Issues of privacy policy conflict in mobile social network

Pingshui Wang¹ and Qinjuan Ma²

Abstract
Privacy protection in mobile social networks is a hot issue in current research. Various privacy protection policies have been proposed. However, the conflict of privacy protection policies inevitably occurs. In this article, aiming at the personalized privacy protection model proposed in our published paper, we analyzed the possible conflict between privacy policies and comprehensively considered the policy conflict brought by the resource-level relationship; meanwhile, we proposed a scheme of consistency verification for privacy policy to improve the previous personalized privacy protection model. We also verified the practical effects of the improved model by experiments on synthetic data sets.

Keywords
Mobile social network, privacy policy, conflict detection, consistency verification

Date received: 6 July 2019; accepted: 19 February 2020
Handling Editor: Cong Wang

Introduction
With the wide application of mobile devices and Web 2.0 techniques, mobile social networks (MSNs) have experienced exponential growth in recent years. They provide users with a platform for communication, sharing information, and making friends. With the popularity and development of social networks, social networking sites store a large number of users’ personal data, which brings much convenience to data analysis. At the same time, it also causes great threat and challenge to individuals’ privacy, because MSN data may contain personal private information. Protecting the privacy of users against unwanted disclosure in such circumstance poses challenging problems. Issues on privacy disclosure are the greatest threat to the personal information security in the era of big data.¹⁻⁹

In recent years, the issues on privacy protection in MSN are deeply researched, and many effective privacy-preserving technologies have been developed. The existing researches on MSN privacy protection mainly concentrate on privacy-preserving data publishing, data mining, and access control,¹⁰⁻²³ in which anonymization is the main privacy-preserving technology for social network data release, so that the data released can meet the need of data analysis while user privacy is not compromised; and social network access control techniques mainly focus on designing social network access control model to solve the problem of social network data access authorization.¹⁸,²⁴⁻³⁵

However, the conflict of privacy protection policies of access control model inevitably occurs. In this article, we summarized the main access control models in MSN, analyzed their contribution, and point out their disadvantages. Especially, in view of the privacy protection model supporting personalized privacy preferences, we put forward in our published paper,²¹ which

¹School of Management Science and Engineering, Anhui University of Finance & Economics, Bengbu, China
²School of Business Administration, Anhui University of Finance & Economics, Bengbu, China

Corresponding author:
Pingshui Wang, School of Management Science and Engineering, Anhui University of Finance & Economics, Bengbu 233041, China.
Email: pshwang@163.com
can meet the user’s personalized privacy policy needs to a certain extent. However, due to the overlapping or hierarchical relationship among rules’ subject attributes, resource attributes, and action attributes, there may be logical inconsistencies in the formulation of privacy policies, for example, both positive authorization and negative authorization may exist on the same subject and object in different strategies, which will result in the privacy policy conflict. Therefore, according to whether the cause of policy conflict is related to the specific data, we defined the relevant privacy rules and analyzed the possible conflict (such as logical conflict and instance conflict) between privacy policies and comprehensively considered the strategy conflict brought by the resource-level relationship, which could improve the privacy protection of MSN users; meanwhile, we verified the feasibility of the improved model by experiments on synthetic data sets.

The related work

Access control in MSN is one of the most common manners of users’ privacy protection. The main access control model is described in the following.

Role-based access control model18 implements access control according to a pre-set role and the corresponding access privilege; however, the method mainly aims at the determined user community and cannot solve the problem of access authorization to unknown users and dynamic resources.

Attribute-based access control model can provide a better solution to the above problem.28,30 It realizes the dynamic access control in open environment using a set of attribute authorization rules based on the subject attribute, object attribute, and environment attribute constraints, but the model is only applied to the situation that the owner and manager of resource are integrated in the social network, which access control policy is developed by the manager of resource, so it is not suitable for the condition that the owner and manager of resource are separated, and it cannot satisfy the requirement of social network users’ personalized privacy preferences.

Rule-based access control model29 defines the relationship between the visitor and owner of resource, the maximum topological distance and minimum confidence, and other restrictions by rules so that the automatic and flexible access control is achieved based on rules reasoning. However, due to the large number of rules, it is prone to result in conflicted policy and cannot guarantee the consistency authorization and effective implementation of policies.

Authorization rules–based access control model31 adds the concepts of user attributes and permissions allocation rules based on rule-based access control model. It achieves the dynamic role permission assignment, but the model does not meet the demand of user-defined privacy policies. Wang et al. propose a practical privacy policy defined through authorization model supporting personalized privacy preferences.

