CHAPTER

11

An Introduction to Various Privacy Models

X. Lu, M.H. Au

The Hong Kong Polytechnic University, Kowloon, Hong Kong

1 INTRODUCTION

Anonymity refers to the absence of identifying information of an individual. In the digital age, user anonymity is critically important since computers could be used to infer individuals' lifestyles, habits, whereabouts, and associations from data collected in different daily transactions (Chaum, 1985). However, merely removing explicit identifiers may not provide sufficient protection. The preliminary reason is that the released information, when combined with publicly available information, can also reveal the identity of an individual. A famous example is the Netflix crowdsourcing competition. In 2012, Netflix released a data set of users and their movie ratings. People could download the data and search for patterns. The data contained a fake customer ID, together with movie, customer's rating of the movie and the date of the rating. It is claimed that since customer identifiers have been removed, the released information would not breach user privacy. However, Narayanan and Shmatikov (2008) showed how customers can be identified when the dataset from Netflix is combined with some auxiliary data (such as data from IMDB).

Location privacy is also of great concern in the mobile setting. Here we briefly review a case related to the location privacy of a location-based social network (LBSN), namely, WeChat, as discussed in (Wang et al., 2015). By using a fake GPS position and mobile phone emulation, it is possible to reveal the exact location of any WeChat user with the nearby service turned on (Fig. 1).

The previous example raises a question: *what kind of information do we wish to protect when we talk about privacy protection?* In other words, how do we define privacy? Traditional models in dealing with data confidentiality are not applicable in this case, since we have to maintain data utility. In the Netflix competition example, the data set is released to the public for mining, while in WeChat Nearby service, the user should be able to obtain the list of users nearby.

Over the years, the research community has developed various privacy models, including *k*-anonymity (Sweeney, 2002) and differential privacy (Dwork, 2006). In this chapter, we discuss these definitions and implications and the techniques to achieve them.

http://dx.doi.org/10.1016/B978-0-12-804629-6.00011-0

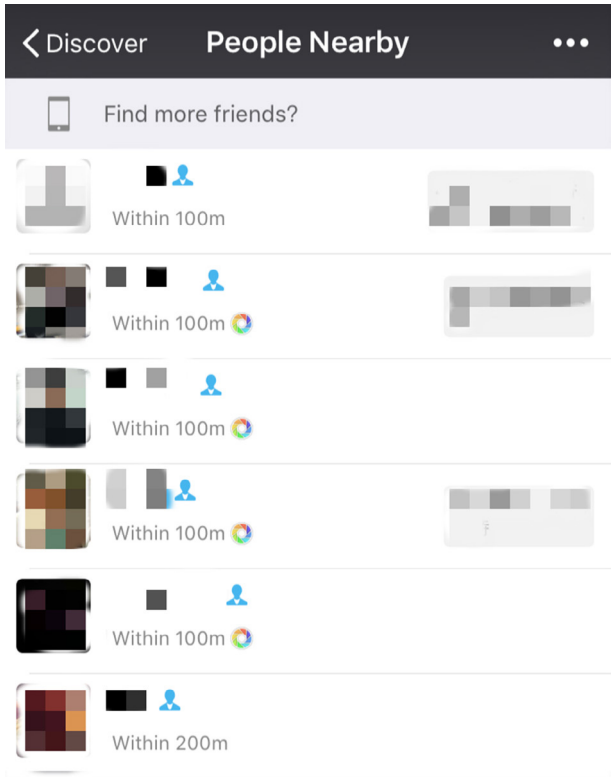

FIG. 1 WeChat nearby people.

1.1 Organizations

This rest of this chapter is organized as follows. In Section 2, we present the definition of *k*-anonymity and discuss its practical implications. In Section 3, we discuss various techniques to achieve the definition. In Section 4, we discuss differential privacy, including its definition and implications. A differentially private mechanism that helps supporting differential privacy is reviewed in Section 5. We conclude in Section 6.

