ICIC Express Letters
Part B: Applications
Volume **7**, Number **4**, April **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **839–845**

A PERSONALIZED SOCIAL NETWORK PRIVACY PROTECTION METHOD BASED ON THE PREGEL-LIKE SYSTEM

Xiaolin Zhang[1,*], Yanlei Guo[1], Jingyu Wang[1,2]
Chen Zhang[1] and Wenchao Zhang[2]

[1]School of Information Engineering
Inner Mongolia University of Science and Technology
No. 7, Inner Mongolia Arding Street, Baotou 014010, P. R. China
[*]Corresponding author: zhangxl@imust.cn

[2]College of Computer and Communication Engineering
University of Science and Technology Beijing
No. 30, Xueyuan Road, Haidian District, Beijing 100083, P. R. China

Received October 2015; accepted January 2016

Abstract. *Current social network privacy protection methods mainly assume that attackers have the same background knowledge, an assumption that cannot satisfy the needs of privacy protection for personalized users. What is more, during the processing of a tremendous amount of social network data, the weaknesses of poor privacy protection and low capacity of data processing were demonstrated. As such, this article proposes a personalized social network privacy protection method based on the Pregel-like system. This method provides different types of privacy protection for different vertices, and adopts the "Think Like a Vertex" concept in the Pregel-like system. It realizes the processing of privacy protection through message communication between the vertex and multiple iterations of the program. The experiment shows that the proposed method can provide personalized privacy protection for diverse users while increasing the efficiency of privacy protection. In addition, it guarantees the serviceability of the released data.*
Keywords: Pregel-like, Social network, Personalized privacy protection, Tremendous amount of data, Label list anonymization

1. **Introduction.** Recent research on the privacy protection of social networks has achieved considerable progress. Researchers have proposed various methods of privacy protection according to the different problems related with social network privacy protection [1]. However, those methods were based on the specific background knowledge of the attackers and did not consider the different needs of different users for personalized privacy protection. What is more, with the continual increase of social network data, the interactive relation between users has become more complex. [2] used a bipartite graph as a new expression of a graphic structure to demonstrate the interactive relation between users through two types of substances. [3] proposed the protection of a bipartite graph network with a (k, m)-label list anonymization algorithm. The algorithm divides the node security into groups through the greed strategy and generates a label list with the size of k for every node. [4,5] first studied the methods of node privacy protection in the cloud environment, although the methods were aimed at cloud service regarding released data. However, the processing prior to data release was carried out by a single machine.

The emergence of cloud computing and distributed parallel technology has popularized the research on the processing of large-scale social network graphic data. Large-scale graphic data processing models are basically classified into [6]: 1) improvement and realization based on the MapReduce model, and 2) realization based on the Bulk Synchronous Parallel (BSP) model. [7] introduced the Pregel system used by Google to process large-scale graphic data on the basis of the BSP model realization. The Pregel system has

various advantages [8] in processing large-scale graphic data under a cloud computing environment. The realization and application of the Pregel system have triggered the development of a series of Pregel-like systems, including Apache's Hama, Apache's Giraph, Mizan [9], GPS [1], and the Graphx of Spark. The use of the Pregel framework can realize various graphic iterative algorithms [7].

This article shows how highly efficient and fast privacy protection can be provided for large-scale social network under the distributed environment and how the serviceability of released data can be ensured under the premise of satisfying the needs for privacy protection. The organizational structure of the article is as follows: Section 2 introduces the problem statement and preliminaries, Section 3 introduces the personalized privacy protection method based on the Pregel-like system, Section 4 conducts the experimental evaluation of the proposed algorithm, and finally, Section 5 provides the conclusions.

2. Problem Statement and Preliminaries.

This article uses a personalized bipartite graph to express social networks. The personalized bipartite graph is defined as follows.

Definition 2.1. *(Personalized bipartite graph) Personalized bipartite graph G can be expressed by using a sextuple: $G = \{V, V', I, I', E, X\}$. V represents the conservative user cluster $V = \{v_1, v_2, \ldots, v_n\}$, V' represents the open user cluster $V' = \{v_1', v_2', \ldots, v_n'\}$, I represents the conservative interaction cluster $I = \{i_1, i_2, \ldots, i_s\}$, I' represents the open interaction cluster $I' = \{i_1', i_2', \ldots, i_s'\}$, E represents the edge cluster $E \subseteq (V \cup V') \times (I \cup I')$, and X represents the property entity cluster of the corresponding vertex. Supposing $v \in (V \cup V')$ and $x_v \in X$, then $x(v)$ represents the identification (or label) of v.*

Figure 1 is an example of an online social network sub-graph. Figure 1(a) and Figure 1(b) are the user information list and interaction information list, in which u3 is the open user, and the interaction game is the open interaction. A personalized bipartite graph is shown in Figure 1(c).

