Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4785142, 11 pages
https://doi.org/10.1155/2017/4785142

*Research Article*

# Privacy Data Decomposition and Discretization Method for SaaS Services

**Changbo Ke,[1,2,3] Zhiqiu Huang,[2] Fu Xiao,[1,3] and Linyuan Liu[4]**

[1]*School of Computer Science & Technology and School of Software, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China*
[3]*Jiangsu High Tech. Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*
[4]*Department of E-Commerce, Nanjing Audit University, Nanjing, Jiangsu 211815, China*

Correspondence should be addressed to Changbo Ke; brobo.ke@njupt.edu.cn

Received 17 April 2017; Accepted 28 May 2017; Published 9 July 2017

Academic Editor: Emilio Insfran



In cloud computing, user functional requirements are satisfied through service composition. However, due to the process of interaction and sharing among SaaS services, user privacy data tends to be illegally disclosed to the service participants. In this paper, we propose a privacy data decomposition and discretization method for SaaS services. First, according to logic between the data, we classify the privacy data into discrete privacy data and continuous privacy data. Next, in order to protect the user privacy information, continuous data chains are decomposed into discrete data chain, and discrete data chains are prevented from being synthesized into continuous data chains. Finally, we propose a protection framework for privacy data and demonstrate its correctness and feasibility with experiments.

## 1. Introduction

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Some characteristics (such as service outsourcing, virtualization, distribution, and multitenancy) of cloud computing enhance the service quality and save computing resources. For example, service outsourcing enhances service capability and specialization through service composition [2]. However, cloud computing is a collaborative computing and transparent interaction system. User privacy data are transparent during collaborative interaction, and then they are stored or used by service participants. Therefore, the user will lose control of personal data, which may easily lead to a disclosure of private data. For instance, on June 2013, about six million Facebook users' personal email addresses and telephone numbers were disclosed. On August 2013 the same event happened to thirty-eight million Adobe users with the disclosure of data including user names, credit account information, credit card expiration dates, credit card passwords, and order information. In 2014 seven million Target users' archives were leaked, including bank account information, telephone numbers, and email addresses.

Privacy was regarded as a kind of human right in the beginning. 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 [3, 4]. In 2002, the World Wide Consortium (W3C) developed the Platform for Privacy Preferences (P3P), which provides a standard and machine-understandable privacy policy which is compared with the user's privacy preferences when the user visits a web site. Users can then select different services according to the results. The Extensible Access Control Markup Language (XACML) 2.0 extends the privacy policy profile. However, XACML, P3P policy, and web service composition design are developed by different designers. During the process of privacy policy development, policymakers do not know what privacy data and limits of authority are actually needed when

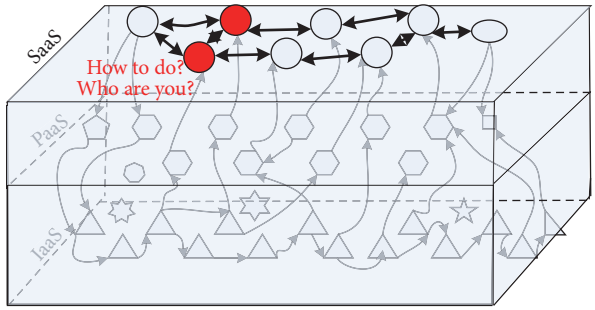



Figure 1: Cloud computing model.


Figure 2: Behaviors of interaction among SaaS services.

the service is running. Therefore, these techniques can hardly be applied to satisfy user privacy protection requirements in a collaborative and transparent interaction environment. Most scholars use traditional information security methods, for instance, encryption or anonymity, to perform research on privacy data protection in cloud computing [5–8].

These methods are effective for static data or bilateral interaction and transmission, but they cannot satisfy privacy protection requirements in a collaborative and transparent interaction environment. In papers [9, 10] privacy protection in cloud computing is defined as the capability of a user to control Personal Sensitive Information (PSI) so that it cannot be collected, used, disclosed, or stored by cloud service providers. These two papers provide certain theoretical guidance but do not put forward specific solutions for privacy protection in cloud computing.

Any application intended to meet users' functional requirements needs a combination of certain services including IaaS, PaaS, and SaaS. In this process, interactive private data must be in plain text among the SaaS services. Encryption can guarantee the security of private data in IaaS and PaaS, but in SaaS private data must be decrypted. Therefore, whether the service accepting plain text satisfies users requirements cannot be guaranteed by encryption or anonymity. Details are shown in Figure 1.

The gray box in Figure 1 represents that private data will be either encrypted or anonymous when it is transmitted through IaaS and PaaS, which is invisible to the service participants of IaaS and PaaS. At the top of gray box, interactive private data among SaaS services is in plain text, which is visible to the service participants.

For instance, user A obtains application services using cloud computing. On Monday, he enquired about the local weather forecast and purchased 3 barrels of environmentally friendly paints through SaaS services. On Tuesday, he purchased lamps and network equipment. On Wednesday, he purchased domestic appliances. He kept in contact with the service provider through Messenger and shared his location. To purchase goods, he provided his name, address, phone number, and Messenger number and made a payment through an online payment service.

If the privacy information is directly provided to the service providers, SaaS services can get the name, address, telephone number, Messenger number, and credit card number of user A. In addition, his or her work address, career, and other privacy information can also be learned by analyzing the purpose and behavior of user A, and the user's purchasing preferences can be obtained by analyzing information about the goods.

Over a period of time, the SaaS service learns more privacy information through mining of the behaviors of user A, analyzing the parameters of the goods, and obtaining sensitive privacy data by analyzing the semantic relationship between the personal preferences and the personal identity information of the user.

When the transaction is executed between user A and the SaaS service, we check and analyze the semantic relationship of the privacy data and decompose and discretize the data that may disclose the user's sensitive privacy information such that the user's sensitive privacy data are not obtained by the SaaS services.

Figure 2 addresses the behaviors of interaction among SaaS services. The colored hollow spheres represent the SaaS services, and the red and green dots represent the privacy information in the process of an interaction.

Based on the above instance analysis, in this paper, we propose a method for decomposition and discretization of a user's privacy information. The main innovations are as follows.

(1) We propose a classification method for user privacy data, which divides the privacy data into continuous and discrete data.

(2) We discuss the methods of decomposition and discretization for continuous privacy data.

(3) We address a method to prevent discrete privacy data from being synthesized into continuous privacy data.

The rest of this paper is structured as follows: Section 2 explains the classification of privacy data. Section 3 addresses protection methods for discrete and continuous privacy data. Section 4 illustrates a case study and a performance analysis. Section 5 reviews related work. Finally, Section 6 presents conclusions and future work.

## 2. Classification of Privacy Data

By mapping using knowledge domain ontology, we can learn relationships among privacy attributes and then describe them with an ontology tree.

