to each other. The jointly sharing is also called equijoin.

However, many of the existing research on privacy-preserving techniques assume a more powerful adversarial model where nodes may arbitrarily deviate from the underlying protocol to gain as much advantage as possible to reveal other parties' data. As a result, the proposed protocols against powerful adversaries are often too inefficient to be used. Especially, in the mobile social networking context, all the processors of mobile devices are not so efficient to compute cryptographic algorithms; while, most solutions are for database environment which require high computation and communication overhead. Therefore, existing privacy-preserving solutions are not being able to be adopted directly by mobile social networks. Actually, the information sharing among different people for mobile social networking is commonly under the honest-but-curious model [3], that is parties involved are honest (e.g. would not inject false values into their data) but at most curious about other's data. So we can weaken the security requirements for practical efficiency for mobile social networking and then design a new protocol for mobile social networks with more efficiency.

III. Related Works

Since the mobile social networking has been popular for several years, privacy issues have been started to be concerned. Several published papers discuss the privacy issues and problems; however, they mainly research the privacy about identity, location, personal profile and privacy policy in mobile social networking system [4], [5].

Actually, few techniques have been developed for privacy-preserving two-party information sharing for mobile social networking. However, over the past years, for the other environments such as database, the researchers have developed a wide range of privacy-preserving techniques for joining or querying the different data sources without revealing information of any individual privacy data. In order to solve the privacy-preserving equi joining problems from two different data sources, employing a trusted third party (TTP) is a natural solution; this TTP receives all players' inputs, computes the desired function, and return the result [6]. However, the level of trust that must be placed in such a TTP is often inadvisable, undesirable, or even illegal. In order to make many applications secure, researchers have removed the TTP, replacing it with secure multi-party computation protocols for privacy-preserving distributed information sharing such as [7]-[9], but most of these privacy-preserving techniques which realize secure multi-party computations of some specific operation rely on heavy weight cryptographic operations (e.g. zero-knowledge proof of knowledge and homomorphic encryption for each data value) to ensure malicious parties not to deviate from the protocol specification, which is not suitable for mobile social networking. While the other techniques [10]-[15] require multiple rounds of interactions between the individual entities which is also impractical for mobile social networking. The high latencies are a fundamental limiting factor that affects the scalability and applicability for mobile social networking.

IV. Protocol of P2P Data Sharing

A. Assumption

In our work, we mainly solve the privacy problems when two individuals exchange their data and information by their mobile devices. In our solutions, no third party is involved. The main two parties directly execute a protocol, which is designed to guarantee that they both hardly learn any more than they would have learnt from the desired results.

We assume both sides involved have honest-but-curious behaviors. Both sides follow the protocol properly (e.g. would not inject false values into their data) with the exception that they may keep a record of all the intermediate computations and received messages, and analyze the messages to try to learn more information. This behavior is also known as semi-honest or passive behavior.

B. Intersection Protocol

Problem statement (Minimal Data Sharing): Let there be two individuals R and S with private data sources V_S and V_R respectively. Given a query Q spanning the data sources in V_S and V_R, compute the intersection answer to Q and return it to R and S with revealing minimal additional information to either party except for the intersection parts.

For instance, let $S = \{S_1, S_2, \ldots .S_n\}$ be a set of data source, each data has an entry identifier, called *EI*, for example, phone books' entry is names, the media files' entry of photos or video is the directory titles. Suppose $\{ID^S{}_1, ID^S{}_2, \ldots\ldots ID^S{}_n\}_S$ is the entry identifier set of S. Every piece of data *S* has several attributes or files like S_1 attributes $\{P_1, P_2, \ldots .P_n\}$. Let $R = \{R_1, R_2, \ldots\ldots, R_n\}$ be the other set of data source, $\{ID^R{}_1, ID^R{}_2, \ldots\ldots ID^R{}_n\}_R$ is the entry identifier set of R, R_1 attributes are $\{Q_1, Q_2, \ldots .Q_n\}$, the aim is that we need to find out those pieces of data where the entry identifier of S equal the entry identifier of R, then join the attributes or files at the entry identifier and finish the Equijoin process. For example, if $ID^R{}_1 = ID^S{}_1$, then join the attributes of S_1 and R_1, $\{P1, P2, \ldots .P_n\} \cup \{Q_1, Q_2, \ldots .Qn\}$. When joins all the attributes with the same entry identifiers, the process completes.