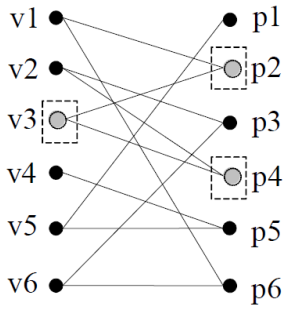
To prevent the disclosure of interaction privacy between individuals and prevent the vertex from attacks of re-identification during the release of a social network, bipartite graph anonymization can be realized through the "label list" generated for every node.

Id	User	Age	Sex	State
v1	u1	29	F	NY
v2	u2	20	M	JP
v1'	u3	24	F	UK
v4	u4	31	M	NJ
v5	u5	18	M	NJ
v6	u6	21	F	CA

(a) User table

Id	Interaction	Information
p1	email1	1024 bytes on 1/5/14
p1'	game1	score 9-7-8
p3	email2	812 bytes on 3/4/14
p2'	game2	score 8-3-6
p5	blog1	subscribed on 9/9/14
p6	friend1	added on 7/6/14

(b) Interaction table



(c) Bipartite graph

FIGURE 1. Data examples

Definition 2.2. *(Personalized label list anonymization) The output of a personalized label list anonymization is a bipartite graph G' on V and I that is isomorphic to the original graph G, The graph G' has a function l from V to $\varphi(X)$ (the powerset of X), so that l(v) is the personalized anonymization label cluster (or label list) of v. We insist that the true lable of v is included in its list, so that $x(v) \in l(v)$.*

However, the formation of label lists with randomly selected identifications will reveal the privacy relations between users. Therefore, creating candidate identifications for the vertex according to the security grouping conditions is necessary.

Definition 2.3. *(Security grouping conditions) Group V safely satisfies the security grouping conditions. With any node $v \in V$, v has an interactive relation with a node in Group $S \subset V$ at the maximum. That is $\forall e(v,i), e(w,i), e(v,j), e(z,j) \in E : w \in S \wedge z \in S \Rightarrow z = w$.*

These conditions can create the (k, m)-label list for all user vertex after security grouping. m represents the size of the group, which is the range of anonymization, and k represents the size of anonymization, which is the strength of anonymization of real vertex.

Definition 2.4. *((k, m)-label list) Suppose the size of C_j is the group of m, $p = \{p_0, p_1, \ldots, p_{k-1}\}$, which is an integer sequence. p is the sub-cluster with the size k in the cluster $\{0, 1, \ldots, m\}$. The node label list in C_j is generated through the following equation, in which $0 \leq i < m$:*

$$list(p, i) = \left\{u_{(i+p_0) \bmod m}, u_{(i+p_1) \bmod m}, \ldots, u_{(i+p_{k-1}) \bmod m}\right\}$$

The list generated when at $p = \{0, 1, 2, \ldots, k-1\}$ is the prefix list; when at $k = m$, the list generated is the complete list. In Figure 1, suppose the results after security grouping when at $m = 3$ are $\{\{\text{v1}, \text{v2}, \text{v4}\}$ and $\{\text{v5}, \text{v6}\}\}$. When $k = 2$ and $p = \{0, 1\}$, the obtained prefix lists of u1, u2, and u4 are $\{\text{u1}, \text{u2}\}$, $\{\text{u2}, \text{u4}\}$, and $\{\text{u1}, \text{u4}\}$, respectively. When at $k = m = 3$, the obtained complete lists of u1, u2, and u4 are $\{\text{v1}, \text{v2}, \text{v4}\}$, $\{\text{v1}, \text{v2}, \text{v4}\}$, and $\{\text{v1}, \text{v2}, \text{v4}\}$, respectively.

3. Personalized Privacy Protection Based on the Pregel-like System.

The basic thought of this method is that under the distribution environment, the bipartite graph vertices are grouped in the Worker tasks of different computing vertices by the Pregel-like system. The initialization state has also been set for every node. In every superstep, the vertices in the Active state receive and send messages. These vertices compare the values of messages and their own values according to the *compute()* method to determine if they should put the current vertex in the groups. The vertex will enter the Inactive state if yes. They will continue to send and receive messages until all vertices are in the Inactive state.

