Composite Service-Oriented Minimum Privacy Disclosure Method

Changbo Ke¹, Zhiqiu Huang² and Linyun Liu³

¹School of Computer Science & Technology and School of Software, Nanjing University of Posts and Telecommunications, 210066 Nanjing, China
²College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 210066 Nanjing, China
³Department of E-Commerce, Nanjing Audit University, 210066 Nanjing, China

ABSTRACT

Service composition has become a main application to satisfy user functional requirements. To assure user privacy information not being disclosed, using the user minimum privacy data set when providing application, is a research focus of privacy enhancement technology in the process of service composition. In this paper, first, we describe and discrete the continuous privacy data. Second, through privacy sensitive analysis and service availability analysis, we obtain the minimum privacy disclosure data set to protect user privacy information. In the end, we prove the feasibility and correctness of our method with case study.

INTRODUCTION

Privacy as a non-functional property, is proposed as a human right in the beginning[1][2]. With the fast development of information technology, the extension of privacy is also enlarged. Privacy enhancement technology is more important in the field of software and internet. With the wide spread of social internet, the enlarged software scale, complexity and the wide application of mobile intelligent terminal, user requires higher software capability and reliability. Besides functional property, non-functional property, such as privacy security, has play a major role in the assessment of software quality[3]. Therefore, how to assure the consistency between user privacy requirement and software model, the correctness of model evolvement and the privacy strategy satisfiability in the runtime of software, has become a research difficulty in the composite service[4][5].

In the domain of information systems and software engineering, privacy protection means the capability to prevent an individual’s information from being collected, disclosed and stored by others [6][7]. In cloud computing, privacy protection is defined as the capability of cloud service user to control personal sensitive information without being collected, disclosed and stored by cloud service provider[8][9]. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [10][11]. Such characters enhance the service quality and reduce the wastage of computing resources. In the mean time, cloud computing is a multilateral collaborative interaction computing paradigm, and user privacy data must be transparent in the lifecycle of collaborative interaction among SaaS service[12][13]. Therefore, user may lose control of their privacy data, and
user privacy data may easily be disclosed by the composite service participants if the composite service evolves. For example, Google was sued by users in America because of its new unified privacy policy implemented from Mar. 1st, 2012. In Europe the implementation of this new privacy policy had been investigated by European Union and postponed. In 2016, at Times Warner, an American cable company, approximately 320,000 customers’ private data were hacked. Yahoo declared that at least 500 million user accounts were hacked including user name, Email address, telephone number, birth day and code.

For instance, user A wants to buy an article of clothes on Amazon. First, A must tell Amazon his name, phone number, address, and other private information including credit card number if A chooses to pay for it online. In addition, Amazon must transmit this information to a shipper to deliver the clothes. In such a process, the private data of user A can be encrypted in the process of storage and transmission. However, it must be decrypted when it is transmitted to the shipper. Under this condition, to solve issues such as “Minimum Privacy Disclosure” becomes very important.

Based on the above instance analysis, in this paper, we propose a method for Minimum Privacy Disclosure. The main innovations are as follows.

(1) We propose a method to describe and discrete the continuous privacy data.
(2) We analyze privacy sensitivity and service availability.
(3) We obtain the minimum privacy disclosure data set to protect user privacy information.

On service computing, research on privacy enhancement technology is currently at the stage of concept modeling. For instance, Anupam Datta, a Carnegie Mellon University professor, proposed a logic reasoning method to protect privacy data, considering the life cycle of software development[14]. Orienting service composite process evolution, Qun Ni in Purdue University proposed a role-based privacy aware model[15].

The rest of our paper is organized as follows: Section 2 presents the privacy disclosure analysis and checking method, including the method of privacy data discretization analysis and description, privacy sensitivity degree and service availability metric. Section 3 presents a case study. Section 4 introduces related works. Section 5 concludes our work and describes future work.

MINIMUM PRIVACY DISCLOSURE ANALYSIS AND DETECTION

By mapping using knowledge domain ontology, we can learn relationships among privacy attributes, and then describe them with an ontology tree. According to the user requirements and the hierarchy structure, we classify the privacy data into two types: discrete privacy data and continuous privacy data.