C. Straightforward but Incorrect Solution

A straightforward idea for computing the intersection $V_S \cap V_R$ would be to use one-way hash functions. Here is a simple protocol that appears to work:

1) Both *S* and *R* apply hash function h to their sets, yielding
2) $X_S = h(V_S) = \{h(v) \mid v \in V_S\}$ and $X_R = h(V_R) = \{h(v) \mid v \in V_R\}$.
3) S sends its hashed set X_S to R.
4) R sets aside all $v \in V_R$ for which $h(v) \in X_S$; these values form the set $V_S \cap V_R$. Unfortunately, *R* can learn a lot more about V_S (with honest-but-curious behavior). For any arbitrary value $v \in V - (V_S \cap V_R)$, R can simply compute $h(v)$ and check whether $h(v) \in X_S$ to determine whether or not $v \in V_S$. In fact, if the domain *V* is small, R can exhaustively go over all possible values and completely learn V_S since R keep the record of all X_S. For example, R can use name generator to simple generate the names he wants to understand and compute the hash to check whether they are inside S. For the photos of viewpoints, R can also use all possible viewpoint names in the city to check whether S has been those places to for pictures.

From the above point, we can understand that it is totally possible for semi-honest player to figure out more information of the other player if he receives all the hashed set of the other player. So the problem lies in the hashed value. Commonly, the hashed set cannot be reversed; however, once all data is hashed and obtained by the other side, it can be compared and checked one by one by brute force checking without limitation of guessing times. Therefore we believe if we can cloak the hashed set, then privacy would not be revealed. Natural solution is employing cryptographic encryption after hash operation; commutative encryption can achieve the goal for the two-party model. However, commutative encryption is not suitable for energy efficient mobile devices because it cost too much computation and communication. Therefore, we need to find out a lightweight way and we adopt obfuscation to obfuscate the set, then it is confused for the player to check whether the current one is inside the other party's real set or noises.

D. Our Protocol

Commonly, the intersection parts are based on intersection names or identities which are called entry identifiers. Our aim is that we need to find out those pieces of data where the entry identifier of A equal the entry identifier of B, then join the attributes or files at the entry identifier and finish the Equijoin process. Our protocol is as follows:

E. Initialize a Protocol

Since no third party is involved, the players should initialize the protocol. Suppose B wants to equijoin the data with A, first, B obfuscates his entry identifiers and hashes the obfuscate entry identifiers, let it to be a set OH, then B sends a request to A to initialize one protocol with the obfuscated entry identifiers hash value OH. To establish a two-party symmetric key, *B* could also send out a set $\{g, m, g^b \bmod m\}$ which is to be used for common key agreement, *b* is the private part of B.

F. Comparison to Find Out the Intersection on A Side

After A receives the request of data sharing, if he does not agree on the information sharing with B, he could send back A "*Data Sharing* Deny" message to B, else he would start to prepare for data sharing. The process is that the program of A's device hashes its items one by one and compares these items with the hashed value OH from *B* to check which items are equal to the value of *B*. And A sends the intersection result of IEI_{ab} to B with a set $\{g, m, g^a \bmod m\}$, where a is the private part of A. Besides, A can calculate a common key by $key = g^{ab} \bmod m$.

G. Comparison to Find out the Intersection on B Side

After B receives the hashed data, B can calculate a common key by $key = g^{ab} \bmod m$, the key would be used to encrypt and decrypt the messages of intersection data result. In this step, B would encrypt the data on the intersection entry identifiers and send it to A.

H. Integration Process on a Side

A decrypts the data from B and find out those data which is not empty and integrates the data, at the same time, A sends out his encrypted data on the same intersection entry identifiers with the same key $\{g^{ab} \bmod m\}$.

I. Integration Process on B Side

After B received the data, B could decrypt it and integrate it with its own data.

V. Security Analysis

The first message transferred from initialize side is hashed and obfuscated. Any adversaries outside can eaves drop the obfuscated message but he cannot figure out the original data. At the same time, we employ key agreement process to establish a key and symmetric encryption to encrypt the intersection data messages. Thus any adversaries outside cannot obtain the intersection data.

An argument is that since in the step 1, after B obfuscates his entry identifiers by adding noises, then in the step 3, it is possible for B to get a result which has more entry identifiers than the real intersection parts since these more entry identifiers maybe intersecting with the noises. Yes, the result is possible to be larger than real intersection items since it may include noises. But it does not affect the final equijoin

result because the noises only are some entry identifiers with empty attributes or data values. Even the integration on those entry identifiers from the noises happens; integration with the empty attributes or data values would not affect the final equijoin results. Here we analyze the potential honest-but-curious attacks for both sides respectively.