Privacy policy conflict analysis

Due to the overlap or hierarchical relationship among the subject attributes, resource attributes, and action attributes of rules, there may be logic inconsistencies in the formulation of privacy policies. For example, in different policies, there are both positive authorization and negative authorization for the same subject and object, which results in privacy policy conflicts. According to whether the cause of policy conflict is related to the specific data, it can be divided into two aspects: logical conflict and instance conflict.

Definition of relevant rules

Definition 1 (visitor role permissions rules, VR-Rule). assign_role (u, role) = \( Q_1 \land \cdots \land Q_m \), \( s_1 \in \Theta s_2 \land \cdots \land \Theta s_n \), in which \( u \in \text{UserS} \), role \( \in \text{RoleS} \). \( \Theta \) may be logical operators such as “and” (\( \land \)) or “or” (\( \lor \)), \( s_i \) \( \in \text{SAttrC} \), \( i = 1, \ldots, n \). \( Q_j \) is an instance variable, \( j = 1, \ldots, m \). \( Q_j \in \{ \exists, \forall \} \); \( \exists \) is existential quantifier and \( \forall \) universal quantifier, that is the visitor “u” gets the role “role” while satisfying all subject attribute constraints.

Example 1. assign_role (x, ‘college classmate’) = \( \forall x \ Larger(x, \text{age}, ’18’ \}) \} \) is (x, class, ‘12computer-1’ \} \} is (x, graduate, ‘AUFE’) means that peoples satisfy the rule of over 18 years old and come from ‘AUFE’ and ‘12computer-1’ are “college classmate.”

Definition 2 (role permission assignment rules, RP-Rule). P_assign[D_assign] (role, r, a) = \( Q_1 \land \cdots \land Q_m \), \( r \in \text{RoleS} \), \( a \in \text{ActS} \). \( \Theta \) may be logical operators such as “and” (\( \land \)) or “or” (\( \lor \)), \( r_i \in \{ \text{RtagC}, \text{EC} \} \), \( i = 1, \ldots, n \). \( x_j \) is an instance variable, \( j = 1, \ldots, m \). \( Q_j \in \{ \exists, \forall \} \); \( \exists \) is existential quantifier and \( \forall \) universal quantifier, P_assign positive and D_assign negative authorization, that is in an EC environment, meeting all constraints object permissions for the \( <r, a> \) is (is not) assigned to role “role.”

Examples 2. P_assign(‘college classmate’, y, ‘comment’) = \( \exists y \ Is(y, \text{type}, \text{photo}) \} \) is (y, tag, graduation), said the college classmatess can comment graduation photos.
Logical conflict

Logical conflict refers to the logical inconsistency in the process of policy definition, such as role contradiction delegation, which refers to the logical conflict that the same role is assigned both positive and negative authorization.

Another typical logical conflict is the privilege inheritance conflict, which is the contradiction between authorization and explicit authorization caused by role hierarchy. As shown in Figure 2, the circle represents the role, the square represents the privilege, \( +P \) and \( -P \) represent the positive and negative authorization to the same resource, respectively, the arrow represents the role hierarchy, and the solid line indicates an existing role—permission assignment relationship, the dashed line represents the newly added role—privilege assignment relationship. According to the inheritance relationship of permissions in the role hierarchy, when low-level roles are assigned positive authorization, high-level roles inherit positive authorization from low-level roles to high-level roles according to positive authorization. If negative authorization is added to high-level roles, it would conflict with the positive authorization of high-level roles and cause policy conflicts, as shown in Figure 1(b); when high-level roles include multiple low-level roles and there are mutually exclusive privileges between low-level roles, and if a new negative authorization is added to high-level roles, it would conflict with the negative authorization of low-level roles and cause policy conflicts, as shown in Figure 1(c).

Figure 1. Example of permission inheritance conflict: (a) policy conflict caused by negative authorization with single low-level role, (b) policy conflict caused by positive authorization, and (c) policy conflict caused by negative authorization with multiple low-level roles.

Instance conflict

Instance conflict means that there is no logical conflict on the policy definition itself, but there are policy conflicts caused by the instances in the database which trigger policy conflict conditions. In the authorization model that supports personalized privacy preference, users are authorized by \( VR-Rule \) and \( RP-Rule \). In the process of defining two kinds of rules, there may be a user instance that satisfies two kinds of role constraints simultaneously, which leads to the application of two opposite strategies at the same time and results in policy conflicts.