2 DEFINITION OF *k*-ANONYMITY

k-anonymity, proposed by Sweeney (2002), is a property of protecting released data from reidentification. It can be used, for example, when a private corporation such as a bank wants to release a version of data concerning clients' financial information to

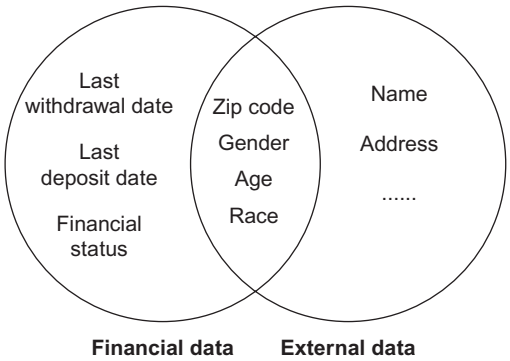

FIG. 2 Linking attack between released data and external data.

some public organizations for research purpose. Under this circumstance, released data should have the property that individual subjects of the data cannot be reidentified so as to protect their privacy. In other words, all the records in the released database should remain unlinkable to the clients. Clients' original data from a bank usually contains information such as name, address, and telephone number that can directly identify clients. One possible way to hide the identity is by directly removing the sensitive information from the database. However, it cannot guarantee clients' privacy. Information like zip code, gender, age and race, clients' identities still can be reidentified. Zip code provides an approximate location. Through searching by specific age, gender, and race, it is still possible to reveal clients' identities. Another possible way to achieve reidentification is called linking attack. Apart from attributes like name and address which can directly break the anonymity of data, there are also attributes called quasi-identifier (QID) which is used to link released data to external data. Gender, age, race and zip code is a typical tuple of QIDs and this tuple of QIDs from released data has high probability that also appears in some external data. If there are external tables like voter registration lists, then by linking the QIDs from released data to voter data, clients' identities may be revealed (Fig. 2).

k-anonymity requires that in the released data, each record can be mapped to at least k records in the original data. In another words, each record from the released data will have at least $k - 1$ identical records in the same released data. For example in Table 1, (a) is the original data and (b) is the data derived from (a). (b) has k-anonymity where $k = 2$. In Sweeney (2002), Latanya Sweeney presented the principle of k-anonymity and proved that if the released data owns the property of k-anonymity, then the linking attack which links the released data to other external data and tries to break the data anonymity can be defended. Intuitively, this is because each record in released data will have at least $k - 1$ same records.

TABLE 1 Example of k-Anonymity ($k = 2$)

(a) Original Data				
Name	**Gender**	**Race**	**Age**	**Zip Code**
Alice	Female	White	17	21103
Lucy	Female	Asian	22	21300
Daniel	Male	Black	27	21110
Kate	Female	White	15	21102
Rose	Female	Black	29	21109
Andy	Male	Asian	24	21304

(b) Sharing Data Derived From (a)			
Gender	**Race**	**Age**	**Zip Code**
F or M	White	15–19	211*
F or M	Asian	20–24	213*
F or M	Black	25–29	211*
F or M	White	15–19	211*
F or M	Black	25–29	211*
F or M	Asian	20–24	213*

