Risk Evaluation for Personal Identity Management Based on Privacy Attribute Ontology

Mizuho Iwaihara[1], Kohei Murakami[1], Gail-Joon Ahn[2], and Masatoshi Yoshikawa[1]

[1] Department of Social Informatics, Kyoto University, Japan
kmurakami@db.soc.i.kyoto-u.ac.jp, iwaihara@i.kyoto-u.ac.jp, yoshikawa@i.kyoto-u.ac.jp
[2] Department of Computer Science and Engineering, Arizona State University, USA
gahn@asu.edu

Abstract. Identity providers are becoming popular for distributed authentication and distributed identity management. Users' privacy attributes are stored at an identity provider and they are released to a service provider upon user's consent. Since a broad range of privacy information of different sensitiveness can be exchanged in advanced web services, it is necessary to assist users by presenting potential risk on financial and personality damage, before releasing privacy attributes. In this paper, we present a model of privacy attribute ontology and risk evaluation method on this ontology. Then we formalize several matching problems which optimize similarity scores of matching solutions under several different types of risk constraints. We show sophisticated polynomial-time algorithms for solving these optimization problems.

1 Introduction

A wide variety of new services are created on the web, by connecting existing web services. To carry out services and/or businesses with their customers, many of service providers (SP) require basic personal information of customers, such as name, address, phone number, as well as more critical information such as credit card number. Identity providers (IdPs) offer identity management functionalities, including user authentication and management of basic personal information. Since basic information such as name and email/postal addresses are frequently asked, provisioning of these information from IdP to SP through the user's one-click action can save the user's workload. Liberty Alliance[10], OpenID[11] and CardSpace[1] are proposed identity management standards which provide single sign-on and trust management. However, in these standards, users are still required to carefully examine requested attributes for sensitiveness and criticality. Then users select appropriate identities to be used for the request, where excessive exposure of identities and attributes should be avoided by users' discretion.

Web services are rapidly evolving to cover every kind of social activities among people, and categories of personal attributes are also growing beyond basic attributes. Social network services are offering exchange of very personal attributes such as such as age, ethnicity, religion, height and eye color. For example, orkut(www.orkut.com)

Q. Li et al. (Eds.): ER 2008, LNCS 5231, pp. 183–198, 2008.

has an registration form having 30 attributes for "social" page, 16 attributes for "professional" page, and 15 attributes for "personal" page. User-centric control of sharing of personal information is required for healthy support of social activities, and an identity provider of the near future should assist the user through categorization and evaluation of attributes from the point of criticality and sensitiveness.

In this paper, we propose the concept of *privacy attribute ontology* (PAO), built on the OWL web ontology language[12]. One of primal objectives of PAO is to provide a taxonomy of privacy attributes. Each class of PAO corresponds to a sensitive attribute or an identity, and an individual of the class corresponds to a value of the attribute. IdP manages a PAO as a shared ontology among users as well as a personal information database for each user. Also PAO provides risk evaluation functionality through financial and personality risk values defined on PAO classes. When a service provider presents a list of requested attributes, IdP matches the list with PAO classes, and then the risk values of the requested attributes are evaluated from matched classes. Here we have a number of issues to be solved. First, we need to design a matching algorithm that maximizes linguistic/structural similarities between PAO classes and requested attributes. Secondly, the algorithm also needs to consider risk constraints such that matched classes must not exceed given upper limits of risk values. The algorithm should select a low-risk combination of identities and attributes associated to these identities, covering requested attributes. In this optimization, we need to consider combination risks which arise if a certain combination of classes is selected for release.

The contribution of this paper is summarized as follows: (1) We present a model of privacy attribute ontology and risk evaluation method on this ontology. (2) We formalize matching problems which optimize similarity scores of matching solutions under three different types of risk constraints. (3) We show sophisticated polynomial-time algorithms for solving the optimization problems of (2).

P3P (Platform for Privacy Preferences Project) [14] is a standard for describing and exchanging privacy policies in XML format. While P3P is targeted at interpreting privacy practices of service providers, our research is focused on identity providers and users for managing linkages between privacy attributes and identities of different aspects.