Privacy policy consistency verification

In order to effectively analyze the contradiction of privacy policy, the verification method of logical programming is adopted. The user-defined privacy policy is transformed into logical form. The conflict of privacy policy is automatically detected by rule reasoning. The specific process is shown in Figure 2, which can be divided into the following steps: (1) users define personalized privacy policy; (2) design access authorization reasoning rules and policy conflict rules according to privacy policy; (3) realize user queries on policy permission assignment and policy conflict; (4) according to conflict query request, call logical transformer to convert data and privacy policies stored in relational databases into facts; (5) reasoning engine completes automatic reasoning of user authorization and policy conflict based on existing facts and reasoning rules; and (6) present the result of policy conflict, and the conflict strategy is corrected by interacting with users.

Build fact base

Facts refer to the relationship between existing entities, which consist of predicate names and variables. The relational data stored in database and policy library are transformed into factual statements through logic transformation program, which is the basis of logical reasoning. The transformation process is shown as follows: first, according to the user’s query request and the content of different tables, the data are extracted
Design reasoning rules

Reasoning rules describe the dependencies between facts in the form of $h$: $b_1, b_2, ..., b_n$, where $h$ is the head of the rule, representing the conclusion of the rule, and $b_1, b_2, ..., b_n$ is the body of the rule, indicating the conditions under which the rule is established. According to the authorization model supporting personalized privacy preference, the formal reasoning rules are designed as follows.

**Definition 3 (visitor-role permission rule).** This rule indicates that the role $Role$ and the visitor $U$ named $UserName$ exist in the fact base, and that the visitor $U$ principal attribute satisfies all the constraints of the principal attribute of the role $Role$, then the visitor $U$ is authorized to the role $Role$.

assign_role(UserName,Role):- role(_Role),user(User,'name',UserName,_),role_on_attr(Role,U_attr,U_attr_range), user(User,U_attr,U_attr_value,U_attr_type),match(U_attr_type,U_attr_value,U_attr_range).

**Definition 4 (role-permission assignment rule).** Role permission includes not only direct permission assignment but also permission inheritance caused by role hierarchy. Therefore, this rule mainly includes three parts: (1) direct permission assignment, $permission_limit$ defines the object access privilege for a given label, $object_tag$ and $match$ query label matching object, $assign_PER$ and $permission$ grant eligible object access privilege to role $Role$; (2) low-level role $Child$ is assigned positive authorization, according to positive authorization forward propagation, and high-level role $Ancestor$ inherits low-level role positive authorization; (3) if high-level role $Ancestor$ is assigned positive authorization, according to negative authorization reverse propagation, low-level role $Child$ inherits high-level role negative authorization. The specific conversion rules are as follows.

assign_role(UserName,Role):- role(_Role),user(User,'name',UserName,_),role_on_attr(Role,U_attr,U_attr_range), user(User,U_attr,U_attr_value,U_attr_type),match(U_attr_type,U_attr_value,U_attr_range).

role_has_pe(PermissionLimit,Role):- permission_limit(PermissionLimit,O_tag_name,O_tag_range,Actions,Flag), object_tag(Object,O_tag_name,O_tag_type,O_tag_value),match('char',A_id,Actions), match(O_tag_type,O_tag_value,O_tag_range), permission(Object,A_id),assign_PER(Role,PermissionLimit).

role_has_pe(PermissionLimit,Role):- is_ancestor(Child,Ancestor), role_has_pe(PermissionLimit,Child),permission_limit(PermissionLimit,O_tag_name,O_tag_range,Actions,'grant').

role_has_pe(PermissionLimit,Role):- is_ancestor(Child,Ancestor), role_has_pe(PermissionLimit,Ancestor)
Definition 5 (user-permission authorization rule). It involves user-permission authorization through visitor-role authorization and role-permission assignment reasoning.

\[ \text{user\_has\_pe}(\text{User}, \text{Role}, \text{PermissionLimit}) \leftarrow \text{assign\_role}(\text{User}, \text{Role}), \text{role\_has\_pe}(\text{PermissionLimit}, \text{Role}). \]

Definition 6 (matching rule). The custom internal function match is used to determine whether a numerical Value is within a given range Range. It can be divided into the following three situations.

\[ \text{match}(\text{Type}, \text{Value}, \text{Range}) : \text{- constant}(\text{Range}), \text{constant}(\text{Value}), \text{be\_equal}(\text{Value}, \text{Range}). \]

\[ \text{be\_equal}(\text{Value}, \text{Range}) : \text{- Value = Range}. \]