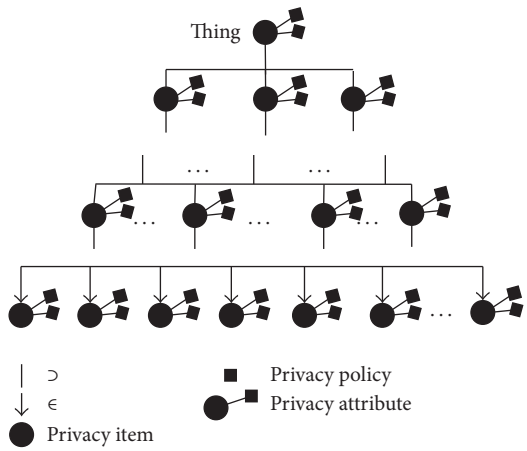



Figure 3: Privacy ontology tree.

The ontology tree of privacy attributes is shown in Figure 3. Thing is a root node, acting as the super class of privacy attributes. Except for the bottom layer, the rest of the layers' relationships are similar to that of a class and subclass, that is, inclusion relations. For example, the class name includes a real name and a nickname. The bottom layer is an instance of the upper layer. For example, country, province, and city belong to the class address. According to the user requirements and the hierarchy structure, we classify the privacy data into two types: discrete privacy data and continuous privacy data.

*Definition 1* (PD (privacy data)). Privacy data is the collection of privacy information which needs to be protected according to the user's privacy request. It can be presented as PD $= \{pd_1, pd_2, \ldots, pd_i, \ldots, pd_n\}$.

*Definition 2* (DC (disclosure chain)). Among the data that users want to protect, the partially ordered set $\langle \text{PD}, \prec \rangle$, DC $\subseteq$ PD, is defined according to the level of sensitivity of the users. If, $\forall pd_i$, $pd_j \in$ DC, $pd_i \prec pd_j$ can be satisfied, then we call DC the disclosure chain. The number of elements of the partially ordered set $|dc|$ is the length of DC.

*Definition 3* (DP (discrete privacy data)). Discrete privacy data mean that, among the privacy data set, the combination of any privacy data will not lead to user privacy information disclosure. It can be presented as DP $= \langle \neg\exists pd(pd_i), pd_j \rangle \rightarrow dc$, in which $i, j \geq 1$ $dc \in$ DC.

*Definition 4* (CP (continuous privacy data)). Continuous privacy data means that, among the privacy data set, the combination of two privacy data sets can cause the exposure of user privacy information. It can be presented as CP $= \langle \exists pd(pd_i), pd_j \rangle \rightarrow dc$, in which $i, j, r \geq 1$.

*Definition 5* (KP (key privacy data)). We define the privacy items that can be used to determine user identity as key privacy data. In the privacy data set, there is certain privacy data whose combination with any other privacy data is continuous; one calls this sort of privacy data KP. It can be presented as KP $= \langle \exists pd(pd_i), pd_k \rangle \rightarrow dc$, in which $i \geq 1, n \geq k \geq 1$, and $i \neq k$.

*Definition 6* (NKP (non-key privacy data)). We define privacy items that cannot confirm user identity as non-key privacy data, which is the negative of key privacy data. It can be presented as $nkp = \neg kp$, in which $nkp \in$ NKP $\wedge$ $kp \in$ KP.

*Definition 7* (DPC (composition discrete privacy data chain)). The chain that consists of a 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 of discrete privacy data and continuous privacy data (or discrete privacy data) cannot bring lead to continuous privacy data. In other words, $[\langle \forall dp(dp_i), cp_i \rangle \mapsto cp] \vee [\langle \forall dp(dp_i), dp_i \rangle \mapsto cp]$, where $cp_i \notin kp$. Usually, we call such a discrete privacy data chain an external-combination discrete privacy data chain E-DPC, or a farther discrete privacy data chain. For example, {name, address, phoneNumber, zipcode, age} is an E-DPC and the address is the farthest node of {country, province, city, district, street}.

(2) Suppose that the chain length is a combination of the set that consists of the privacy data chain and that the 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 data chain can be presented as $[\langle \{dp_1, dp_2, \ldots, dp_k\}, kp \rangle \rightarrow cp] \wedge [\langle \forall dp(dp_i), kp \rangle \mapsto cp]$, where $k = a$. Such a discrete data chain is called an internal-combination discrete privacy data chain I-DPC, or the son of a discrete data privacy data chain. For example, {country, province, city, district, street} is I-DPC, and they are the son nodes of the address in the privacy ontology tree.

**Theorem 8.** *Disclosure chain DC must include one or more key privacy data; that is,* $kp \cap dc \subseteq kp$.

*Proof.* According to the definition of key privacy data, if $kp = \langle \exists pd(pd_i), pd_k \rangle \rightarrow dc$, $\exists pd(pd_i) \subseteq dc(kp \subseteq dc)$. Therefore, $kp \cap dc \subseteq kp$. □

According to the above definition, we can acquire the semantic privacy data classification, which is shown in Box 1.

## 3. Method of Privacy Data Protection

### 3.1. Detection of Disclosure Chains

*Definition 9* (PIS (privacy item set)). The minimum privacy data set is exposed by users according to the service requirements; namely, PIS $= \{pd_1, pd_2, \ldots, pd_i, \ldots, pd_k\}$, in which $pd_i$ represents a privacy item and represents the subset of service input and precondition in the set; namely, $pd_i \subseteq (P_i, I_i)$, $0 \leq i \leq k$. PIS are service privacy item sets; $p$ and $I$ represent service preconditions and input, respectively.

*Definition 10* (MDCP (matching between disclosure chain and privacy items)). There are two conditions when matching the user disclosure chain and the service privacy item set.

Data Types: Description: Instances
PD: $\{pd_1, pd_2, \ldots, pd_i, \ldots, pd_n\}$: {name, IDcode, address, phoneNumber, ...}
DC: $dc \subseteq pd$: {name, address}, {name, phoneNumber}
DP: $\langle \neg\exists pd(pd_i), pd_j\rangle \rightarrow dc(i, j,\ r \geq 1)$: {name, zipcode, province, age, career}
CP: $\langle \exists pd(pd_i), pd_j\rangle \rightarrow dc$: {name, IDcard, address, phoneNumer, zipcode, age}
KP: $kp = \langle \exists pd(pd_i), pd_k\rangle \rightarrow dc\ (i \geq 1,\ n \geq k \geq 1,\ i \neq k)$: IDcard
NKP: $\neg kp$: {name, address, phoneNumber, zipcode, age}
E-DPC: $[\langle \forall dp(dp_i), dp_i\rangle \mapsto cp](cp_i \notin kp)$: {name, zipcode, province, age, career}
I-DPC: $[\langle \{dp_1, dp_2, \ldots, dp_k\}, kp\rangle \rightarrow cp] \wedge [\langle \forall dp(dp_i), kp\rangle \mapsto cp]$: {country, province, city, district, street}

Box 1: Semantic privacy data classification.

(1) All services in the service set satisfy the user requested disclosure chain.