Developing ontologies for privacy and trust management on the web has been discussed in the literature[4][5][7]. Our research is different in the way that we focus on risk evaluation for attribute disclosure and selecting disclosing attribute values (individuals) that have minimum risk values. Utilizing semantic web technologies for security and trust management on the web is discussed in [4], which covers authentication, delegation, and access control in a decentralized environment. But an ontology for assessing privacy risk values is not considered.

Matching and aligning ontologies have been extensively studied for integrating ontologies. As a linguistic approach, OntoGenie[13] uses WordNet[16] for extracting ontologies from web pages. Structural similarity is considered in [9] for neural network-based schema matching. Udrea et al.[15] combined data and structural matching as well as logical inference to improve quality. Our algorithms utilize these linguistic and structural approaches. But we need to deal with the new problem of considering risk values during matching. We have successfully solved ontology matching under various types of risk constraints.

The rest of the paper is organized as follows. In Section 2, we introduce an existing risk evaluation method for privacy information, and discuss automated risk evaluation based on privacy attribute ontology. In Section 3, we formalize privacy attribute ontology. In Section 4, we discuss matching requested attributes with PAO classes, and define optimization problems under certain risk constraints. In Section 5, we discuss several issues that need to be solved, and present polynomial-time algorithms for the optimization problems. Section 6 is a conclusion.

2 Risk Evaluation for Personal Identity Management

2.1 JNSA Privacy Risk Evaluation

Service providers holding customer's privacy data are having risk of privacy leakage. Several measures for evaluating risk of privacy leakage have been proposed. Japan Network Security Association (JNSA) published surveys on information security incidents[6]. The report also presents a method for estimating amount of compensation if a certain portion of privacy data are leaked. The JNSA model is based on classifying reported cases from court decisions and settlements, and the model was validated on these cases. Its evaluation proceeds as follows:

The value of leaked privacy data of an individuation is evaluated in terms of (a) economical loss and (b) emotional pain. The Simple-EP Diagram contains representative privacy attributes according to the dimensions of (a) and (b). Given an attribute, an integer from 1 to 3 is chosen as the value for each dimension. Let x (resp. y) be the value for (a) economical loss (resp. (b) emotional loss). Then the *sensitiveness factor* is defined as $EP = (10^{x-1} + 5^{y-1})$.

Given a collection of privacy attributes for an individual, we take maximum values for x and y from the Simple-EP Diagram. Suppose a record of an individual consists of the attributes: real name, address, birth date, sex, phone, medical diagnosis, bank account and password. Then by the Simple-EP Diagram, the value (x, y) is equal to $(1, 1)$ for real name, address, birth date, sex, and phone. On the other hand (x, y) is equal to $(2, 1)$ for medical diagnosis, and (1, 3) for bank account and password. Since the maximum value for x is 2 and the maximum value for y is 3, we obtain $EP = 35$.

Let the *basic information value* BIV be 500 points, and let the *identifiability factor* IF be defined as: $IF = 6$ if the individual can be easily identified (for example, real name and address are included), $IF = 3$ if the individual can be identified by a certain effort (for example, real name is included, or address and phone are included), and $IF = 1$ otherwise (for the case identification is difficult). The leaked privacy information value $LPIV$ is computed by: $LPIV = BIV * EP * IF$. LPIV is designed to approximate the amount of compensation in Japanese yen paid to each leakage victim. The LPIV is further adjusted to reflect other factors such as the social status of the information holder and evaluation on the response after the incident. However, these factors are not directly related to our goal.

The JNSA risk evaluation model can be a basis of risk evaluation for risk-aware identity management, from the points that the model can capture the emotional and financial losses according to a classification of privacy attributes, and it enables quantitative

comparison of the risks between attributes. However, the method requires human reasoning in determining values from the diagram.