3 MECHANISMS THAT SUPPORT k-ANONYMITY

After k-anonymity was proposed, various attempts had been made in designing a good algorithm that turns a database into a form that satisfied this definition. The main two techniques used to enforce k-anonymity in released data are generalization and suppression. Generalization consists of replacing attributes considered to be QIDs with a more general value. In Table 1, the values of gender, age, and zip code from (a) are all substituted by a generalized version in (b). Generalization can be applied in levels from a single cell to a tuple of attributes to achieve k-anonymity. Suppression consists of removing sensitive attributes to reduce the amount of generalization when achieving k-anonymity. Same as generalization, suppression can also be applied in cells or whole attributes. The combination of generalization and suppression has been used to construct different algorithms to help data satisfy k-anonymity. The conventional framework of such an algorithm always starts by suppressing several sensitive attributes and then partitions tuples of remaining attributes into groups and substituting accurate QIDs' values into generalized ones for each group, which are also called equivalent classes. This kind of generalization is homogeneous generalization and has been used to address k-anonymity in Iwuchukwu and Naughton (2007), Ghinita et al. (2007), and LeFevre et al. (2008). A property of homogeneous generalization is that if an original record t_i matches the released record t'_j whose corresponding original record is t_j, then t_j also matches t'_i. This property is called reciprocity. The most significant point for homogeneous generalization is how to divide the equivalent classes. The partitioning strategy will directly influence

the utility of released data. There are two ways to do the partitioning job: global recording (full-domain anonymization) (LeFevre et al., 2005, 2006; El Emam et al., 2009) and local recording (Xu et al., 2006; Aggarwal et al., 2010). Global recording means that within a column, the same generalization strategy is applied to the equal value. So if two tuples in the original data have identical QID values, then they must have the same released value. However, in local recording, two tuples with identical QID values may have different generalized values. Incognito algorithm proposed in LeFevre et al. (2005) uses dynamic programming and is shown to be outperformed by previous algorithms on two real-life databases. The main idea of Incognito is that any subset of the tuple of QIDs with k-anonymity should also have the property of k-anonymity. Mondrian algorithm presented in LeFevre et al. (2006) uses a strategy called multidimensional global recording. In Mondrian, each attribute in the dataset represents a dimension and each record represents a point in the space. Instead of partitioning each records, Mondrian algorithm partitions the space into several regions and in each region, there are at least k points.

Algorithms using local recording may guarantee more anonymity in specific situation (Ninghui Li and Su, 2011).

Another generalization method is called nonhomogeneous generalization (Wong et al., 2010; Xue et al., 2012; Doka et al., 2015). For nonhomogeneous generalization, the property of reciprocity does not necessarily hold for all records. In Table 2, (b) is the released data derived from (a) using homogeneous generalization, and it is clear that (t'_1, t'_2, t'_5) is an equivalent class and (t'_3, t'_4) is another . In an equivalent class, all the generalized QID values are the same. However, in a nonhomogeneous generalized table (c), t'_1, t'_2 and t'_5 have different

TABLE 2 Example of k-Anonymity (k = 2) From Homogeneous and Nonhomogeneous Generalization

(a) Original Data			
Tuple ID	**Gender**	**Age**	**Zip Code**
t_1	Female	17	21103
t_2	Male	29	21110
t_3	Male	27	21210
t_4	Male	15	21202
t_5	Female	22	21109
(b) Sharing Data Generated by Homogeneous Generalization			
Tuple ID	**Gender**	**Age**	**Zip Code**
t'_1	F or M	17–29	211*
t'_2	F or M	17–29	211*
t'_3	Male	15–27	212*
t'_4	Male	15–27	212*
t'_5	F or M	17–29	211*

Continued

TABLE 2 Example of k-Anonymity ($k = 2$) From Homogeneous and Nonhomogeneous Generalization—cont'd

(c) Sharing Data Generated by Nonhomogeneous Generalization			
Tuple ID	**Gender**	**Age**	**Zip Code**
t'_1	Female	17–22	2110*
t'_2	Male	22–29	211*
t'_3	Male	15–27	212*
t'_4	Male	15–27	212*
t'_5	F or M	17–29	211*

generalized QID values. While both table (b) and (c) have 2-anonymity, (c) offers higher data utility since the generalized QID ranges in (c) is either smaller or equivalent to the corresponding ones in (b). This illustrates that by using nonhomogeneous generalization, one may achieve a higher data utility on the released data.