Minimum Privacy Disclosure Analysis

Definition 1 Privacy data Disclosure Chain. According to sensitive degree, user private data is represented as a partial order set \( \langle PD, \preceq \rangle \), \( DC \subseteq PD \), in which PD is privacy data set, namely, \( PD = \{pd_1, pd_2,......pd_i,......pd_n\} \). Suppose
∀pd_i, pd_j ∈ DC , and pd_i < pd_j , then DC is disclosure chain. The number of elements in partial order set |dc| is the length of disclosure chain.

**Definition 2 Discrete Privacy Data/Continuity Privacy Data.** Discrete privacy data means that, among the privacy data set, the combination of any privacy data will not lead to user privacy information disclosure. It can be represented as \(DP = \prec \exists pd(pd_i), pd_j \succ dc\), where \(i, j \geq 1\) \(dc \in DC\). Continuous privacy data means that, among the privacy data set, a combination of two privacy data items can cause the exposure of user privacy information. It can be represented as \(CP = \prec \exists pd(pd_i), pd_j \succ dc\), where \(i, j, r \geq 1\).

**Definition 3 KP (Key Privacy Data).** We define the privacy items that can assure that user identity is key privacy data. In a privacy data set, there are certain privacy data whose combination with any other privacy data is continuous; we call this type of privacy data KP. These data can be presented as \(KP = \prec \exists pd(pd_i), pd_k \succ dc\), where \(i \geq 1, n \geq k \geq 1\) and \(i \neq k\). The privacy items that cannot assure the user’s identity is defined as Non-Key Privacy Data, which is the opposite of Key Privacy Data. It can be presented as \(nkp = \neg kp\), where \(nkp \in NKP \wedge kp \in KP\).

**Definition 4 DPC (Composition Discrete Privacy Data Chain).** The chain that consists of the discrete privacy data set is called the composition discrete privacy data chain. It must satisfy either condition below.

(1) In the discrete privacy data set, any combination between discrete privacy data and continuous privacy data (or discrete privacy data) cannot lead to continuous privacy data. In other words, \([\prec \forall dp(dp_i), cp \succ cp] \lor [\prec \forall dp(dp_i), dpj \succ cp]\), where \(cp \notin kp\). Usually, we call such a discrete data chain an external-combination discrete privacy data chain E-DPC.

(2) Suppose that the chain length is \(a\) and a combination of the set that consists of the privacy data chain and key privacy data can elicit continuous privacy data, while random elements in the set together with the key privacy data cannot elicit continuous privacy data. This arrangement can be presented as \([\prec dp(dp_i), kp \succ cp] \wedge [\prec \forall dp(dp_i), kp \succ cp]\), where \(k = a\). Such a discrete data chain is called an internal-combination discrete privacy data chain I-DPC.

**Axiom 1** Disclosure chain DC must contain one or more key privacy data KP, namely, \(kp \cap dc \subseteq kp\).

**Proof:** According to the definition of key privacy data KP, we know that \(kp = \prec \exists pd(pd_i), pd_k \succ dc\). Then \(\exists pd(pd_i) \subseteq dc\), namely \(kp \subseteq dc\). Therefore, \(kp \cap dc \subseteq kp\).

According to definition 2, continuous privacy data must include a disclosure chain, or contain one or more key privacy data.

Continuous privacy data can be protected by decomposing them into discrete privacy data. Because CP must include a disclosure chain, we can decompose the parent data with I-DPC in the disclosure chain into E-DPC.

First, matching the existing disclosure chains in continuous privacy data, we set the element in the disclosure chain to be the root node and search the privacy ontology tree with a breadth-first search to find the existing child node set. Second, forming the child node set into the internal discrete data chain I-DPC. Third, using the internal discrete data chain I-DPC as a substitute for privacy data \(pd_i\). Fourth, deleting the data outside
the chain in I-DPC, and substituting the former disclosure chain to a non-disclosure chain.

For CP contain one or more key privacy data, we first traverse the privacy ontology tree when we set the key privacy data to be the root node. If there is no child node for the key privacy data, we check the privacy data set, search for the disclosure chain and decompose it.

If there is a child node for the key privacy data, we decompose the existing disclosure chains according to the following steps.