2.2 Risk Evaluation at Identity Provider

The basic scenario of personal information management by an identity provider (IdP) utilizing PAO proceeds as follows:

1. IdP manages and holds personal information of the user.
2. The user requests execution of a service to the service provider (SP). SP sends to IdP *requested attributes* $\mathcal{RA}$ necessary for the service. $\mathcal{RA}$ includes basic identity information as well as privacy attributes of the user.
3. IdP matches attributes of $\mathcal{RA}$ with classes of PAO, to compute *releasing classes* $\mathcal{RC}$. In the matching process, IdP evaluates risks of releasing information held in $\mathcal{RC}$, and IdP tries to find $\mathcal{RC}$ which has maximum conceptual similarities with $\mathcal{RA}$, while $\mathcal{RC}$ satisfies a certain risk constraint imposed by the user.
4. $\mathcal{RC}$ is presented to the user. The user modifies and supplements $\mathcal{RC}$ if necessary. Some requested attributes A may not be included in $\mathcal{RC}$, because either A's risk is intolerable to the user or the user has declined release of of A. After SP and the user agree on $\mathcal{RC}$, the information on $\mathcal{RC}$ is sent from IdP to SP.

IdP manages a number of identities of the user, such as student ID, a number of email addresses, citizenship, net identities used for blogs and social network services. Some of these identities are anonymous, while others have solid identities. One identity is associated with a number of attributes, as well as other identities. In selecting $\mathcal{RC}$, IdP needs to find low-risk combination of attributes and avoid linking of identities if it is prohibited by the user.

2.3 Risk Evaluation Using Privacy Attribute Ontology

In the following, we summarize the basic notions of our risk evaluation method utilizing PAO.

Risk value is a numerical scale of 1 to 5 representing severity of the risk, where 1 is least severe and 5 is most severe. Risk values are categorized into financial and personality risk values. PAO holds risk values in its classes. However, some classes may not have risk values defined. If a risk value of C is undefined, then the risk value is inherited from C's super classes. If a class C is in the releasing class $\mathcal{RC}$, then the risk values of C become *effective*. The risk value of releasing classes $\mathcal{RC}$ is the maximum effective risk value in the classes of $\mathcal{RC}$.

Financial risk value (f-risk value for short) is a risk value for financial damage to the information subject (user). Credit card number, bank account number, and social security number should have high financial risk values. We use $r_f(\cdot)$ to denote the financial risk value function on various constructs such as class C and releasing classes $\mathcal{RC}$.

Personality risk value (p-risk value for short) is a risk value for personality damage to the user, including emotional pain, damage to social reputation, and generic damage caused by privacy breach. We use $r_p(\cdot)$ to denote the personality risk value function.

Combined risk value $r_c(\mathcal{RC})$ combines f-risk and p-risk values by the function $r_c(\mathcal{RC}) = cr(r_f(\mathcal{RC}), r_p(\mathcal{RC}))$ such that $cr(x, y) = c_1 \log(\mathcal{F}^x + \mathcal{P}^y) + c_2$, where the risk values x and y are converted into an exponential scale by the exponential functions of bases $\mathcal{F}$ and $\mathcal{P}$, and the average of these values are converted back to risk values by the logarithmic function. The bases $\mathcal{F}$ and $\mathcal{P}$ assign weights between the financial and personality risk values, and we can choose $\mathcal{F} = 10$ and $\mathcal{P} = 5$ following the JNSA model. Constants c_1 and c_2 shall be determined to let $cr(x, y)$ have a range between 1 and 5.

Combination risk is a risk arising from combination of attributes. Some privacy attributes, such as age and income, may be disclosed under an anonymous username, but combining these attributes with the real name raises the risk of privacy breach. Thus the user should be notified of such high risk combination. Also, the user holding a number of identities at IdP can choose one identity or a combination of identities to cover requested attributes. In this scenario, the user should be advised of the risk in linking several identities. For modeling combination risks, we need to introduce *combination risk classes* to PAO.

Risk limit is a given upper limit on f-risk, p-risk or combined risk values. If the user gives his/her tolerable risk limit, then disclosing attributes should not exceed the limit. Here exists an optimization problem for finding most-similar matching between the PAO classes and requested attributes, while satisfying the risk limit. Trustability of service providers can be reflected to risk limits, in a way that when dealing with a questionable service provider, the user can define a lower, more cautious risk limit. Detailed linkage between risk limits and existing trustability models is beyond the scope of the paper.