In Wong et al.'s work (Wong et al., 2010), original data and released data are seen as a graph and records from data are vertices. To achieve k-anonymity, each vertex from the graph should have exactly k matches in the same graph including the vertex itself. If we consider a matching between two vertices as an edge, then the former sentence can be rewritten as each vertex in the graph should have out degree and in degree k. So in such graph, there are k disjoint assignments can be extracted and each assignment represents a correspondence between vertices. Even though Wong et al.'s work use nonhomogeneous generalization, there is still the requirement that the generalized graph should form a ring in their strategy which causes redundancy.

Recently Doka et al. (2015) proposed a new algorithm called freeform generalization to implement k-anonymity in a nonhomogeneous way. They defined the problem as how to obtain high data utility in k-anonymity and wanted to solve this problem as an assignment problem in a bipartite graph that has two parts, namely, original and released. Each vertex from original part should have exactly k matches in the released part, and each vertex in the released part should also have k matches in the original part. Doka et al. (2015) proposed an approach to constructing the bipartite graph which contains k disjoint components. To construct such graph, the idea is choosing k different perfect matchings from all the possible matchings including the self-matching from original data to released data for vertices. After choosing, each vertex in the released graph should have k possible identities. The construction is secure since each disjoint assignment has the same probability $1/k$ to be the true one for an adversary. So, each time the adversary wants to find the identities of the released records, he/she will have k possible results. In the construction, each edge between two vertices will be assigned a weight based on Global Certainty Penalty (GCP). GCP is used to measure the information loss of matching an original record to a released record. The released data should keep k-anonymity and data utility. So when choosing the k perfect matchings, the total GCP should be kept as small as possible. Finally, a greedy algorithm was presented in Doka et al. (2015). The input to the greedy algorithm is a weighted completed bipartite graph $G = (S, T, E)$, and the output is a perfect match with a total weight close to the minimum. S represents vertices in original data and T represents in released data. A successful running of the algorithm is called an iteration. In each

iteration, the algorithm tries to find perfect matching from S to T with a low total weight. And the self-matching from original data to released data with zero GCP will be found out in the first iteration. After one iteration, all the selecting edges will be removed from the bipartite graph and all the weights (GCP) on the edges will be redefined. After k iterations, k disjoint perfect matchings with low GCP will be presented. The algorithm can be used in the real word for a practical value k and the complexity for all k iterations is $O(kn^2)$, where n is the number of records in the original data.

4 DIFFERENTIAL PRIVACY

Since the introduction of k-anonymity, weaknesses of it as a model have been discussed, and these weaknesses lead to the proposal of stronger models including ℓ-diversity (Machanavajjhala et al., 2007), t-closeness (Li et al., 2007), or β-likeness (Cao and Karras, 2012). In this chapter, we do not go into details of these definitions and refer interested readers to the respective papers. Informally speaking, the main weakness in k-anonymity is that it does not guarantee proper protection of the sensitive attributes. For example, from Table 1(b), an adversary can safely conclude that if a target user is of age from 20-29 living in a place with zip code starting with 211, the target user is an African American with high probability. Since in the table, only Asians and African Americans are of age from 20-29 and all the Asians' zip codes start with 213.

4.1 Overview

Differential privacy, introduced by Dwork (2006), is an attempt to define privacy from a different perspective. This seminal work consider the situation of privacy-preserving data mining in which there is a trusted curator who holds a private database D. The curator responses to queries issued by data analysts. Differential privacy guarantees that the query results are indistinguishable for two databases that differ only in one entry. From an individual point of view, it means that inclusion of one's information in the private database D would not cause noticeable changes in the observed query outcome; thus, privacy is protected. This is made possible via adding noise to the query result. The setting is shown in Fig. 3:

Note that it is possible to create a synthetic database by issuing a query that output the private database D, as discussed in Chen et al. (2011). However, as pointed out in Clifton and Tassa (2013), the utility of this synthetic database maybe too low for it to be useful.