To increase operating efficiency, the vertex must first be sequenced by the degree before the personalized security grouping. Then, the user vertex must be renumbered. The basic steps of the *compute()* method of personalized security grouping are as follows.

Step 1 All conservative users on the left at the initialization state are in the Active state. All open users and the interaction vertex on the right are in the Inactive state.

Step 2 When at $superstep = 0$, the vertex in the Active state will send their number values to their adjacent vertex.

Step 3 When at $superstep\%2 = 1$, the user vertices on the left are set as Inactive, and the interaction vertices are in the Active state. The smallest node numbers are obtained from the messages. The values are returned to user vertex.

Step 4 When at $superstep\%2 = 0$, the interaction node is set as Active, and the user node is in the Active state. The value is compared with its own value. It will then put

itself in the current group if they are equal to each other. The state will be transferred to Inactive, or the state will remain unchanged if they are not equal to each other.

Step 5 Repeat **Step 3** and **Step 4** until all vertices are in the Inactive state.

Using the bipartite graph in Figure 1, the user vertices have been numbered according to the degree sequence, as shown in Figure 2. After three supersteps, the security grouping results are {1, 2, 4} and {5, 6}, and the corresponding user vertices are {u1, u2, u5} and {u4, u6}.

When the security grouping has been completed, personalized label list anonymization can be conducted for the target vertex according to the grouping and $p = \{p_0, p_1, \ldots, p_{k-1}\}$. As the range parameter m has not been considered in the process of grouping. Additionally, conducting a simple adjustment of the groups obtained according to the parameter m is only necessary when the execution of security grouping has been completed. The basic steps of the *compute()* method of personalized label list anonymization are provided below.

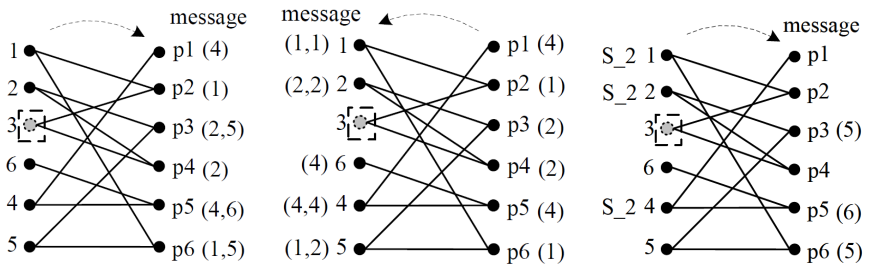
Step 1 Building pseudo vertex according to the number of security groups. The value of the pseudo vertices is the group, and the adjacent vertices are all vertices in the group.

Step 2 All conservative user vertices at initialization are in the Inactive state, and the pseudo vertices are in the Active state.

Step 3 When at *superstep* = 0, the pseudo vertices send the grouping information to the adjacent user vertices.

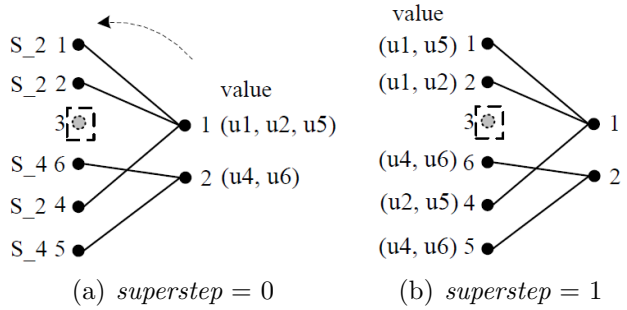
Step 4 When at *superstep* = 1, the pseudo vertices are set as Inactive, and the user vertices are in the Active state. Having received the information of their own group, the user vertices will create an anonymization label list for themselves according to sequence $p = \{p_0, p_1, \ldots, p_{k-1}\}$ and will correct the current node values. As such, the algorithm is stopped.

The personalized label list anonymization based on the Pregel-like only needs two supersteps. Suppose the anonymization sequence $p = \{0, 2\}$, the two supersteps are shown in Figure 3. As seen in the figure, the u1 anonymization label list is $\{u1, u5\}$, the u2 anonymization label list is $\{u1, u2\}$, and so on.