First, searching the privacy ontology tree when setting the key privacy data KP as the root node to determine the child node set. Second, forming the child node set in the internal discrete data chain I-DPC. Third, using the internal discrete data chain I-DPC as a substitute for the privacy data KP. Fourth, deleting the data outside the chain in the internal discrete data chain and substituting the former disclosure chain with a non-disclosure chain.

**Minimum Privacy Disclosure detection**

**Definition 5 Privacy Sensitivity Degree.** Privacy sensitivity degree is user’s sensitive to personal privacy data. Suppose \( p_s = \{p_{s0}, p_{s1}, \ldots, p_{sn-1}\} \) is user’s privacy data chain. Privacy sensitivity degree can be represented as \( sv = \{sv_0, sv_1, \ldots, sv_{n-1}\} \), where \( sv_i \) represents the sensitivity degree of \( p_i, 1 \leq i \leq n \). There are two conditions for user’s privacy sensitivity degree:

1. If user has clarified privacy requirement, the privacy sensitivity degree is defined as a random real number among interval \([0,1]\), where 0 means the weakest sensitivity degree, and 1 means the strongest sensitivity degree.

2. If user does not have clarified privacy requirement, the privacy sensitivity degree can obtained through recursion formula, according to the probability of using privacy data. The recursion formula is as following,

\[
f(n) = \begin{cases} \infty & (n = 0) \\ \omega & (n = 1) \\ f(n-1) \times [1 - m \alpha + k] & (n \geq 2) \end{cases}
\]  

(1)

In which \( m \) is the probability of using privacy data, \( \alpha \) is coefficient and \( k \) is constant.

**Axiom 1** Suppose the user privacy data is assorted in an ascending order according to the privacy sensitivity degree, namely \( \frac{1}{sv_0} \geq \frac{1}{sv_1} \geq \ldots \geq \frac{1}{sv_{n-1}} \), and privacy data is decomposable. When select those privacy data which its sensitivity degree is no more than user required, then the privacy data set, which is sequentially selected, must be the optimum data set that satisfying user privacy requirement.

Proof: Suppose \( X = (x_0, x_1, \ldots, x_{n-1}) \), \( 0 \leq x_i \leq 1 \), \( 0 \leq i < n \), is the optimum privacy data set that satisfying user privacy requirement. If all privacy data could not be disclosed when user obtained the required service, namely, \( x_i = 0 \), then this is must be the optimum solution. Or else, suppose \( r \) is the minimum subscript that make \( x_r \neq 1 \).

From the sequentially selection way, we know that the solution of privacy data set is as following.
\[ X = (1, \ldots, 1, x_r, 0, \ldots, 0), \quad 0 \leq x_r \leq 1 \]

If \( X \) is not the optimum solution that satisfying user privacy requirement, and \( Y = (y_0, y_1, \ldots, y_k, y_{k+1}, \ldots, y_{n-1}) \) is the optimum solution, which make \( \sum_{i=0}^{n-1} \frac{1}{s v_i} y_i > \sum_{i=0}^{n-1} \frac{1}{s v_i} x_i \), suppose \( k \) is the minimum subscript that makes \( y_k \neq x_k \), then there must be \( y_k < x_k \).

We explain from the following three aspects, if \( k < r \), since \( x_k = 1 \), \( y_k \neq x_k \), therefore, \( y_k < x_k \).

(2) If \( k = r \), then \( x_r \) is the maximum subset that the decomposed privacy data which sequentially numbered as \( k \) can be joined in the privacy data set satisfying user requirement. Therefore, it is impossible that \( y_k > x_k \). Since \( y_k \neq x_k \), therefore, \( y_k < x_k \).

(3) If \( k > r \), since \( x_l = 0 \), \( r < i < n \), if \( y_k \neq 0 \), then the privacy data set must do not satisfy user privacy requirement.

From the above three aspects, we can get that \( y_k < x_k \).