The corresponding privacy item set PIS = $\{pd_1, pd_2, \ldots, pd_i, \ldots, pd_k\}$ of ready-to-be-combined services is such a set that satisfies the disclosure chain with respect to $S_i$. It satisfies the following formula: $\{\text{PIS} \wedge \text{service}(S_i) \wedge \langle \text{PIS}\rangle dc_i\} \wedge \Phi$, in which service($S_i$) means the service which participates in the service combination, $\langle \text{PIS}\rangle dc_i$ represents the matching relationship between the privacy item set and the disclosure chain, and $\Phi$ represents all services that participate in the service composition that satisfies the user disclosure chain. It can be shown as $\text{service}(S_1) \mapsto \langle pd_1\rangle dc_1 \wedge \cdots \wedge \text{service}(S_k) \mapsto \langle pd_k\rangle dc_k$.

(2) Certain services in the service set satisfy the user disclosure chain.

The corresponding privacy item set PIS = $\{pd_1, pd_2, \ldots, pd_i, \ldots, pd_k\}$ of ready-to-be-combined service is such a set which does not satisfy the disclosure chain as to certain services of $S_i$. It satisfies the following formula: $\{\text{PIS} \wedge \text{service}(S_i) \wedge \langle \text{PIS}\rangle dc_i\} \wedge \Gamma$, in which $\Gamma$ represents certain services that participate in the service composition and satisfy the user disclosure chain. $\Gamma$ can be presented as $\text{service}(S_1) \mapsto \langle pd_1\rangle dc_1 \vee \cdots \vee \text{service}(S_k) \mapsto \langle pd_k\rangle dc_k$.

To test whether disclosure chains defined by the user are included in the privacy data set requested by service provider, we put forward a specific implementation method (see Algorithm 1).

The first and second lines of Algorithm 1 are the input and output, respectively. From the third to the fifth lines, we initiate the queues of the privacy data set that are required by the composite services, DC, and the privacy data set in the DC. From the sixth to the eighth lines, we enter the privacy data set and DC into the queues. From the ninth to the twentieth lines, we check the consistency between the privacy data sets of the composite services and that of the DC. From the tenth to the fourteenth lines, we take advantage of the Tableau algorithm to match the privacy data sets of the composite services and that of the DC. From the fifteenth to nineteenth, we judge whether the service participants satisfy the DC. If true, we break the cycle; otherwise, we check the next DC.

*3.2. Protection of Discrete Data.* The method for the protection of discrete data aims to prevent the privacy data set which is requested by the service provider from forming a continuous privacy data chain. The discrete privacy data chain contains an external discrete data chain and an internal discrete data chain. We analyze the protection method as follows.

(1) Consider that the E-DPC is a discontinuous virtual chain; namely, certain data with logical relations are included in the data set but not included in the disclosure chain. However, when one or more sets of key privacy data are included in the discrete privacy data, the discontinuous virtual chains may be combined into a continuous disclosure chain. With the evolution of service, privacy data that the service requires will also evolve. With the joining of evolved privacy data, the discontinuous virtual chain may turn into a complete and continuous disclosure chain. Such an evolutionary process is shown in the upper part of Figure 4.

(2) Similarly, at the beginning of service composition, I-DPC is a continuous virtual chain. I-DPC has the character that only a whole chain of I-DPC may turn into a disclosure chain. With the joining of key privacy data, even if a continuous chain that contains the key privacy data is formed, it is not included in the disclosure chain and is not equivalent to the disclosure chain. However, with the evolution of service, a complete continuous disclosure chain may be formed. Such an evolutionary process is shown in the lower part of Figure 4.

The service composition process is also a process for combining user privacy data. From the above analysis, it can be seen that, just as with external discrete privacy data, preventing key privacy data from joining in combination can effectively protect user privacy data. For internal discrete privacy data, preventing key privacy data from combining with internal discrete privacy into a complete chain can avoid privacy data disclosure.

*3.3. Protection of Continuous Privacy Data.* According to Definition 4 of continuous privacy data, continuous privacy data includes such features as the following:

(1) Continuous privacy data definitely includes a disclosure chain.

(2) Continuous privacy data may include key privacy data.

Protection of continuous privacy data is managed by decomposing continuous privacy data into discrete privacy data. For feature (1), we can decompose parent data with I-DPC in the disclosure chain into E-DPC. Specifically, the following steps can be executed, as shown in Figure 5.

(1) *Input:* $(pd_i, dc_i)$
(2) *Output:* 0/1
(3) *Init Queue* $(ServiceReq_r\langle pd_i\rangle)$;
(4) *Init Queue* $(dc_k)$;
(5) *Init Queue* $(dc_k\langle pd_j\rangle)$;
(6) *EnQueue* $(Queue\ (ServiceReq_r\langle pd_i\rangle)\{pd_1, pd_2, pd_3, \ldots, pd_i, \ldots, pd_n\})$;
(7) *EnQueue* $(Queue\ (dc_k)\{dc_1, dc_2, dc_3, \ldots, dc_i, \ldots, dc_n\})$;
(8) *EnQueue* $(Queue\ (dc_k\langle pd_j\rangle)\{pd_1, pd_2, pd_3, \ldots, pd_i, \ldots, pd_n\})$;
(9) *for* $(i = 1; dc_i\ ! = \emptyset; i{+}{+})$
(10) ***while*** $(Queue\ (ServiceReq_r\langle pd_i\rangle)\ ! = \emptyset \wedge dc_k\langle pd_j\rangle ! = \emptyset)$ *do*
(11) *GetHead* $(Queue\ (ServiceReq_r\langle pd_i\rangle), pd_i)$;
(12) *GetHead* $(Queue\ (dc_k\langle pd_j\rangle), pd_j)$;
(13) *Tableau* $(pd_i, pd_j)$; // Check the privacy conflicts
(14) ***end while;***
(15) ***if*** $(MDCP\ \{PES \wedge service(S_i) \wedge \langle PES\rangle dc_i\} \wedge \Phi = false)$ *do*
// Judge whether the service participants satisfy the DC
(16) *EnQueue* $(Queue\ (dc_{i+1}\langle pd_j\rangle), dc_k)$;
(17) ***else;***
(18) ***break;***
(19) ***end if;***
(20) ***end for***

Algorithm 1: Match $((pd_i, dc_i), 0/1)$.

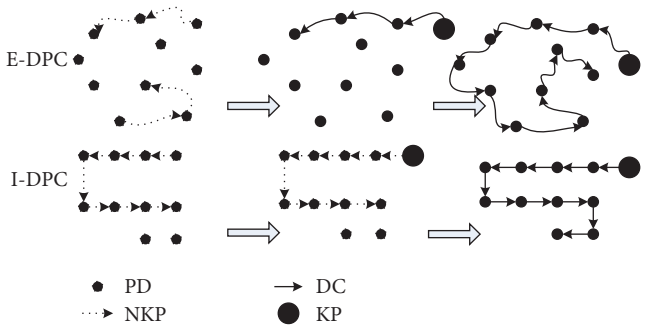



Figure 4: Evolution processes of E-DPC and I-DPC.