(a) *superstep* = 0 (b) *superstep* = 1 (c) *superstep* = 2

FIGURE 2. Node message communication processes



(a) *superstep* = 0 (b) *superstep* = 1

FIGURE 3. Process of node label list anonymization

4. **Experimental Evaluation.** The experiment was operated on a Hadoop cluster consisting of five servers and a single unit with Windows7 system. The Hadoop cluster is installed with Hama-0.6.4, Giraph-1.1.0, and Spark-1.2.1. The data used for this experiment are the real dblp thesis data. As of March 2015, the data included 4,430,580 papers and 1,504,237 authors. To satisfy the needs of the experiment, we randomly divided the original data clusters into five parts and reintegrated the data according to the rate of 1:2:3:4:5.

To analyze the processing time, some open users were created from the user vertex at the rate of 5% in the integrated data cluster. Then, the threshold value was set to $k = m = 5$. Figure 4 shows a comparison chart of processing times of the traditional anonymization method. From the chart, we can see that in processing small-scale data clusters, the operating time of the traditional method was close to the proposed method based on the Pregel-like system. However, the traditional methods took a longer time as the data scale increased.

To analyze the data serviceability, the experiment evaluated the difference of query results between the anonymization data and the original data. The proposed query S is as follows: "In the 10 years between 2001 and 2010, how many Chinese authors have published articles on the periodical Wireless Networks?" The query is conducted on Split_4. The experimental result is shown in Figure 5. The query results of prefix list anonymization and the complete list anonymization are close to the queries of the original data cluster.