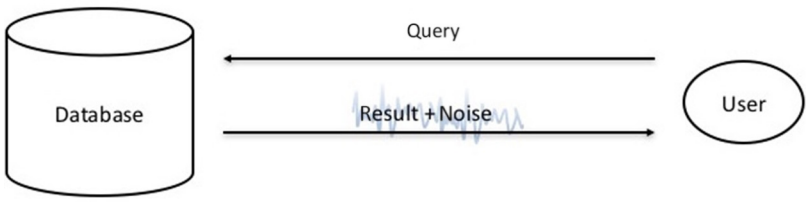

FIG. 3 Privacy-preserving data mining.

4.2 Definition of Differential Privacy

Now we can recap the definition of differential privacy (Dwork, 2006). We first establish the notation. Let $\mathcal{M}: \mathrm{D} \to \mathrm{R}$ be a randomized algorithm with domain D and range R. In concrete terms, we can think of $\mathcal{M}$ as a mechanism that answers a query to a database. Then we can formally define whether or not $\mathcal{M}$ provides differential privacy as follows.

Definition 1. A randomized algorithm $\mathcal{M}$ is ϵ-differentially private if for all possible subrange of $\mathcal{M}$, say $S \subset \mathrm{R}$, and for all databases $D_1, D_2 \in \mathrm{D}$ that differs by only one record, the probability that $\mathcal{M}$ gives the same output on input D_1 and D_2 with similar probability. More formally,

$$Pr(\mathcal{M}(D_1) \in S) \leq e^{\epsilon} Pr(\mathcal{M}(D_2) \in S).$$

Here ϵ controls how much information is leaked. For a small ϵ, the answer given by mechanism $\mathcal{M}$ on two databases that differ by one record is very likely to be the same. In other words, whether or not an individual's information is included in the database would not affect the outcome of the query significantly.

Example. Suppose the query we would like to make is whether or not Alice is a smoker. Consider mechanism $\mathcal{M}$ defined as follows. $\mathcal{M}$ first flips a fair coin $b \in \{0,1\}$. If $b = 0$, return the true answer. Otherwise, flip another coin $b' = \{0,1\}$. If $b' = 0$, return "yes," otherwise return "no." Now that there are two possible databases, namely, Alice is a smoker or Alice is not a smoker. If Alice is a smoker, $\mathcal{M}$ output "yes" with probability 3/4 and "no" with probability 1/4. If Alice is not a smoker, $\mathcal{M}$ output "yes" with probability 1/4 and "no" with probability 3/4. For any possible outcome, namely, "yes," or "no," the probability difference is at most three times. In other word, $\mathcal{M}$ is (ln 3) differentially private.

Remarks. Perhaps one of the most useful properties of this definition is that differential privacy holds during composition. Suppose we have a database D. The data owner releases the query result $\mathcal{M}_1(D)$. Later, he releases another query result $\mathcal{M}_2(D)$. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are ϵ_1 and ϵ_2 differentially private, the outcome of releasing both $\mathcal{M}_1(D)$ and $\mathcal{M}_2(D)$ is $(\epsilon_1 + \epsilon_2)$ differentially private.

5 LAPLACE MECHANISM TO ACHIEVE DIFFERENTIAL PRIVACY

In general, the more noise we add, the more privacy we can guarantee. However, one should bear in mind that one usually aim to get as little noise as possible so as to maintain data utility. For query that returns real numbers as response, the Laplace mechanism is one of the basic mechanisms to provide differential privacy. We first recall the definition of Laplace distribution (Dwork and Roth, 2014).