3 Modeling Privacy Attribute Ontology

In this section, we formalize privacy attribute ontology. We follow the definitions of OWL[12] as the underlying ontology model. A *class* represents a concept. A class is associated with zero or more *individuals* belonging to that class. An ontology can be represented as a directed graph, where nodes are labeled with a class name or an individual, and directed edges are labeled with link types. A link labeled `type` from an individual to a class represents the membership relation between the individual and the class. A link labeled `subClassOf` from class C_1 to class C_2 indicates that C_1 is a subclass of C_2 meaning that C_1 is a concept more specific than C_2 and an individual belonging to C_2 also belongs to C_1. A link labeled `partOf` from class C_1 to class C_2 indicates that C_2 is a *composite class* composed of a number of *component classes*, including C_1. Formally, if a class C_1 is connected to a class C_2 through a directed path of `partOf` and `subClassOf` links, then C_1 is a component class of C_2. `partOf` links are not allowed to form a directed cycle. We define *composite attributes* for requested attributes, similarly to composite classes.

PAO has two special link types named `financialRisk` and `personalityRisk`, representing the financial risk value $r_f(C)$ and and personality risk value $r_f(C)$ of a class

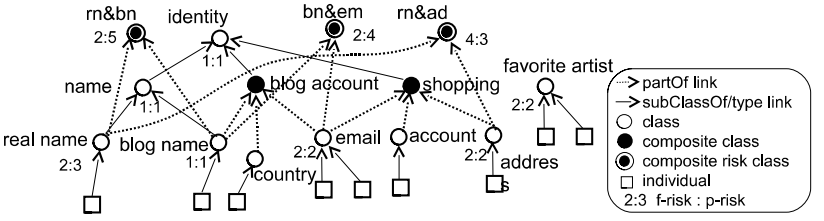

Fig. 1. Privacy attribute ontology

C, leading to individuals of real numbers in the range [1.0, 5.0]. An example of privacy attribute ontology is shown in Figure 1, where risk values are shown as numbers of the form $r_f : r_p$. Also, composite classes are depicted as black circles.

In PAO, we assume that each individual belongs to a single class. For an individual i belonging to multiple classes, we can insert a virtual class between i and these classes, to satisfy the single-class restriction. Thus this is not a tight restriction. Also, if some risk values need to be defined on particular individuals, we create a class for such an individual, and let all the risk values be defined on classes.

As discussed in Section 2.3, we introduce a *combination risk class*, which is a composite class connected by `partOf` links from its component classes. In Figure 1, combination risk classes are depicted as double circles. The risk value of a combination risk class is applied if all of its component classes are selected for release. For example, the class `rn&bn` represents that if `real name` and `blog name` are going to be released, then its risk values 2:5 will be applied. These values are higher than that of classes `real name` and `blog name` alone, indicating that combination of these classes increase the risk values, or it can be interpreted that the user is not allowing linking of these identities. Thus a combination risk class should have f-risk and p-risk values no less than that of its component classes.

PAO can be shared by a group of users so that the users' common knowledge on risks can be reflected. However, each user may have different views on privacy, and individuals in the ontology are also user-dependent. Thus personalization of PAO is necessary. Personalization of PAO can be done by the following ways: (a) overriding financial and/or personality risk values of a class, (b) adding individuals to a class, and (c) adding a class as a subclass of an existing class. Sharing and personalization of PAO is beyond the scope of this paper, so we do not elaborate on this direction any further.

4 Matching PAO and Requested Attributes

4.1 Matching Problems

Now we discuss evaluating risk of a set $\mathcal{RA}$ of requested attributes sent by a service provider, utilizing PAO. Then using the risk evaluation method, we consider optimization problems to find an optimum combination of releasing individuals that achieves given risk constraints.

For associating individuals of PAO and requested attributes $\mathcal{RA}$, we consider the following two-staged approach: First find a bipartite matching between classes of PAO

and $\mathcal{RA}$, then choose an individual from each class selected by the matching. A bipartite matching finds a one-to-one mapping between classes and $\mathcal{RA}$. Since we assumed that each individual belongs to a single class, this process is straightforward.