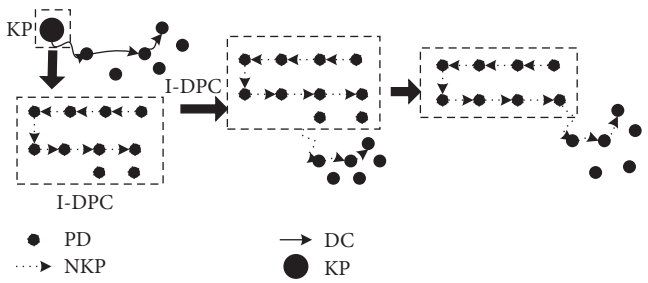



Figure 6: Decomposing KP into I-DPC.

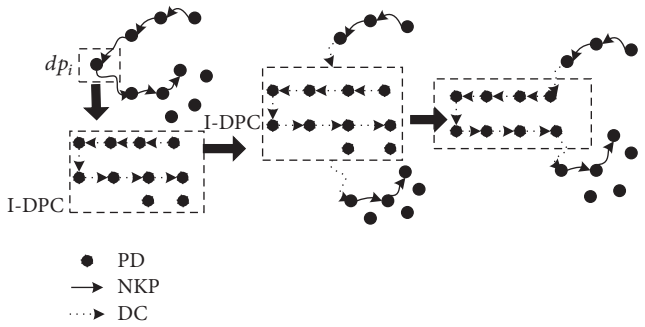



Figure 5: Decomposing privacy data in DC into I-DPC.

*Step 1.* Matching existing disclosure chains in the continuous privacy data, we set the elements in the disclosure chain as the root node, and search the privacy ontology tree with a breadth-first search, in order to find existing child node sets.

*Step 2.* Form the child node set into internal discrete data chain I-DPC.

*Step 3.* Use the internal discrete data chain I-DPC to substitute privacy data $pd_i$.

*Step 4.* Delete data outside the chain in I-DPC.

*Step 5.* Substitute the former disclosure chain into a nondisclosure chain.

Regarding feature (2), we first traverse the privacy ontology tree when we set the key privacy data as the root node. If there is no child node for the key privacy data, check the privacy data set and search for the disclosure chain; then decompose the disclosure chain as shown in Figure 5.

If there is child node for the key privacy data, decompose existing disclosure chains according to the following steps, as shown in Figure 6.

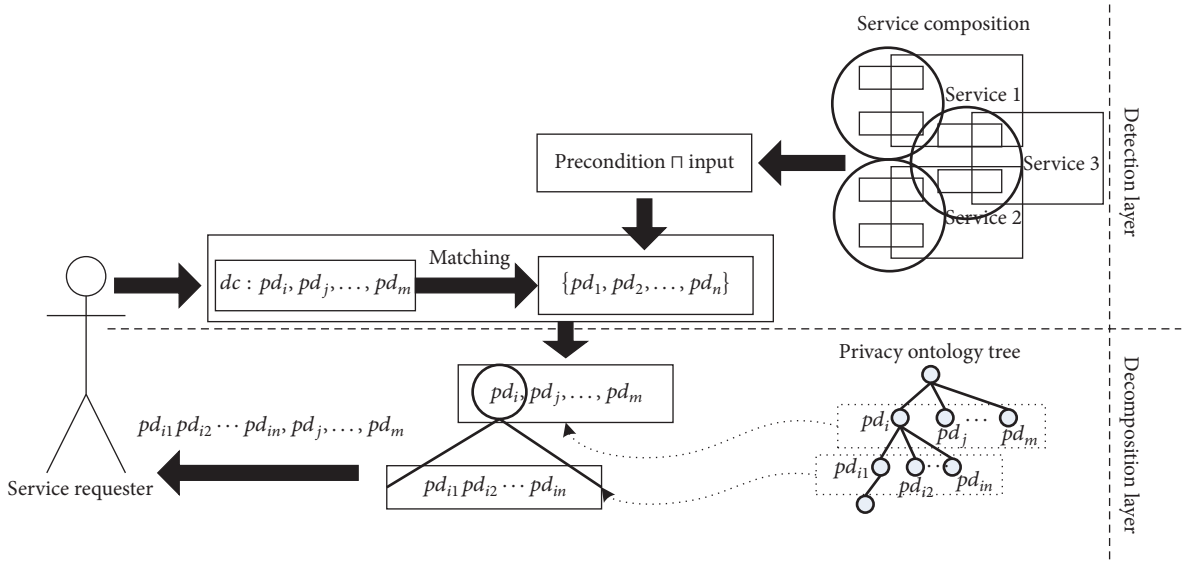



Figure 7: Privacy data decomposition and discretization framework.

*Step 1.* Search the privacy ontology tree when setting key privacy data KP as the root node, in order to find the child node set.

*Step 2.* Form the child node set into an internal discrete data chain I-DPC.

*Step 3.* Use the internal discrete data chain I-DPC to substitute privacy data KP.

*Step 4.* Delete data outside the chain in the internal discrete data chain.

*Step 5.* Substitute the former disclosure chain for the nondisclosure chain.

*3.4. Protection Framework of Privacy Data.* The privacy protection framework according to the data classification can be described as two layers, as shown in Figure 7.

*Detection Layer*. When user sends application requests to the cloud computing or composition service, the cloud computing or composition service will ask the user to provide corresponding privacy data as a precondition and as input. To protect their privacy, a user usually refuses to provide certain key privacy data or combinations of certain privacy data. Therefore, this layer mainly provides a kind of matching between the preconditions and input of the service provider and user privacy requests, namely, checking the disclosure chain and key privacy data in the privacy data set requested by the service provider.

*Decomposition Layer*. After getting key privacy data or the disclosure chain through matching in the detection layer, traverse the privacy ontology tree when setting key privacy data or certain privacy data in the disclosure chain as the root node. To find their corresponding child node, recompose them with the corresponding child node in order to satisfy the user privacy requirement. Lastly, send the privacy data set that is recomposed to the user for confirmation.

## 4. Case Study and Performance Analysis

*4.1. Case Study.* In this paper, we consider online shopping as an example to show the feasibility and validity of our methods, as shown in Figure 8. First, suppose that the composition satisfies user functional requirements, namely, that no evolution is involved when combining the service. Therefore, this example is for continuous privacy data. Completing this service involves an online cloud shopping platform Service Composer, a Customer (Tom), a Seller, a Shipper, and an E-Commerce Service, in which the Customer Name (Name), Address, Postal Code, Phone Number, and Age are the individual's privacy data. The complete trading process is as follows. When the Customer uses the E-Commerce Service to send goods requests to the Seller using the service of the Service Composer, according to the trading process, the E-Commerce Service, the Seller, and the Shipper ask the Customer for privacy data as input and as a precondition of service by the cloud Service Composer. After getting the data, the Seller sends the goods to the Customer according to the Customer Name, Address, and Phone number with the Shipper, and asks for payment.