Suppose \( y_k \) is substituted in the \( Y = (y_0, y_1, \ldots, y_k, y_{k+1}, \ldots, y_{n-1}) \) by \( x_k \), then new privacy data set \( Z = \{z_1, z_2, \ldots, z_k, z_{k+1}, \ldots, z_{n-1}\} \) are obtained. Noted that before substitution, \( z_i = y_i, \quad 0 \leq i < k - 1 \), while after substitution, \( z_k = x_k \). To assure \( Z \) satisfying user privacy requirement, the equation \( \sum_{i=0}^{n-1} (y_i - z_i) = z_k - y_k \) makes sense.

Since \( \sum_{i=0}^{n-1} \frac{1}{s v_i} z_i = \sum_{i=0}^{n-1} \frac{1}{s v_i} y_i - \sum_{i=0}^{n-1} \frac{1}{s v_i} (y_i - z_i) = \sum_{i=0}^{n-1} \frac{1}{s v_i} y_i - (y_k - z_k) \frac{1}{s v_k} - \sum_{i=k+1}^{n-1} \frac{1}{s v_i} (y_i - z_i) = \sum_{i=0}^{n-1} \frac{1}{s v_i} y_i \)

Therefore, \( X = (x_0, x_1, \ldots, x_{n-1}) \), \( 0 \leq x_i \leq 1 \), \( 0 \leq i < n \), is the optimum privacy data set that satisfying user privacy requirement.

**Definition 6 Service Availability Privacy Data Set.** Service Availability Privacy Data Set is defined as a minimum privacy data set that allowing service to provide user application. The privacy data set can be constructed with a state space tree. Suppose \( P_A = \{P_{A1}, P_{A2}, \cdots, P_{An}\} \) is the user privacy data set, \( W = \{w_1, w_2, w_3, \ldots, w_n\} \) is the corresponding availability metric value of each user privacy information, in which the value is set by service provider, and \( W \) is an ascending sequence, namely, \( w_i \leq w_{i+1} \), and \( X = \{x_1, x_2, x_3, \ldots, x_n\} \) is the state of service availability set, in which \( x_i \in \{1, 0\} \). The constraint conditions of state space tree are defined as \( \sum_{i=0}^{k-1} w_i x_i + \sum_{i=k}^{n-1} w_i \geq m \), and \( \sum_{i=0}^{k-1} w_i x_i + w_k \leq m \), where \( M \) is the sum of the corresponding availability metric value, for each user private information in the minimum privacy data set that allowing service to provide user application. The node of state tree can be expressed as a triple form, namely, in which \( r = \sum_{i=0}^{k-1} w_i x_i + w_k \), \( s = \sum_{i=0}^{k-1} w_i x_i + \sum_{i=k}^{n-1} w_i \). When \( \sum_{i=0}^{k-1} w_i x_i + w_k \geq m \), the feasible solution for service availability privacy data set can be obtained. Only
when \( \sum_{i=0}^{k-1} w_i x_i + w_k = m \), the optimum solution for service availability privacy data set can be obtained.

The algorithm to solve the service availability privacy data set is as following:

```c
void Service-Availability (float s, int k, float r, int*x, float m, float* w)
{
    x[k] = 1;
    if (s+w[k] = m){
        for (int j=0; j<=k; j++)cout<<x[j]<<"\n";
        cout<<\n
    }
    else if (s+w[k]+w[k+1]<=m)
    SubOfPrivacy(s+w[k], k+1, r-w[k], x, m, w);
    if ((s+r-w[k]>=m)&&(s+w[k+1]<=m))
    x[k]=0;
    SubOfPrivacy(s, k+1, r-w[k], x, m, w);
}

void SubOfPrivacy(int*x, int n, float m, float* w)
{

    float r = 0;
    for(int i=0; i<n; i++) r=r+w[i];
    if(r>=m && w[0]<=m)
    { SubOfPrivacy(0, 0, r, x, m, w); }
}
```

The initial condition of Algorithm SubOfPrivacy() is \( s = 0, r = \sum_{i=0}^{n-1} w_i x_i \geq m \), and \( w_o \leq m \). Although \( k > n - 1 \) is not tested, we can know that, according to the precondition of recursive algorithm, when at the level of \( k = n - 1 \), there is \( s + w_{n-1} \leq m \), and \( s + r \geq m, r = w_{n-1} \), therefore \( s + w_{n-1} = m \). Or else, if one condition is false, then will not enter to the level of \( k = n - 1 \). Therefore the function will terminate as long as assuring the initial condition of the Algorithm SubOfPrivacy() is true.