A. Honest-but-Curious Attack

1) Suppose party A is *semi-honest*, A could obtained the hashed value from party B in the step *1*, he want to find out if entry identifier x $(x \notin A \cap B)$ is inside party B's set, he can $hash(x)$ and compare the $hash(x)$ with the items of the hashed data sent from B, however, the hashed value is obfuscated by B, then even A tries to brute force guess one by one and get some more extra bingos, he hardly make sure whether these items are from B or B's noises. And the probability of bingo is related to the size of noise. Noise size is bigger, the obfuscation degree is higher.
2) Suppose party B is *semi-honest* and he wants to find out if entry identifier x $(x \notin A \cap B)$ is inside party A's set, he can $hash(x)$ and insert it into the items of the hashed data sent from B as a noise, since the result is the intersection items which could be more than the real intersection parts. However, since the obfuscation is one time operation, that means B has only one chance to insert the noises. So the probability of one time guessing bingo is relative small. We analyze the probability of success rate for this type of attacks as follows.
3) Model analysis

The system achieves privacy cloaking by obfuscation of the entry identifiers of protocol requester. Suppose the passive player A is attacker, according our protocol, the obfuscation degree of B could be configured. For example, if initial side B expects more confusion, then more noises could be added. Suppose that the obfuscation degree is 100%, thus the added noise size is the same as the original data size, named obfuscated set. Then even the other side A compares the hashed value and get some more results, it is still 50% probability to be noise. If the noises are added more, the effective of obfuscation is better.

On the other side, suppose B is attacker, actually, it is a small probability event for B to guess some more information of entry identifier about A; for example, B requests an equijoin of phone books with A. B firstly obfuscates his phone books by adding some noises. Then A can compute and compare with the obfuscated set to check which is intersecting with B's obfuscated set. Therefore, some items which are not intersecting with real *B*'s set could be counted since these items intersect with the noises. However, since B has only one chance to obfuscate his set, suppose the domain space of all related entry identifiers is *n*, and each player has m items, the probability p of guessing k bingos is as following formula:

$$p = C(m, k) / C(n, k)$$

So for a phone books example, if $k = 1$, $n = 8$ Million (all possible names in one medium country), $m = 200$ (each phone book has average *200* items), the probability of guess 1 extra entry identifier is 2.5^{-5}, the probability is relative small, if $k = 2$, the probability of guess 2 extra entry identifiers is smaller than 1.6^{-10}.

From above analysis, we can find that the possibility of one time guessing is relative small. Even if by any chance the lucky guessing happens, the information leakage is also a small piece of privacy (some entry identifiers), the privacy leakage is minimal.

VI. Evaluation

We simulated our protocol in Google Android Emulator; the Android 2.0 SDK supports many cryptographic functions derived from the java package "javax.crypto" and "javax.security" which provide the classes and interfaces for cryptographic applications implementing algorithms for encryption, decryption, or key agreement, such as MD5, SHA, AES, DES, DH key agreement. In the simulation, we set communication channel as Bluetooth. The transmission rate of the second generation of Bluetooth is1Mbps.

We have tested our protocol on 2KB phone books equijoin which include 200 items respectively. We set the intersection parts are 10% of each side data which is the common scenario. So the intersection parts are 0.2KB, the AES encryption time is 2ms, communication round is 2, the total communication size of two rounds is about 13KB, the communication overhead is 110ms for Bluetooth channel, the total time expense for the protocol is 1.38s. The simulation result shows that the cost is acceptable for the common mobile devices.

VII. Conclusion

We propose a protocol for privacy-preserving P2P information sharing for mobile social networking. We designed a lightweight obfuscation model to defend against honest-but-curious behavior attack. The simulation performance is encouraging and in the coming future, we are going to evaluate our protocol on real mobile devices on multiple mobile platforms and communication channels.

Acknowledgment

This research was supported by National Natural Science Foundation of China (No.61100192), Research Fund for the Doctoral Program of Higher Education of China (No.20112302120074), Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (No.HIT.NSFIR.2010129), and supported by National Key Technology R&D Program of MOST China (No. 2012BAK17B08), and was partially supported by Shenzhen Strategic Emerging Industry Development Foundation (No.JCYJ20120613151032592). The authors thank the reviewers for their comments.

REFERENCES

[1] *Mobile Internet Traffic: Analysing Global Usage Trends*, Informa Telecoms and Media, 2010.

[2] Analysis International, *Analysis International: Mobile Internet Created 21.8 Billion Yuan in China 2011Q3*, December 22, 2011.

[3] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," *SIGCOMM Computer Communication Review*, vol. 42, issue 4, 2012.