Suppose that the Service Composer collects all of the service participant's input and preconditions such as their Name, Address, Phone Number, Postal Code, and Age. Imagine Customer Hao Wang sets his privacy data: Name, Address, and Phone Number as the disclosure chain, in the meantime setting Name as the key privacy data. Therefore, the privacy data set composed of input and preconditions requested by the service provider is continuous privacy data. We can adopt the methods below to protect user privacy data.

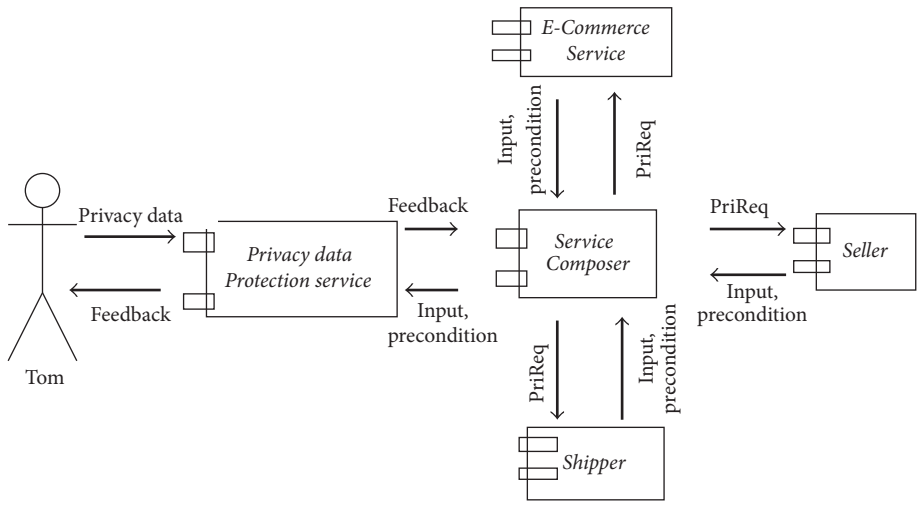



Figure 8: Online shopping case.

*Step 1.* Form a privacy ontology tree according to the relationships among the privacy data, as shown in the lower left part of Figure 9.

*Step 2.* Get the corresponding privacy data Name, Address, and Phone Number of the disclosure chain in the continuous privacy data after matching. Traverse the privacy ontology tree and find that there is a child node FirstName, SecondName, and LastName for Name, and a child node Country, Province, City, Street, and Community for Address. For a specific example of this step the reader can refer to our previous work in [3]. We do not include a corresponding expression in Figure 9.

*Step 3.* Decompose the privacy data Name and Address, using the child node for substitution.

*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).

*4.2. Performance Analysis.* The pseudo-distributed environment includes 1 IBM computer (Pentium D925, 4 G RAM). There are 10 computing nodes in the fully distributed environment, in which three computing nodes are virtual machines (512 M RAM) built on a server (8 G RAM) and 9 computing nodes are LENOVO computers (Pentium D925 CPU, 512 M RAM). The operating system is Ubuntu Linux 10.10. The experimental platform is built on Jdk 1.6.0_37, hadoop1.1.1, and juno eclipse. We mine private data from an artificial data set with COP [11], SOD [12], and LoOP [13] algorithms. For the implementation code for SOD, LoOP, and COP refer to http://elki.dbs.ifi.lmu.de/wiki#.

We designed eight artificial data sets with methods in reference [11–13], for a total of 3000 pieces of private data with different dimensions: 25, 50, 75, 100, 125, 150, 175, and 200. In each artificial data set, there are 102 pieces of private data and three clusters, each including 966 pieces of data. Ten, 15, and 20 attribute dimensions from the first, second, and third clusters are data with normal distributions. Data values for the rest of the attribute dimensions are evenly distributed.

We discretized the private data in the artificial set with our method and implemented the experiment with LoOP, COP, and SOD data mining algorithms, comparing against the artificial data set without discretization. The results are shown in Figures 10 and 11.

*Experiment 1.* Looking at the artificial set without discretization, the precision rate for private data mining changes with the different attribute dimensions (50, 100, 150, and 200) as shown in Figure 10(a). Figure 10(b) shows the results based on the discretized artificial set. Compared to the artificial set without discretization, the average precision rate decrease is approximately 12% for the discretized artificial set with LoOP, COP, and SOD algorithms.

*Experiment 2.* We take the retrieved data set from Experiment 1, as shown in Figures 10(a) and 10(b), as the experimental data. We use the SOD, COP, and LoOP algorithms to mine the KPD and PDC from the experimental data. By performing a comparison with the undiscretized data set, we find the precision of KPD and PDC in the discretized data set to have an average decrease of 69.17%.

The result means our method can reduce the precision rate for private data mining, because our method just reduces the semantic correlation between the private data, without affecting the effectiveness of data mining for the private data.