We introduce *similarity score* $\sigma(C, A) \geq 0$ on a PAO class C and a requested attribute $A \in \mathcal{RA}$. When $\sigma(C, A) > \beta$ holds for a given lower threshold β, C and A are regarded as distinct concepts. We discuss construction of σ by linguistic similarities in Section 4.2. We construct a *matching graph* $G_{\sigma,\beta} = (\mathcal{C}, \mathcal{RA}, E)$ which is a bipartite graph such that $\mathcal{C}$ is the set of classes in PAO, $\mathcal{RA}$ is the set of requested attributes, and E is the set of edges (C_i, A_j) such that $C_j \in \mathcal{C}, A_j \in \mathcal{RA}$, and $\sigma(C, A) > \beta$ is true. We also use the similarity function σ for edge weights of the bipartite graph $G_{\sigma,\beta}(\mathcal{C}, \mathcal{RA}, E)$. The weighted bipartite matching problem can be solved in $O(N^3)$ time by the Hungarian method [8], where N is the number of nodes in $G_{\sigma,\beta}$. A *matching* M on bipartite graph $G_{\sigma,\beta} = (\mathcal{C}, \mathcal{RA}, E)$ is a bipartite subgraph $(\mathcal{C}_M, \mathcal{RA}_M, E_M)$ such that $CC_M \subseteq \mathcal{C}$, $\mathcal{RA} \subseteq \mathcal{RA}_M$, $E_M \subseteq E$, and no edge in E_M conflicts each other, that is, any two edges in E_M are not adjacent at either end. Let $\sigma(M)$ denote the sum of the edge weights of E_M. A matching M on $G_{\sigma,\beta}$ is a *maximum matching* if $\sigma(M) \geq \sigma(M')$ holds for any matching M' on $G_{\sigma,\beta}$.

Figure 2 shows an example of matching graphs. The nodes on the left are classes of PAO, and the nodes on the right are requested attributes. Here, '+' sign means a composite class or attribute, and '-' sign means a component class or attribute. Edge weights are not displayed in the graph. A matching is shown as bold edges in Figure 2. Notice that this matching includes edges (`email, e-mail`) and (`address, address`). These associations may appear reasonable, but unacceptable because structural integrity is ignored. The `email` class of PAO is a component of class `blog account`, while `address` of PAO is a component of class `shopping`. These composite classes represent distinct identities, and component classes should not be intermixed. Intuitively, a proper matching should preserve component-composite relationships. In Section 5.1, we discuss this component integrity and present a solution. We note that combination risk classes should be excluded from matching candidates, because they are just for internally defining combinational risk values.

Recall that the combined risk value is determined by maximum f-risk value r_f and p-risk value r_p found in releasing classes. In a matching $M = (\mathcal{C}_M, \mathcal{RA}_M, E_M)$, the set of matched classes $\mathcal{C}_M$ is the releasing classes. Let $r_f(M)$ and $r_p(M)$ be the maximum f-risk and p-risk values in M, respectively. Then the combined risk value $r_c(M)$ is computed by $cr(r_f(M), r_p(M))$.

Requested attributes $\mathcal{RA}$ may not have any matchable class in $\mathcal{C}$. Such a dangling attribute can be reported by the matching algorithm. In this case, the system needs to start a dialog with the user to create a new class in PAO for the attribute.

By using predefined parameters on risk limits and similarity score limits, a number of optimization problems can be defined:

1. **(similarity score maximization)** Tolerable upper risk limits m_f and m_p are given by the user, where $m_f > 0$ and $m_p > 0$ are *maximum f-risk value* and *maximum p-risk value*, respectively. The optimization problem is to find a matching M such that similarity score $\sigma(M)$ is maximum and $r_c(M) \leq cr(m_f, m_p)$ holds. The user may specify $m_f = \infty$ and/or $m_p = \infty$ if he/she does not restrict one or both of

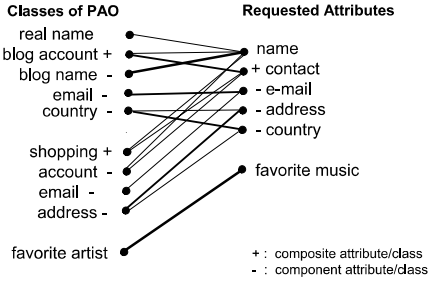

Fig. 2. Matching graph

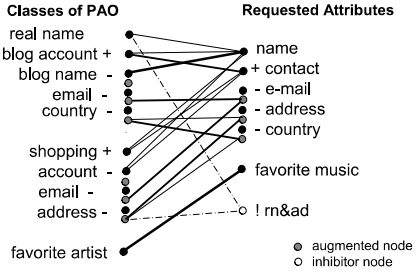

Fig. 3. Augmented matching graph

the risk values. Another version of the problem is **similarity score maximization under combined-risk limit** m_c, which is to find a maximum matching under the constraint $r_c(M) \leq m_c$. In this case, we need to test varying combinations of m_f and m_p that satisfy $cr(m_f, m_p) \leq m_c$.