Case 1 does not consider the data type, and the data Value is exactly same as the range Range, matching is valid:

\[ \text{match}(\text{Type}, \text{Value}, [\text{M}, \text{N}]) : \text{- Type = 'int', constant}(\text{M}), \text{constant}(\text{N}), \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{M} \leq \text{N}, \text{Value} \geq \text{M}, \text{Value} \leq \text{N}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{N} = \text{'null'}, \text{M} = \text{'null'}, \text{Value} \geq \text{M}, \text{Value} \leq \text{N}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \leq \text{M}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \geq \text{N}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \leq \text{N}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \geq \text{M}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \leq \text{N}). \]

\[ \text{contain\_num}([\text{P}, \text{Q}], [\text{M}, \text{N}]) : \text{- contain\_num}(\text{P}, [\text{M}, \text{N}]), \text{contain\_num}(\text{Q}, [\text{M}, \text{N}]). \]

Case 2 judges the matching relation of number type data:

\[ \text{match}(\text{Type}, \text{Value}, [\text{M}, \text{N}]) : \text{- Type = 'int', constant}(\text{M}), \text{constant}(\text{N}), \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{M} \leq \text{N}, \text{Value} \geq \text{M}, \text{Value} \leq \text{N}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{N} = \text{'null'}, \text{M} = \text{'null'}, \text{Value} \geq \text{M}, \text{Value} \leq \text{N}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \leq \text{M}). \]

\[ \text{contain\_num}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), (\text{M} = \text{'null'}, \text{N} = \text{'null'}, \text{Value} \geq \text{N}). \]

\[ \text{contain\_num}([\text{P}, \text{Q}], [\text{M}, \text{N}]) : \text{- contain\_num}(\text{P}, [\text{M}, \text{N}]), \text{contain\_num}(\text{Q}, [\text{M}, \text{N}]). \]

Case 3 judges the matching relation of character type data:

\[ \text{match}(\text{Type}, \text{Value}, [\text{M}, \text{N}]) : \text{- Type = 'char', constant}(\text{M}), \text{constant}(\text{N}), \text{contain\_str}(\text{Value}, [\text{M}, \text{N}]). \]

\[ \text{contain\_str}(\text{Value}, [\text{M}, \text{N}]) : \text{- constant}(\text{Value}), \text{member}(\text{Value}, [\text{M}, \text{N}]). \]

\[ \text{contain\_str}([\text{P}, \text{Q}], [\text{M}, \text{N}]) : \text{- contain\_str}(\text{P}, [\text{M}, \text{N}]), \text{contain\_str}(\text{Q}, [\text{M}, \text{N}]). \]

\[ \text{contain\_str}([\text{P}, \text{Q}], [\text{M}, \text{N}]) : \text{- contain\_str}(\text{P}, [\text{M}, \text{N}]), \text{contain\_str}(\text{Q}, [\text{M}, \text{N}]). \]

Policy conflict query

Through the analysis of query requests for policy conflict rules, the policy consistency verification is completed, including direct conflict query and personalized customized query. Direct conflict queries refer to queries that do not set the limits of queries and directly conduct policy conflict queries according to the inference rules of policy conflict. This query not only provides the results of policy conflict detection but does not list the reasons for policy conflict. In order to help users find the causes of policy conflict and achieve the revision of policy conflict, the query of the following two rules is provided:
1. Logical conflict path rule: give the complete authorization path of user-role authorization conflict so that users can get the authorization path of logical conflict.

2. Instance conflict path rule: give the complete authorization path of user-permission authorization conflict so that users can get the authorization path of instance conflict.

3. User authorization path rule: giving the complete path of visitor authorization, users can restrict certain variables according to their own needs, implement personalized customized queries, and find possible policy conflicts in the whole authorization.

Because direct conflict query uses enumeration method, it is inefficient to execute when there are a large number of rules. Therefore, by adding personalized customized query, users can customize the scope of query restriction, and achieve rapid verification and accurate location of the cause of policy conflict.

Experimental result analysis

To integrate the authorization model supporting personalized privacy preference into the existing social network system, we designed a personalized privacy policy management system, which allows users to define personalized privacy policies and implement access control based on privacy policy.

Experiments have been carried on synthetic data sets. The experimental results show that the proposed privacy protecting model with privacy policy consistency verification could effectively improve the security of the MSN while keeping high execution efficiency. The system experimental environment is described as follows. CPU: Intel® Core™ i7-6500U @2.50GHz, RAM: 8 GB, software environment: Windows 7, development language: Anaconda 3, Database System: SQL-Server 2012.