[4] J. Blasbalg, R. Cooney, and S. Fulton, "Defining and exposing privacy issues with social media," *Journal of Computing Sciences in Colleges*, vol. 28, issue 2, 2012.

[5] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao, "Understanding and capturing people's privacy policies in a mobile social networking application," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 401-412, 2009.

[6] P. P. Tsang, A. Kapadia, and S. W. Smith, "BLAC: Revoking repeatedly misbehaving anonymous users without relying on TTPs," *ACM Transactions on Information and System Security*, vol. 13, no. 4, article 39, December 2010.

[7] M. Yakout, M. J. Atallah, and A. Elmagarmid, "Efficient and practical approach for private record linkage," *Journal of Data and Information Quality (JDIQ)*, vol. 3, issue 3, 2012.

[8] J. Groth, R. Ostrovsky, and A. S. New, "Techniques for noninteractive zero-knowledge," *Journal of the ACM (JACM)*, vol. 59, issue 3, 2012.

[9] T. Tassa and E. Gudes, "Secure distributed computation of anonymized views of shared databases," *Transactions on Database Systems (TODS)*, vol. 37, issue 2, 2012.

[10] L. Xiong, S. Chitti, and L. Liu, "Preserving data privacy in outsourcing data aggregation services," *Transactions on Internet Technology TOIT*, vol. 7, issue 3, 2007.

[11] B. Yang, H. Nakagawa, I. Sato, and J. Sakuma, "Collusion-Resistant privacy-preserving data mining," in *Proc. the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 483-492.

[12] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-Preserving data publishing: A survey of recent developments," *Computing Surveys (CSUR)*, vol. 42, issue 4, no. 14, 2010.

[13] S. J. Stolfo and T. Gene, "Privacy-Preserving sharing of sensitive information," *IEEE Security and Privacy*, vol. 8, issue. 4, pp. 16-17, 2010.

[14] V. Pappas, M. Raykova, B. Vo. S. M. Bellovin, and Tal Malkin, "Private search in the real world," in *Proc. the 27th Annual Computer Security Applications Conference*, 2011, pp. 83-92.

[15] R. Ma, X. Meng, and Z. Wang, "Preserving privacy on the searchable internet," in *Proc. the 13th International Conference on Information Integration and Web-based Applications and Services*, 2011, pp. 238-245.

Eric Ke Wang was born in China, 1980. He received a Ph. D of network security at the department of computer science, the University of Hong Kong in 2009. The major Ph. D study field is network security. He has been a research assistant in French National Institute for Research in Computer Science and Control (INRIA), Nancy，France, 2007. After Ph. D study, he was employed in Harbin Institute of Technology, Shenzhen Graduate School. Current research interests mainly involve network security, mobile security, social computing and cyber physical systems. Dr. Wang is a senior member of IACSIT, and is the member of ACM and CCF.

Yunming Ye was born in China, 1976. He received a Ph. D of data mining at the department of computer science, Shanghai Jiaotong University in 2003. The major research field is data mining. Currently, he is a professor in Shenzhen Graduate School, Harbin Institute of Technology. His research interests include Web mining, Web Search, and social computing. Professor Ye is the member of ACM and CCF.

Xiaofei Xu was born in China, 1962. He received a Ph. D at the department of computer science, Harbin Institute of Technology in 1988. The major research field is computing applications. Currently, he is a chief professor at Computer Science Department in Harbin Institute of Technology. His research interests include service computing, enterprise intelligent computing, database, business intelligent. Professor Xu is the life-member of ACM and a senior member of IEEE.

Siu-Ming Yiu was born in Hong Kong. He received a BSc in Computer Science from the Chinese University of Hong Kong, a MS in Computer and Information Science from Temple University, and a PhD in Computer Science from The University of Hong Kong. He is concurrently an associate professor in the Department of Computer Science, The University of Hong Kong. Before he joined the Department as a faculty member, he has worked as an Analyst Programmer for a couple of years and has been involved in a number of projects (UFIA, FIT) led by Professor Chin. His current research interests include bioinformatics, computer security and cryptography. Dr. Yiu is a senior member of IEEE.

Lucas C. K. Hui was born in HongKong. He received received his BSc and MPhil degrees in computer science from The University of Hong Kong, and his MSc and PhD degrees in computer science from the University of California, Davis. He is the founder and Honorary Director of the Center for Information Security and Cryptography, and concurrently an associate professor in the Department of Computer Science, The University of Hong Kong. Besides actively publishing research papers, he is also involved in consultation work in security systems, and in industrial collaboration projects. His research interests include information security, authentication services, network security, cryptography, and electronic commerce. Dr. Hui is a senior member of IEEE.