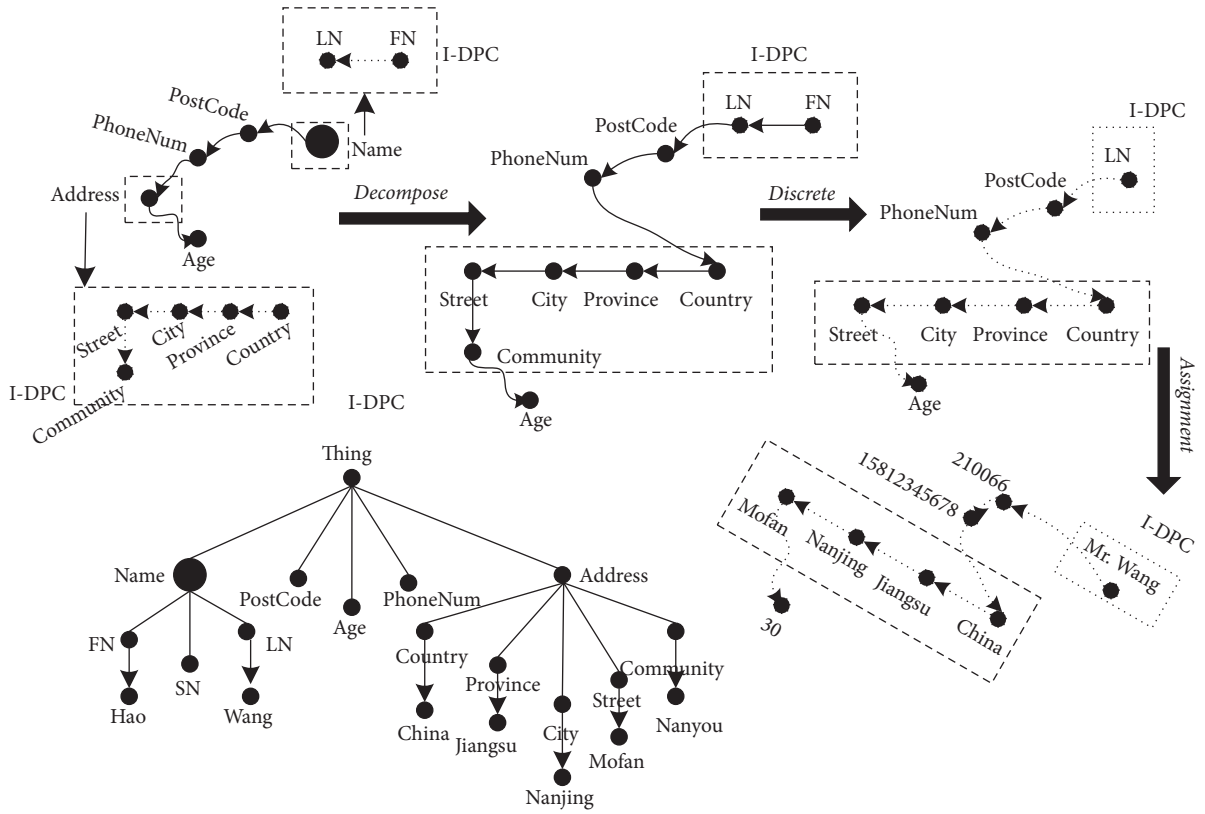



Figure 9: Analyzing process of decomposition and discretization for privacy data.

*Experiment 3.* Looking at the artificial set without discretization, the recall rate for private data mining is seen to change with the different attribute dimensions (50, 100, 150, and 200) in Figure 11(a).

Figure 11(b) shows the results based on the discretized artificial set. When compared to the artificial set without discretization, the average recall rate decreases approximately 43% for the discretized artificial set using the LoOP, COP, and SOD algorithms.

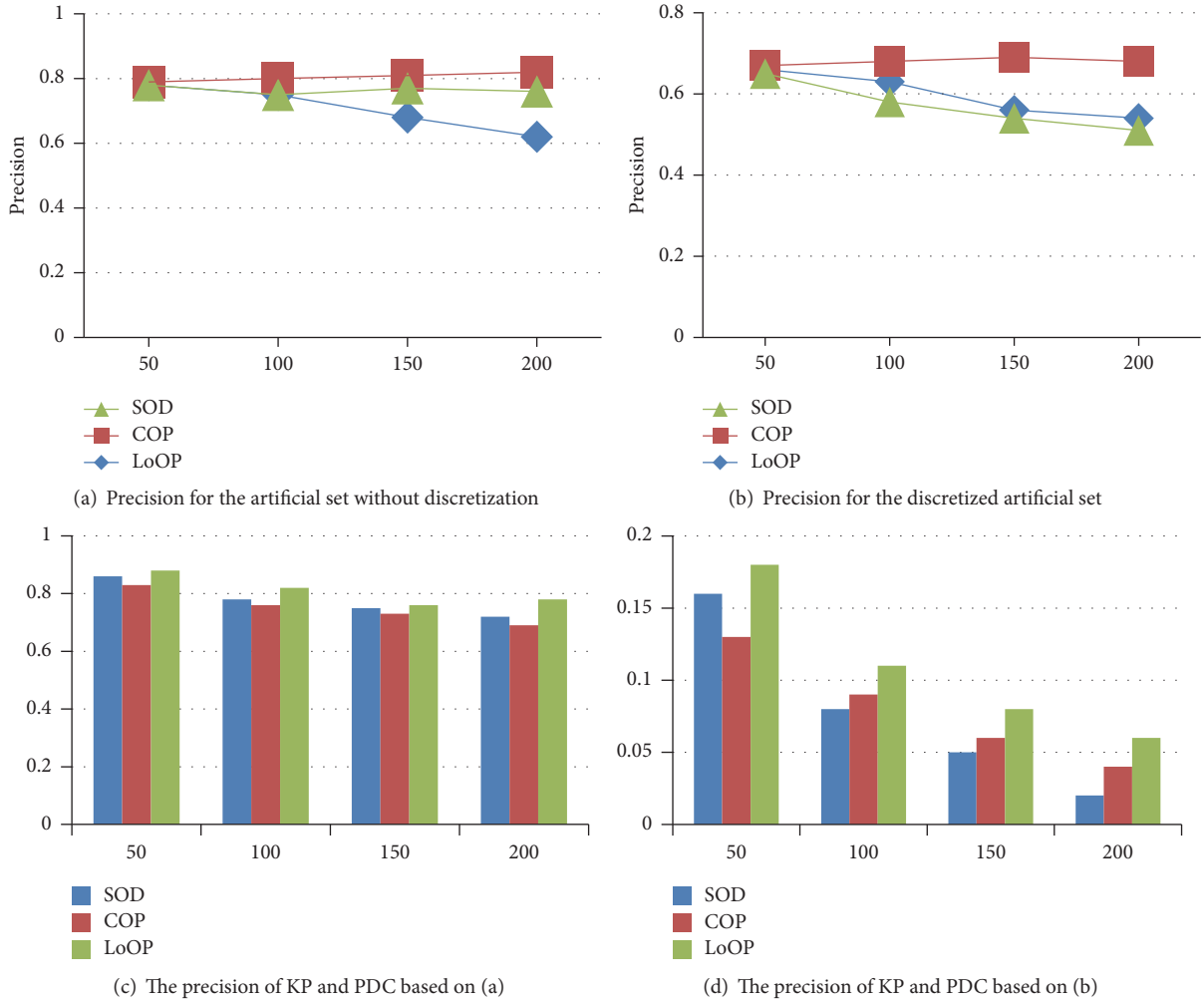
*Experiment 4.* We take the retrieved data set of Experiment 3, as shown in Figures 11(a) and 11(b), as the experimental data. We use the SOD, COP, and LoOP algorithms to mine the KPD and PDC from the experimental data. By performing a comparison with the undiscretized data set, we find the recall of KPD and PDC in the discretized data set to have an average decrease of 51.25%.

The result means our method can reduce the recall rate for private data mining. The range of decrease is wide, because our method reduces the semantic correlation between the private data, greatly affecting the effectiveness of data mining for private data chains.