2. **(risk minimization)** This problem assumes that a lowerbound w_{min} for the total similarity score is given. The problem is to find a matching M such that $\sigma(M) > w_{min}$ and the combined risk value $r_c(M)$ is minimum.
3. **(combined score maximization)** Let $s_c(M)$ be *combined total score* defined by $s_c(M) = \sigma(M)/r_c(M)$. The problem is to find a matching M that maximizes $s_c(M)$. Unlike others, this problem does not have a predefined parameter.

Since specifying all the risk values is tedious, it is likely that some classes or individuals may not be given risk values in PAO. For this issue, we utilize subClassOf links of PAO for inferring risk values, based on the principle that a subclass inherits a missing property value from its parent. Here we also adopt the principle of taking the highest possible risk value, for conservative risk estimation.

(subClassOf rule) Let C be a class in PAO such that $r_f(C)$ is undefined. Let $C_1, \ldots, C_k$ be classes in PAO such that there is a path S_i of `subClassOf` links from C to each C_i and C_i is the only class in P_i where $r_f(C_i)$ is defined. Then let $r_f(C) := \max(r_f(C_1), \ldots, r_f(C_k))$. For the case $r_p(C)$ is undefined, simply replace r_f with r_p.

Applying the subClassOf rule to every class that has an undefined risk value gives us a unique PAO that has no undefined risk value, and this procedure can be done in linear time.

4.2 Linguistic Similarity

We use Jaro-Winkler score[2] for string similarity and WordNet similarity [16] for score on synonymity. Jaro-Winkler is effective for matching pairs having common substrings, such as between e-mail, email, Email and netmail. WordNet is a lexical dictionary, where words are ground into synonyms (synsets), each synset expressing a distinct concept. WordNet similarity is measured by conceptual-semantic and lexical relations between synsets. We use the sum of Jaro-Winkler score and WordNet score as the similarity score σ between PAO classes and $\mathcal{RA}$.

Table 1 shows an experimental result on matching names of PAO classes and attributes from web sign-up forms. We constructed a PAO containing 186 classes. The class names of this PAO are matched with two attribute sets from web sign-up forms of eBay and PayPal, using the above similarity score σ. Case 2 is matches detected by string similarity, which included pair "Primary telephone number" and "Primary telephone". Case 3 is matches detected by synonymity, which included pair "Secret Question" and "Security Question". Overall, the similarity score σ is showing enough accuracy for matching requested attributes with PAO.

Table 1. Linguistic matching result on PAO having 186 classes

		eBay	PayPal
case	total attributes	17	23
1	string match with PAO classes	7	15
2	attributes matched by string similarity(Jaro-Winkler)	3/3	6/6
3	attributes matched by synonymity score (WordNet) (Excluding case 2)	3/5	1/2
4	attributes having no matching class in PAO	2	0

(detected matches)/(correct matches)

5 Matching Algorithms

5.1 Component Integrity and Two-Level Matching

First we formalize component integrity, and present a matching algorithm that achieves a certain type of component integrity while maximizing similarity score. At this moment, we assume that $m_f = \infty$ and $m_p = \infty$ hold, namely no constraint is given on risk values. We also assume that the PAO has no combination risk class. We extend the algorithm later in this section.

(**component integrity**) Let us consider a matching M between PAO classes $\mathcal{C}$ and requested attributes $\mathcal{RA}$. A matching M is said to satisfy *component integrity*, if the following holds: Let (C, A) and (D, B) be any pair of edges in M such that $C, D \in \mathcal{C}$ and $A, B \in \mathcal{RA}$. Then C is a component class of D if and only if A is a component attribute of B.