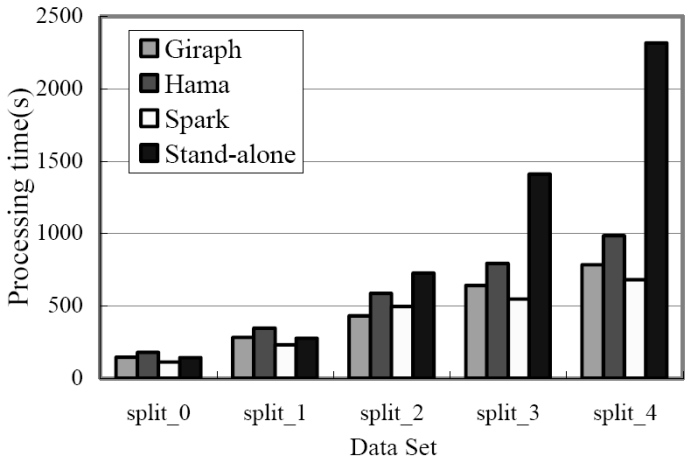


FIGURE 4. Comparison of processing times

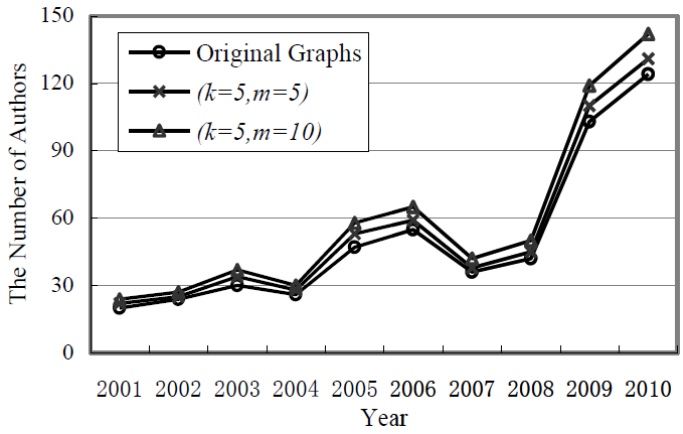


FIGURE 5. Query difference before and after

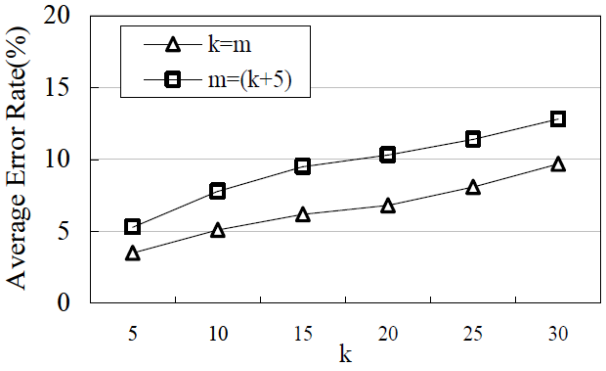


FIGURE 6. When k and m remain changed

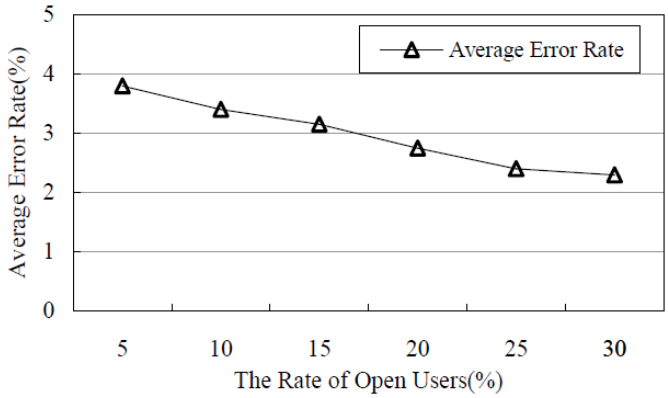


FIGURE 7. Different rates of open users

To further verify the influence of k and m on the serviceability of data, 20 aggregate queries were proposed. These queries covered the three types of queries in [2]. The average query error rate is shown in Figure 6. When $k = m$, the average query error rate of data will become larger with the increase of m, while the serviceability will be lower. When k remains unchanged, the average query error rate will become larger with the increase of m.

To study the influence of the number of personalized users on the serviceability of data, the experiment created some open users according to different rates in the Split_4 data cluster. Twenty aggregate queries were conducted for the anonymously processed ($k = m = 5$) data. As seen in Figure 7, personalized privacy protection can satisfy the needs of users and increase the serviceability of data to a certain extent after anonymization.

5. **Conclusions.** This article proposed a personalized social network privacy protection method based on the Pregel-like system. The proposed method provides different types of privacy protection for different vertices, and adopts the "Think Like a Vertex" concept in the Pregel-like system. The experiment shows that it can increase the efficiency in massive social network privacy protection and provides personalized privacy protection for diverse users. Furthermore, it guarantees the serviceability of released data. In the future, other models may be needed if we wish to anonymize both users and interactions between them.

Acknowledgment. This work is partially supported by Natural Science Foundation of China (Nos. 61562065, 61462069). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. Y. Liu, B. Wang and X. C. Yang, Survey on privacy preserving techniques for publishing social network data, *Journal of Software*, vol.25, no.3, pp.576-590, 2014.

[2] G. Cormode, D. Srivastava and T. Yu, Anonymizing bipartite graph data using safe groupings, *Proc. of the VLDB Endowment*, vol.1, no.1, pp.833-844, 2008.

[3] S. Bhagat, G. Cormode and B. Krishnamurthy, Class-based graph anonymization for social network data, *Proc. of the VLDB Endowment*, vol.2, no.1, pp.766-777, 2009.

[4] J. Gao, J. X. Yu and R. Jin, Neighborhood-privacy protected shortest distance computing in cloud, *Proc. of the ACM SIGMOD International Conference on Management of Data*, New York, pp.409-420, 2011.

[5] G. Wang, Q. Liu and F. Li, Outsourcing privacy-preserving social networks to a cloud, *Proc. of INFOCOM*, Piscataway, NJ, pp.2886-2894, 2013.

[6] G. Yu, Y. Gu and Y. Bao, Large scale graph data processing on cloud computing environments, *Chinese Journal of Computers*, vol.34, no.10, pp.1753-1767, 2011.

[7] G. Malewicz, M. H. Austern and A. J. C. Bik, Pregel: A system for large-scale graph processing, *Proc. of the ACM SIGMOD International Conference on Management of Data*, New York, pp.135-146, 2010.

[8] S. Salihoglu and J. Widom, Optimizing graph algorithms on pregel-like systems, *Proc. of the VLDB Endowment*, vol.7, no.7, pp.577-588, 2014.

[9] Z. Khayyat, K. Awara and A. Alonazi, Mizan: A system for dynamic load balancing in large-scale graph processing, *Proc. of the 8th ACM European Conference on Computer Systems*, New York, pp.169-182, 2013.

[10] S. Salihoglu and J. Widom, GPS: A graph processing system, *Proc. of the 25th International Conference on Scientific and Statistical Database Management*, New York, pp.22-23, 2013.