## 5. Related Work

### 5.1. Computing Process Privacy

*(1) Modeling and Verification of Privacy Requirement.* Tang et al. [14] formalized the multitenancy authorization system (MTAS), proposed extensions for finer-grained cross-tenant trust, and developed a prototype system to demonstrate utility and practical feasibility. Hamadi et al. [15] proposed a conceptual model of a privacy-aware web service protocol and built a model of the service protocol integrating privacy requirements with an extended state machine, so that it became easier to develop and manage a privacy-aware web service protocol with a model-driven method. Guermouche et al. [16] presented a method for exchanging privacy web service protocols which extended the service protocol, built a privacy model based on rules, and analyzed the exchangeability of privacy service protocols using a finite-state machine. Mokhtari et al. [17] put forward a verification method of the privacy time attribute in a web service protocol and built a model of the private data maintenance period in a web service protocol using timed automata, to verify if the service protocol violated the maintenance period requirements of private data. Although this research modeled and verified the privacy property in a web service protocol, it mainly pointed to a single web service protocol without considering the privacy requirement in service composition. She et al. [18] focused on access control validation at composition time, which may be used to control service providers' access to a user's private data. Tout et al. [19] took advantage of both the Unified Modeling Language (UML) and the Aspect Oriented Paradigm (AOP) to enforce Business Process Execution Language (BPEL) security policies. Li et al. [20] proposed a graph-transformation based framework to check whether an




(a) Precision for the artificial set without discretization

(b) Precision for the discretized artificial set

(c) The precision of KP and PDC based on (a)

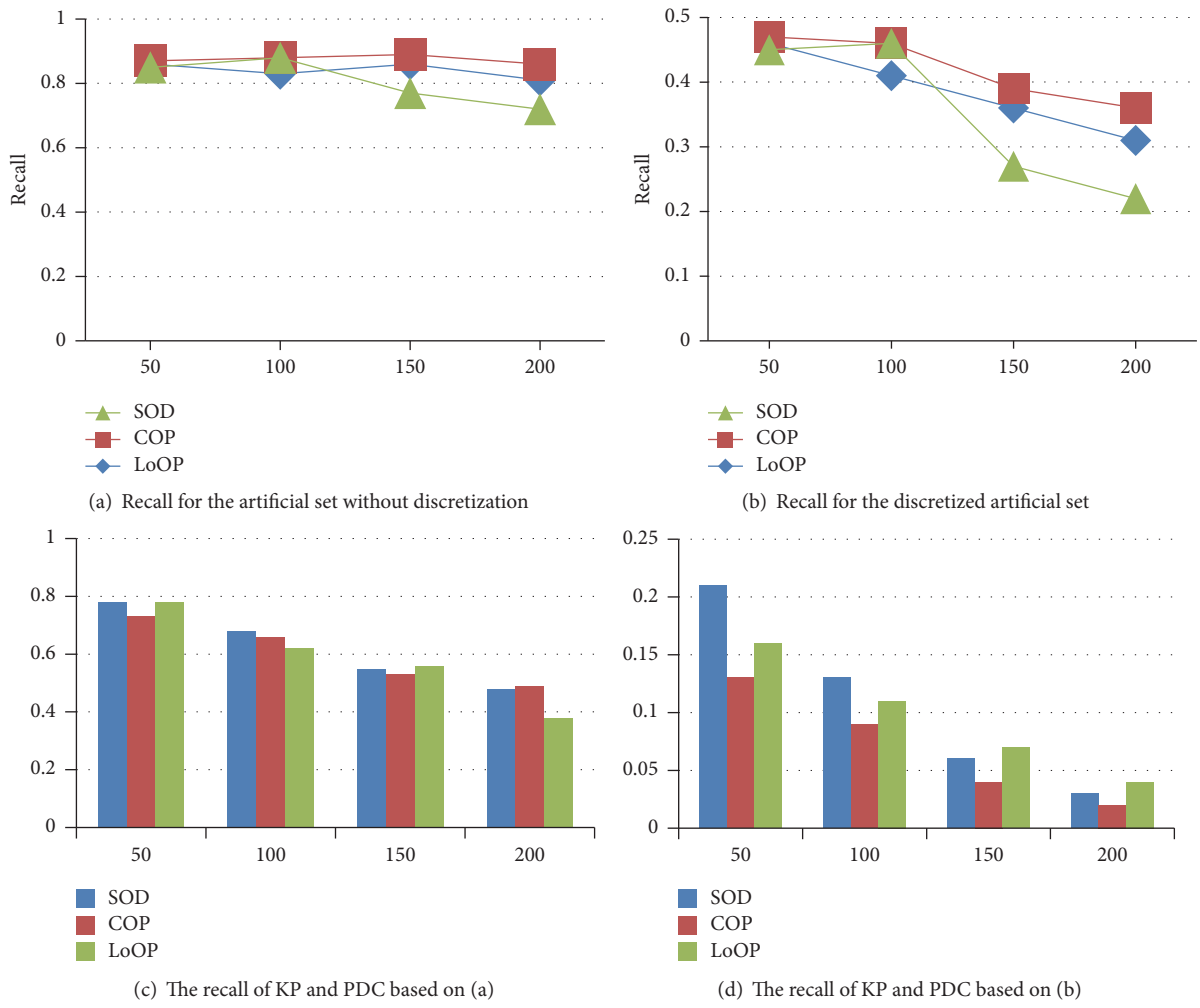
(d) The precision of KP and PDC based on (b)

Figure 10

internal business process BPEL adheres to an organization's privacy policies.

*(2) Matching and Negotiation of Privacy Policy.* Ke et al. [21] defined user and service provider privacy policies, respectively, on the basis of an analysis of current privacy rules and proposed an automatic privacy policy matching method, which can check the type of private data, the objective of private data disclosure, and the collector and maintenance period of private data. Wei et al. [22] researched private data protection policy in the application of pervasive computing, building privacy models and privacy policy axioms using many-sorted logic and description logic, and proposed a reasoned method for privacy policy which can check the inconsistency among policies. Ke et al. [23] proposed privacy negotiation methods between the user and the service provider in cloud computing, to achieve privacy requirements. Tbahriti et al. [24] presented a negotiation protocol pointing to inconsistencies in user and service provider privacy policies and put forward a privacy system, in which a privacy policy based on P3P can be defined. Zhang and Fen [25] set up a framework for parsimonious semantic trust negotiation, which can greatly reduce the degree of disclosed privacy identity information without exchanging entire attribute certificates.

*5.2. Data Privacy.* Kolter et al. [26] proposed a user-centered private data log tool, which provides a visual interface to display disclosed data in a past transaction. Liu et al. [27] provided a method of computing service trust degree and presented interface automata that can extend privacy semantics, by which a privacy model can be built for service composition behaviors. In the meantime, it also provided a transforming method from BPEL to privacy interface automata and verified if service composition behaviors meet authorized privacy constraints formalisms. Liu et al. [28] analyzed privacy disclosure and authorization in web service composition and used a minimum path algorithm to analyze the privacy authorization issue. Roy et al. [29] presented a privacy protection system Airavat, a novel integration of decentralized information flow control (DIFC) and differential privacy, which prevents unauthorized leakage of sensitive data during computation and supports automatic declassification of results when the latter do not violate individual privacy. Chou [30] analyzed privacy risk during the business




(a) Recall for the artificial set without discretization

(b) Recall for the discretized artificial set

(c) The recall of KP and PDC based on (a)

(d) The recall of KP and PDC based on (b)

Figure 11

process lifecycle in cloud computing and put forward an audit method. Sen et al. [31] presented specifications of privacy policies, tracked users' data flows, and built and operated a system to automate privacy policy compliance checking in Bing.