Definition 2. The Laplace distribution with constant b is defined by the probability density function:

$$Lap(x \mid b) = \frac{1}{2b} e^{-\frac{|x|}{b}}.$$

Fig. 4 shows a plot of the Laplace distribution with $b = 0.045$:

Intuitively, the noise added to the answer should be sufficient to cover the maximum effect of a single data on the query outcome. Let F be this value. The Laplace mechanism is defined

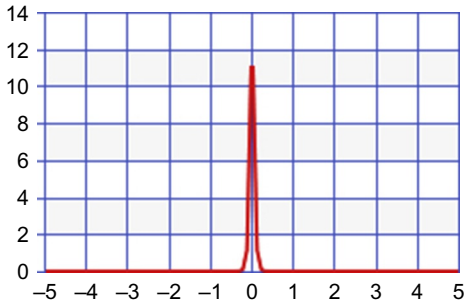

FIG. 4 Laplace distribution with $b = 0.045$.

as follows: if f is the actual query result, return $f + noise$, where *noise* is drawn from the Laplace distribution with $b = F/\epsilon$. This mechanism is ϵ-differentially private.

Example. Suppose the database contains the grade point average (GPA) of all students. Assume the goal is to release the average GPA of the students in the database. We further assume that there are 1000 students and that the maximum GPA is 4.5. One could easily see that the maximum effect F of one record on the final outcome is $4.5/1000 = 0.0045$. Assume we would like to guarantee 0.1-differential privacy. We add noise following Laplace distribution with $b = F/\epsilon = 0.0045/0.1 = 0.045$. The distribution of the noise is given in Fig. 4.

6 CONCLUSION

In this chapter, we presented various definitions in relation to user privacy protections. We also discussed the various mechanisms to support these definitions. For an in-depth treatment of the subject, readers are referred to the book by Dwork and Roth (2014).

References

Aggarwal, G., Panigrahy, R., Feder, T., Thomas, D., Kenthapadi, K., Khuller, S., Zhu, A., 2010. Achieving anonymity via clustering. ACM Trans. Algor. 6 (3), 1–19.

Cao, J., Karras, P., 2012. Publishing microdata with a robust privacy guarantee. PVLDB 5 (11), 1388–1399.

Chaum, D., 1985. Security without identification: transaction systems to make big brother obsolete. Commun. ACM 28 (10), 1030–1044.

Chen, R., Mohammed, N., Fung, B.C.M., Desai, B.C., Xiong, L., 2011. Publishing set-valued data via differential privacy. PVLDB 4 (11), 1087–1098.

Clifton, C., Tassa, T., 2013. On syntactic anonymity and differential privacy. Trans. Data Privacy 6 (2), 161–183.

Doka, K., Xue, M., Tsoumakos, D., Karras, P., 2015. k-anonymization by freeform generalization. In: Bao, F., Miller, S., Zhou, J., Ahn, G.J. (Eds.), Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS'15), April 14–17, 2015, Singapore. ACM, pp. 519–530.

Dwork, C., 2006. Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (Eds.), Proceedings of 33rd International Colloquium, Automata, Languages and Programming (ICALP 2006), July 10–14, 2006, Venice, Italy. Part II, Lecture Notes in Computer Science, 4052. Springer, pp. 1–12.

Dwork, C., Roth, A., 2014. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. 9 (3-4), 211–407.

El Emam, K., Dankar, F.K., Issa, R., Jonker, E., Amyot, D., Cogo, E., Corriveau, J.P., Walker, M., Chowdhury, S., Vaillancourt, R., Roffey, T., Bottomley, J., 2009. Research paper: a globally optimal k-anonymity method for the de-identification of health data. JAMIA 16 (5), 670–682.

Ghinita, G., Karras, P., Kalnis, P., Mamoulis, N., 2007. Fast data anonymization with low information loss. In: Koch, C. (Ed.), Proceedings of the 33rd International Conference on Very Large Data Bases, 23-27 September 2007, University of Vienna, Austria. ACM, pp. 758–769.

Iwuchukwu, T., Naughton, J., 2007. K-anonymization as spatial indexing: toward scalable and incremental anonymization. In: Koch, C. (Ed.), Proceedings of the 33rd International Conference on Very Large Data Bases, September 23–27, 2007, University of Vienna, Austria. ACM, pp. 746–757.