Note that PAO can have a multi-level component class, i.e., a composite class can be a component class of another class. PAO can also include a component class shared by multiple composite classes. In such a DAG-structured PAO, imposing the above component integrity becomes a hard problem:

Theorem 1. *Given a bipartite graph $G_{\sigma,\beta} = (\mathcal{C}, \mathcal{RA}, E)$ and a minimum weight w, deciding whether $G_{\sigma,\beta}$ has a matching M having weight $\sigma(M) > w$ and satisfying component integrity is NP-complete.*

Proof. (sketch) Transformation from SET PACKING[3].

Thus it is intractable to enforce the above composite integrity. Also this integrity does not consider link connectivities. In object-oriented modeling, link connectivities are often used to add different perspectives to a class. For example, consider the following subgraphs containing the class email in Figure 1: email → blog account, email → shopping, and email. Note that the last subgraph is a singleton node. These subgraphs represent e-mail of the blog account, email of the shopping identity, and emails of the person, respectively. Thus each subgraph is representing a different concept.

Now let us consider the following multi-level nesting of composite classes for matching: A class C is a *level-k component class* of D if (1) for $k = 1$, C is a component class of D, and (2) for $k > 1$, C is a component class of a level-$(k-1)$ component class of D.

We can adopt the interpretation such that for each different k, each path from a composite class to its level-k component class represents a distinct concept. To treat these paths as distinct concepts in matching, new nodes shall be created for each path for varying k. However, since PAO can have shared component classes, the number of such paths can be exponential to k. Thus considering all the paths to level-k component classes as matching candidates is impractical. In the following, we resrict level k to be 1, and augment the matching graph $G_{\sigma,\beta}$ with new nodes representing pairs of composite classes and their level-1 component classes. For each composite class D and component class C, we create a new node labeled with the concatenation $D.C$. Likewise, we create a new node labeled with the concatenation $B.A$ for composite attribute B and composite attribute A. Formally, let $G^a_{\sigma,\beta} = (\mathcal{C}^a, \mathcal{RA}^a, E^a)$ be the bipartite such that $\mathcal{C}^a = \mathcal{C} \cup \{D.C \mid D, C \in \mathcal{C}, C \text{ is a component class of } D\}$, $\mathcal{RA}^a = \mathcal{RA} \cup \{B.A \mid B, A \in \mathcal{RA}, A \text{ is a component attribute of } B\}$. The edge set E^a is obtained by adding edge $(D.C, B.A)$ to E for each new class $D.C$ and new attribute $B.A$ satisfying $\sigma(D.C, B.A) > \beta$, and removing edge (C, A) from E where C and A are component class and attributes, respectively. Here we remove the edge (C, A) because it will be represented by the new component-level edge $(D.C, B.A)$. We call $G^a_{\sigma,\beta}$ a *composite-augmented graph*. Also, we call a bipartite matching M^a on $G^a_{\sigma,\beta}$ an *augmented matching*.

For a class C shared by composite classes $D_1, \ldots, D_m$ in $G_{\sigma,\beta}$, $G^a_{\sigma,\beta}$ has duplicated nodes $C, D_1.C, \ldots, D_m.C$. Thus an augmented matching M^a can include one or more nodes from $C, D_1.C, \ldots, D_m.C$ in its edges. The following realizes integrity of level-1 component classes in augmented matching:

(**augmented component integrity**) An augmented matching M^a is said to satisfy *augmented component integrity*, if the following holds: Let $(D.C, B.A)$ and (D_1, B_1) be any pair of edges in M^a such that $D_1, C_1 \in \mathcal{C}$, $D.C \in \mathcal{C}^a$, $A, B \in \mathcal{RA}$, and $B.A \in \mathcal{RA}^a$. Then $D_1 = D$ holds if and only if $B_1 = B$ holds.

To satisfy augmented component integrity, we divide the matching of PAO and $\mathcal{RA}$ into two phases: First, we take each composite class D and each composite attribute B and solve matching between the component classes and attributes of D and C, and then augment the (linguistic) similarity score $\sigma(D, B)$ with the matching score **(component-level matching)**. Secondly, we solve matching between the component classes and component attributes using the augmented scores **(composite-level matching)**. The matching algorithm PAOMatch is shown in Figure 4.