**Definition 7 Privacy data- Service satisfiability** \( P_{S,A} \) (\( P_{\text{Sensitivity- Availability}} \)). The satisfiability between the privacy data and service can be measured, by computing the intersection between the privacy data set that satisfying user privacy requirement and the privacy data set that allowing service to provide user application, namely, \( P_{S,A} = P_s \cap P_a = \{p_{a1}, p_{a2}, \ldots, p_{an}\} \cap \{p_{A1}, p_{A2}, \ldots, p_{A\text{\#}}\} \). There are four intersection results.

(1) The intersection is an availability set, namely, \( P_{S,A} = P_s \cap P_a = \{p_{a1}, p_{a2}, \ldots, p_{an}\} \). It represents that \( P_s \supseteq P_a \), and the privacy data set satisfying the input and precondition of the service, meaning the service is available.

(2) The intersection is a sensitivity privacy data set, namely, \( P_{S,A} = P_s \cap P_a = \{p_{a1}, p_{a2}, \ldots, p_{an}\} \). It represents that \( P_s \subset P_a \), and the privacy data set does not satisfy the input and precondition of the service, meaning the service is not available. Under this circumstance, the elements can be decomposed in the sensitivity privacy data set, and started intersection again.

(3) The intersection is the subset of sensitivity privacy data set or service availability set, namely, \( P_{S,A} = P_s \cap P_a = \{p_{a1}, p_{a2}, \ldots, p_{an}\} \) or \( P_{S,A} = P_s \cap P_a = \{p_{a1}, p_{a2}, \ldots, p_{an}\} \), and
This privacy data set does not satisfy the input and precondition of the service, meaning the service is not available. Under this circumstance, the elements can be decomposed in the complementary set of intersection, and started intersection again.

(4) The intersection is a null set, namely, \( P_{S \setminus A} = P_S \cap P_A = \emptyset \). This privacy data set does not satisfy the input and precondition of the service, meaning the service is not available.

CASE STUDY

Suppose the input and precondition of all service participants that collected by service composer is user’s Name, Address, Phone Number, Postcode, Age and so on. User Hao Wang set personal privacy data Name, Address and Phone Number as disclosure chain, and Name as key privacy data. The privacy data set composed of input and pre-conditions requested by the service provider is continuous privacy data. We can protect user privacy data through the following steps:

**Step 1.** Construct privacy ontology tree according to the relationship among privacy data, as shown in lower left corner of Figure 1;

**Step 2.** Through matching obtain the corresponding privacy data Name, Address and Phone Number of the disclosure chain in continuous privacy data. Traversing the privacy ontology tree, we can find that Name has the child node of FirstName, SecondName and LastName, and Address has the child node of Country, Province, City, Street and Community. The Detailed work can be referred to our previous research paper[3], which is not illustrated in Figure 1;

**Step 3.** Decompose the privacy data, namely, using child node for substitution;

Figure 1. Privacy Data Discomposing Process.
Step 4: Discretize the newly formed privacy data chain. Namely, delete the rear privacy data in the internal discrete data chain and turn the continuous data chain into a discrete data chain;

Step 5: Assign a value to the privacy data in the discrete data chain to get an instance of the discrete privacy data chain. Such a process can also be deemed as the User behavior for sending a privacy data set to the service provider, namely:

Name (HAO WANG); Street (MOFAN); City (NANJING); Province (JIANGSU); Country (CHINA); PhoneNumber (+86-123456789); Postcode (210033); Age (30);

According to the defined user disclosure chain and key privacy data, the final obtained privacy data chain satisfy user privacy requirement.

Suppose that we obtain the corresponding sensitivity degree value of discretized privacy data through formula 1 in definition 5 as follows, (LN, Postcode, PhoneNumber, Street, City, Province, Country, Age) = (0.3, 0.2, 0.8, 0.22, 0.18, 0.1, 0.03, 0.6). We assort the privacy data in an ascending order according to sensitivity degree, therefore the assortment is (Country, Province, City, Postcode, Street, LN, Age, PhoneNumber).