LeFevre, K., DeWitt, D.J., Ramakrishnan, R., 2005. Incognito: efficient full-domain k-anonymity. In: Özcan, F. (Ed.), Proceedings of the ACM SIGMOD International Conference on Management of Data, June 14–16, 2005, Baltimore, MD. ACM, pp. 49–60.

LeFevre, K., DeWitt, D.J., Ramakrishnan, R., 2006. Mondrian multidimensional k-anonymity. In: Liu, L., Reuter, A., Whang, K.Y., Zhang, J. (Eds.), Proceedings of the 22nd International Conference on Data Engineering (ICDE 2006), April 3–8, 2006, Atlanta, GA. IEEE Computer Society, p. 25.

LeFevre, K., DeWitt, D., Ramakrishnan, R., 2008. Workload-aware anonymization techniques for large-scale datasets. ACM Trans. Database Syst. 33 (3), 1–47.

Li, N., Li, T., Venkatasubramanian, S., 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In: Chirkova, R., Dogac, A., Özsu, M.T., Sellis, T. (Eds.), Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007), April 15–20, 2007, The Marmara Hotel, Istanbul, Turkey. IEEE Computer Society, p. 106–115.

Li, N., Qardaji, W.H., Su, D., 2011. Provably private data anonymization: or, k-anonymity meets differential privacy. CoRR. Abs/1101.2604.

Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M., 2007. L-diversity: privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data 1 (1) Article 3.

Narayanan, A., Shmatikov, V., 2008. Robust de-anonymization of large sparse datasets. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy (S&P 2008), May 18–21, 2008, Oakland, CA. IEEE Computer Society, pp. 111–125.

Sweeney, L., 2002. k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzz. Knowl. Based Syst. 10 (5), 557–570.

Wang, R., Xue, M., Liu, K., Qian, H., 2015. Data-driven privacy analytics: a wechat case study in address-based social networks. In: Xu, K., Zhu, H. (Eds.), Proceedings of 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015), August 10–12, 2015, Qufu, China. Lecture Notes in Computer Science, 9204. Springer, pp. 561–570.

Wong, W.K., Mamoulis, N., Cheung, D.W.L., 2010. Non-homogeneous generalization in privacy preserving data publishing. In: Elmagarmid, A., Agrawal, D. (Eds.), Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2010), June 6–10, Indianapolis, IN. ACM, pp. 747–758.

Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Ada Wai-Chee, F., 2006. Utility-based anonymization using local recoding. In: Eliassi-Rad, T., Ungar, L., Craven, M., Gunopulos, D. (Eds.), Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 20–23, 2006, Philadelphia, PA. ACM, pp. 785–790.

Xue, M., Karras, P., Chedy Raïssi, J.V., Tan, K., 2012. Anonymizing set-valued data by nonreciprocal recoding. In: Yang, Q., Agarwal, D., Pei, J. (Eds.), The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12), August 12–16, 2012, Beijing, China. ACM, pp. 1050–1058.

ABOUT THE AUTHORS

Xingye Lu received a bachelor's degree from the School of Computer Science and Engineering, Southeast University, China, in 2014. Currently she is a PhD student at the Department of Computing at Hong Kong Polytechnic University.

Man Ho Au received his undergraduate and graduate degrees from the Department of Information Engineering, Chinese University of Hong Kong, in 2003 and 2005, respectively,

and a PhD from the University of Wollongong, Australia, in 2009. Currently, he is an assistant professor in the Department of Computing at Hong Kong Polytechnic University. Dr. Au's research interests include public key cryptography, information security, accountable anonymity, and cloud security. He has published over 90 papers in these areas. He has served as a program committee member for over 30 international conferences. He is an associate editor of the *Journal of Information Security and Applications*, Elsevier.