Aiming at different ways of conflict query, we first test the impact of the number of users on query performance. Suppose that the user information table has 10 attributes, according to each additional 10 users for a group of experiments, each group of queries carry on 50 tests, we calculate the average query time of 10 rounds. The experimental results are shown in Figure 3, where the direct conflict query refers to querying instance conflict rules directly without setting query restriction range, that is conflict (User, Object, Action), represented by dotted lines. Personalized query refers to restricting certain variables to query user authorization path rules, that is conflict (User, photo1, Action), restricting Object = photo1, which is expressed in a straight line. The experimental results show that with the increase in the number of users, the direct conflict query time increases linearly, because the direct conflict
query is detected by enumeration, the number of users increases, and the enumeration query number increases correspondingly, which leads to the rapid growth of the query time; compared with the direct conflict query execution, the personalized query is more efficient than direct conflict query, because personalized queries limit some variables, which can quickly locate the causes of policy conflicts, and is less affected by the number of users.

Second, we test the impact of resource quantity on query performance. The number of selected users is 100. The experimental results are shown in Figure 4, where the direct conflict query is conflict (User, Object, Action), represented by dotted lines, and the personalized query named as query_per (User, Role, PermissionLimit, Object, Action) is restricted to User = “Alice,” represented by a straight line. The experimental results show that with the increase in the number of resources, the direct conflict query time increases linearly first and then gradually stabilizes, because the access authorization of the model is aimed at satisfying all the resources of the object label, not the authorization of a resource. Although the number of resources increases, the policy conflict query time is relatively stable when the resources satisfying the object label constraint are determined; personalized query is more efficient than direct conflict query and less affected by resource quantity.

Finally, we test the effect of the number of users on the performance of personalized queries under different conditions. The experimental results are shown in Figure 5. Queries with three variables, query_per (‘Alice’, Role, PermissionLimit, ‘photo1’,read), are restricted by three query conditions: User = “Alice,” Object = “photo1,” Action = “read,” whose query time is represented by a solid line. Queries restrict one query condition, that is query_per (User, Role,
the previous personalized privacy protection model. We also verified the practical effects of the improved model by experiments on synthetic data sets. In the next step, we would verify the feasibility of the model by experiments on real data sets and try to embed our model in real MSNs, for example, we can embed our systems in MSNs in the future.

Acknowledgements

The authors the anonymous reviewers and editors for their very constructive comments.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Social Science Foundation Project of China under grant 16BTQ085.