The corresponding service availability measured value is (LN, Postcode, PhoneNumber, Street, City, Province, Country, Age) = (0.4, 0.1, 0.9, 0.6, 0.5, 0.2, 0.15, 0.13), M=2.4. By constructing the state tree according to the constraint condition in definition 6, we obtain the minimum service availability privacy data set (PhoneNumber, LN, Street, City).

Client each time select a privacy data with minimum sensitivity degree, to compute the intersection with the minimum service availability privacy data set, namely,

\{\text{Country}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \emptyset

\{\text{Province}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \emptyset

\{\text{City}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \{\text{City}\}

\{\text{Postcode}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \emptyset

\{\text{Street}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \{\text{Street}\}

\{\text{LN}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \{\text{LN}\}

\{\text{Age}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \emptyset

\{\text{PhoneNumber}\} \cap \{\text{PhoneNumber}, \text{LN}, \text{Street}, \text{City}\} = \{\text{PhoneNumber}\}

Then client compute a union set of the results of intersection. Therefore obtain the minimum privacy data set that satisfying user requirement \{\text{City}, \text{Street}, \text{LN}, \text{PhoneNumber}\}.

\emptyset \cup \emptyset \cup \{\text{City}\} \cup \emptyset \cup \{\text{Street}\} \cup \{\text{LN}\} \cup \emptyset \cup \{\text{PhoneNumber}\}

= \{\text{City}, \text{Street}, \text{LN}, \text{PhoneNumber}\}.

RELATED WORKS

Description and Modeling of Privacy Requirement

Bo Tang et al. [16] proposed extensions for finer-grained cross-tenant trust and developed a prototype system to demonstrate the utility and practical feasibility. Omoronyia I [17] draws connections between the imprecision in user privacy preferences, and reasoning about the satisfaction of privacy requirements to protect user privacy. Smullen D et al. [18] use eddy to analysts and express requirements over data
practices, detect conflicting requirements and to trace data flows within and across specifications. Bahri L et al. [19] explored the available literature on web services privacy during transactions, identified 20 works that address privacy related problems in web services consumption, and proposed evaluation framework.

**Analysis and Verification of Privacy Requirement**

Oltramari A et al. [20] introduce PrivOnto, a semantic framework to represent annotated privacy policies, represent issues identified as critical to users and/or legal experts. Wei She et al. [21] focused on access control validation at composition time, which may be used to control the service providers accessing the user's private data. Tout H, Mourad A et al. [22] took advantage of both the Unified Modeling Language (UML) and the Aspect-Oriented Paradigm (AOP) to enforce Business Process Execution Language (BPEL) security policies. Changbo Ke et al. [23] propose a privacy data decomposition and discretization method for SaaS services. Changbo Ke et al. [24] propose a private data disclosure checking method that can be applied to the collaboration interaction process.

**Matching and Negotiation of Privacy Policy**

Changbo Ke et al. [25] defined user and service provider privacy policies based on analysis of current privacy rules and proposed a privacy policy automatic matching method. Wei Zhiqiang et al. [26] researched private data protection policies in the application of pervasive computing, and checked the inconsistency among policies. Changbo Ke et al. [27] proposed a privacy negotiation method between the user and the service provider in cloud computing to achieve the privacy requirement. Tbahriti S E et al. [28] presented a negotiation protocol pointing to the inconsistency of user and service provider privacy policies and also put forward a privacy system in which a privacy policy based on P3P can be defined.

**CONCLUSIONS**

We describe the privacy data and discrete the continuous privacy data. Through privacy sensitive analysis and service availability analysis, we obtain the minimum privacy disclosure data set. So as to prevent user privacy disclosure. Future work will focus on privacy extensions in service level agreements in order to support the monitoring of service composition.

**ACKNOWLEDGEMENT**

This work was supported by National Natural Science Foundation of China (Grant 61602262), Jiangsu Natural Science Foundation of China (Grant BK20150865, BK20130735), Jiangsu University Natural Science Foundation (Grant 15KJD520001, 13KJB520011), Sponsored by NUPTSF (Grant No. NY214164, NY215120), China Postdoctoral Science Foundation Funded Project (Project No. 2016M591842) and the
REFERENCES