## 6. Conclusions and Future Work

In this paper, we advanced a classification method for privacy data. To protect user privacy information, continuous data chains are decomposed into discrete data chains, and discrete data chains are prevented from being composed into continuous data chains. We proposed a protection framework for privacy data, and we performed experiments to demonstrate its correctness and feasibility. Future work will focus on privacy extensions in service level agreements in order to support the monitoring of service composition.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant 61602262), the Jiangsu Natural Science Foundation of China (Grants BK20150865, BK20130735), the Jiangsu University Natural Science Foundation (Grants 15KJD520001, 13KJB520011), sponsored by NUPTSF (Grants nos. NY214164, NY215120), the China Postdoctoral Science Foundation Funded Project (Project no. 2016 M591842), and the Jiangsu province Postdoctoral Science Foundation Funded Project (Project no. 1601198C).

## References

[1] D. C. Chou, "Cloud computing: a value creation model," *Computer Standards & Interfaces*, vol. 38, pp. 72–77, 2015.

[2] M. Armbrust, A. Fox, R. Griffith et al., "Above the clouds: a Berkeley view of cloud computing," Tech. Rep. UCB-EECS-2009-28, University of California, Berkeley, Berkeley, Calif, USA, 2009.

[3] M. L. Damiani and C. Cuijpers, "Privacy challenges in third-party location services," in *Proceedings of the 14th International*

*Conference on Mobile Data Management (MDM '13)*, vol. 2, pp. 63–66, Milan, Italy, June 2013.

[4] Z. Liu, H. Yan, and Z. Li, "Server-aided anonymous attribute-based authentication in cloud computing," *Future Generation Computer Systems*, vol. 52, pp. 61–66, 2015.

[5] F. Casino, J. Domingo-Ferrer, C. Patsakis, D. Puig, and A. Solanas, "A k-anonymous approach to privacy preserving collaborative filtering," *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 1000–1011, 2015.

[6] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proceedings of the 34th IEEE Symposium on Security and Privacy (SP '13)*, pp. 334–348, May 2013.

[7] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "SoK: security and privacy in implantable medical devices and body area networks," in *Proceedings of the 35th IEEE Symposium on Security and Privacy (SP '14)*, pp. 524–539, San Jose, Calif , USA, May 2014.

[8] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams, "Hummingbird: privacy at the time of Twitter," in *Proceedings of the 33rd IEEE Symposium on Security and Privacy (SP '12)*, pp. 285–299, May 2012.

[9] S. Pearson, "Taking account of privacy when designing cloud computing services," in *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pp. 44–52, Vancouver, Canada, May 2009, Also available as HP Labs Technical Report, HPL-2009-54.

[10] S. Pearson and A. Charlesworth, "Accountability as a way forward for privacy protection in the cloud," HP Labs Technical Report HPL-2009-178, 2009.

[11] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in arbitrarily oriented subspaces," in *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM '12)*, pp. 379–388, December 2012.

[12] H. P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 831–838, Springer, 2009.

[13] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," in *Proceedings of the ACM 18th International Conference on Information and Knowledge Management (CIKM '09)*, pp. 1649–1652, ACM Press, November 2009.

[14] B. Tang, R. Sandhu, and Q. Li, "Multi-tenancy authorization models for collaborative cloud services," *Concurrency Computation*, vol. 27, no. 11, pp. 2851–2868, 2015.

[15] R. Hamadi, H. Y. Paik, and B. Benatallah, "Conceptual modeling of privacy-aware Web service protocols," in *In proceeding of the 19th International conference on Advanced Information System Engineering (CAiSE '07)*, pp. 233–248, 2007.

[16] N. Guermouche, S. Benbernou, E. Coquery, and M.-S. Hacid, "Privacy-aware Web service protocol replaceability," in *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS '07)*, pp. 1048–1055, July 2007.

[17] K. Mokhtari, S. Benbernou, M. Said, E. Coquery, M. S. Hacid, and F. Leymann, "Verification of privacy timed properties in web service protocols," in *Proceedings of the IEEE International Conference on Services Computing (SCC '08)*, pp. 593-594, July 2008.

[18] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Security-aware service composition with fine-grained information flow control," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 330–343, 2013.

[19] H. Tout, A. Mourad, H. Yahyaoui, C. Talhi, and H. Otrok, "Towards a BPEL model-driven approach for web services security," in *Proceedings of the 10th Annual International Conference on Privacy, Security and Trust (PST '12)*, pp. 120–127, IEEE Computer Society, July 2012.

[20] Y. H. Li, H. Paik, and B. Benatallah, "Formal consistency verification between BPEL process and privacy policy," in *Proceedings of the International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services (PST '06)*, pp. 212–223, Markham, Canada, October 2006.

[21] C. Ke, Z. Huang, W. Li, Y. Sun, and F. Xiao, "Service outsourcing character oriented privacy conflict detection method in cloud computing," *Journal of Applied Mathematics*, vol. 2014, Article ID 240425, 11 pages, 2014.

[22] Z.-Q. Wei, M.-J. Kang, D.-N. Jia, B. Yin, and W. Zhou, "Research on privacy-protection policy for pervasive computing," *Chinese Journal of Computers*, vol. 33, no. 1, pp. 128–138, 2010.

[23] C. Ke, Z. Huang, and M. Tang, "Supporting negotiation mechanism privacy authority method in cloud computing," *Knowledge-Based Systems*, vol. 51, pp. 48–59, 2013.

[24] S.-E. Tbahriti, C. Ghedira, B. Medjahed, and M. Mrissa, "Privacy-enhanced web service composition," *IEEE Transactions on Services Computing*, vol. 7, no. 2, pp. 210–222, 2014.

[25] Y. Zhang and D.-G. Fen, "Parsimonious semantic trust negotiation," *Chinese Journal of Computers*, vol. 32, no. 10, pp. 1989–2003, 2009.

[26] J. Kolter, M. Netter, and G. Pernul, "Visualizing past personal data disclosures," in *Proceedings of the 5th International Conference on Availability, Reliability, and Security (ARES '10)*, pp. 131–139, February 2010.

[27] L. Liu, H. Zhu, Z. Huang, and D. Xie, "Minimal privacy authorization in web services collaboration," *Computer Standards and Interfaces*, vol. 33, no. 3, pp. 332–343, 2011.

[28] L. Liu, H. Zhu, and Z. Huang, "Analysis of the minimal privacy disclosure for web services collaborations with role mechanisms," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4540–4549, 2011.

[29] I. Roy, H. E. Ramadan, S. T. V. Setty et al., "Security and privacy for MapReduce," in *Proceedings of the 7th Usenix Symposium on Networked Systems Design and Implementation*, M. Castro, Ed., pp. 297–312, USENIX Association, 2010.

[30] D. C. Chou, "Cloud computing risk and audit issues," *Computer Standards and Interfaces*, vol. 42, pp. 137–142, 2015.

[31] S. Sen, S. Guha, A. Datta, S. K. Rajamani, J. Tsai, and J. M. Wing, "Bootstrapping privacy compliance in big data systems," in *Proceedings of the 35th IEEE Symposium on Security and Privacy (SP '14)*, pp. 327–342, May 2014.