ORCID iD

Pingshui Wang https://orcid.org/0000-0002-8702-0332

References

1. Garcia D, Goel M, Agrawal AK, et al. Collective aspects of privacy in the Twitter social network. EPJ Data Sci 2018; 7(1): 3.
2. Heravi S and Raymond C. Information privacy in online social networks: uses and gratification perspective. Comput Human Behav 2018; 84: 441–459.
3. Liu X, Wang B and Yang X. Survey on privacy preserving techniques for publishing social network data. J Software 2014; 25(3): 576–590.
4. Rathore S, Sharma P and Loia V. Social network security: issues, challenges, threats, and solutions. Inform Sci 2017; 421: 43–69.
5. Wang Y, Sun Y M, et al. Specification and enforcement of personalized privacy policy for social network. J Commun 2012; 33(Suppl. 1): 239–249.
6. Yang Z, Huang YF, Li X, et al. Efficient secure data provenance scheme in multimedia outsourcing and sharing. Comput Mater Cont 2018; 56(1): 1–17.
7. Zeng D, Dai Y, Wang J, et al. Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. J Intell Fuzzy Syst 2019; 36(5): 3971–3980.
8. Gu K, Yang L and Yin B. Location data record privacy protection based on differential privacy mechanism. Inform Technol Contr 2018; 47(4): 639–654.
9. Gu K, Jia W, Wang G, et al. Efficient and secure attribute-based signature for monotone predicates. Acta Inform 2017; 54(5): 521–541.
10. Cheng Y, Park J and Shu R. An access control model for mobile social networks using user-to-user relationships. IEEE Trans Depend Secure Comput 2016; 13(4): 424–436.
11. Kokicivan N and Yolum P. PriGuard: a semantic approach to detect privacy violations in online social networks. IEEE Trans Knowl Data Eng 2016; 28(10): 2724–2737.
12. Kumar S and Kumar P. Upper approximation based privacy preserving in online social networks. Exp Syst Appl 2017; 88: 276–289.
13. Schlegel R, Chow C, Huang Q, et al. Privacy-preserving location sharing services for social networks. IEEE Trans Serv Comput 2017; 10(5): 811–825.
14. Soliman L and Girdzijauskas S. CADIVA: cooperative and adaptive decentralized identity validation model for social networks. Soc Netw Anal Min 2016; 6(1): 36.
15. Sun C, Yu P, Kong X, et al. Privacy preserving social network publication against mutual friend attacks. Trans Data Priv 2014; 7(2): 71–97.
16. Tai C, Yu P, Yang D, et al. Privacy-preserving social network publication against friendship attacks. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, CA, 21–24 August 2011, pp.1262–1270. New York: ACM.
17. Thapa W, Liao M and Li P. SPA: a secure and private auction framework for decentralized online social networks. IEEE Trans Para Distrib Syst 2016; 27(8): 2394–2407.
18. Li J, Tang Y and Mao C. Role based access control for social network sites. In: Proceedings of joint conferences on pervasive computing, Taipei, Taiwan, 3–5 December 2009, pp.389–394. New York: IEEE.
19. Wang T, Srivatsa M and Liu L. Fine-grained access control of personal data. In: ACM symposium on access control models and technologies, Newark, NJ, 20–22 June 2012, pp.145–156. New York: ACM.
20. Zou L, Chen L and Ozu M. K-automorphism: a general framework for privacy preserving network publication. In: Proceedings of the 35th international conference on very large databases, Lyon, 24–28 August 2009, pp.946–957. New York: ACM.
21. Wang PS, Wang ZC, Chen T, et al. Personalized privacy protecting model in mobile social network. Comput Mater Cont 2019; 59(2): 533–546.
22. Ren YJ, Leng Y and Cheng YP. Secure data storage based on blockchain and coding in edge computing. Math Biosci Eng 2019; 16(4): 1874–1892.
23. Liu YP, Ren YJ, Ge CP, et al. A CCA-secure multi-conditional proxy broadcast re-encryption scheme for cloud storage system. J Inform Secur Appl 2019; 47: 125–131.
24. Wang J, Gao Y, Yin X, et al. An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks. Wireless Commun Mobile Comput 2018; 2018: 9472075.
25. Gu K, Wang Y and Wen S. Traceable threshold proxy signature. J Inform Sci Eng 2017; 33(1): 63–79.
26. He S, Zeng W, Xie K, et al. PPNC: privacy preserving scheme for random linear network coding in smart grid. KSII Trans Int Inform Syst 2017; 11(3): 1510–1533.
27. Adam N, Atluri V, Bertino E, et al. A content-based authorization model for digital libraries. *IEEE Trans Knowl Data Eng* 2002; 14(2): 296–315.
28. Yuan E and Tong J. Attributed Based Access Control (ABAC) for web services. In: *Proceedings of the IEEE international conference on web services*, Orlando, FL, 11–15 July 2005, pp.561–569. New York: IEEE.
29. Carminati B, Ferrari E and Perego A. Rule-based access control for social networks. In: Meersman R, Tari Z and Herrero P (eds) *Lecture notes in computer science, on the move to meaningful internet systems: OTM’06 workshops*, vol. 4278. Berlin: Springer, 2006, pp.1734–1744.
30. Cirio L, Cruz I and Tamassia R. A role and attribute based access control system using semantic web technologies. In: *Proceedings of international federation for information processing workshop on semantic web and web semantics*, 2007, pp.1256–1266, https://www.cs.uic.edu/~ifc/webpapers/swws07-camera.pdf
31. Jayaraman K, Rinard M and Tripunitara M. Automatic error finding in access-control policies. In: *Proceedings of 18th ACM conference on computer and communications security*, Chicago, IL, 17–21 October 2011, pp.17–21. New York: IEEE.
32. Ma L, Tao L, Zhong Y, et al. RuleSN: research and application of social network access control model. In: *Proceedings of the IEEE international conference on intelligent data and security*, New York, 9–10 April 2016, pp.418–423. New York: IEEE.
33. Such JM and Criado N. Resolving multi-party privacy conflicts in social media. *IEEE Trans Knowl Data Eng* 2016; 28(7): 1851–1863.
34. Ren YJ, Leng Y, Zhu FJ, et al. Data storage mechanism based on blockchain with privacy protection in wireless body area network. *Sensors* 2019; 19(10): 2395.
35. Gu K, Jia W and Zhang J. Identity-based multi-proxy signature scheme in the standard model. *Fund Inform* 2017; 150(2): 179–210.