1. For each class D in $\mathcal{C}$ and for each attribute B in $\mathcal{RA}$, compute augmented score $\sigma^a(D, B)$ as follows:
1.1 If either D or B is not a composite class/attribute, then let $\sigma^a(D, B) = \sigma(D, B)$ and goto Step 1.
1.2 /* Now D is a composite class and B is a composite attribute. */
Let C_i $(i = 1, \ldots, k)$ be the component classes of D. Let A_j $(j = 1, \ldots, m)$ be the component attributes of B.
1.3 Let G_{DB} be the bipartite graph such that its two node sets are $\{D.C_i\}$ and $\{B.A_j\}$, respectively, and each edge $(D.C_i, B.A_j)$ has augmented weight $\sigma^a(D.C_i, B.A_j)$. If $\sigma^a(D.C_i, B.A_j)$ is undefined for some i and j, then recursively apply Step 1.1-1.4 to obtain $\sigma^a(D.C_i, B.A_j)$.
1.4 Solve weighted bipartite matching on G_{DB} to obtain matching M_{DB} and its total maximum weight w_{DB}. Let $\sigma^a(D, B) = \sigma(D, B) + \lambda \cdot w_{DB}$. Here, $0 < \lambda < 1$ is a pre-defined damping factor.
2. /* Now $\sigma^a(D, B)$ is defined for each D and B. Note that $G_{\sigma^a,\beta}$ does not include augmented nodes. */
Solve weighted bipartite matching on $G_{\sigma^a,\beta}$, where edge weight σ is replaced by σ^a, and obtain matching M.
3. Construct solution matching M^a as follows: For each matching edge (D, B) in M, add the matching M_{DB} obtained at Step 1.3 to M.

Fig. 4. PAOMatch: Two-phased structural matching

Step 1 of PAOMatch computes maximum matching for each component class-attribute pair. Then the resulting weight w_{DB} is added to the linguistic similarity score $\sigma(D, B)$, to reflect structural similarity of the components of D and B (Step 1.4). Here, damping factor $0 < \lambda < 1$ is introduced to reflect the nesting level of component hierarchy. A component class or attribute far from its composite root will have a reduced influence to the score.

After solving maximum matching for each composite class and each composite attribute, the top-level matching is carried out (Step 2). Here, we use $G^a_{\sigma,\beta}$ to exclude component classes and component attributes, since component-level matching is already done at Step 1.

Figure 3 shows application of PAOMatch. Gray nodes are augmented nodes created for each component class/attribute at Step 1.3 of PAOMatch. At Step 1.4, Component-level matching is done between the augmented nodes of composite classes {`blog account`, `shopping`} and attribute {`contact`}. Using the scores of these matchings, composite-level matching is carried out (Step 2). In Figure 3, edge (`blog account`, `contact`) is chosen as one of the four composite-level edges. Thus edges (`blog account.email`, `contact.e-mail`) and (`blog account.country`, `contact.country`) are added at Step 3, as the result of component-level matching. On the hand, although component-level edges (`shopping.email`, `contact.e-mail`), (`shopping.address`, `contact.address`) are matched at component-level matching, they are eventually discarded because their parents `shopping` and `contact` are not matched. Notice that `blog name` is matched to `name` at the composite level, not as the composite class `blog account.blog name`.

Theorem 2. *For a matching graph $G_{\sigma,\beta} = (\mathcal{C}, \mathcal{RA}, E)$, let N be the number of nodes and E be the number of* `partOf` *links in $G_{\sigma,\beta}$. Then* PAOMatch *returns a maximum matching satisfying augmented component integrity in $O(N^3 + E^3)$ time.*

Proof. For augmented component integrity, suppose that augmented matching M^a includes edges $(D.C, B.A)$ and (D_1, B_1) such that $D_1, C_1 \in \mathcal{C}$, $D.C \in \mathcal{C}^a$, $A, B \in \mathcal{RA}$, and $B.A \in \mathcal{RA}^a$. Now, assume that $D_1 = D$ holds. Since $D.C$ is an augmented node, the edge $(D.C, B.A)$ must be added at Step 3 of PAOMatch as one of the edge in M_{DB}. Since matching at Step 2 guarantees that (D_1, B_1) is the only edge in M^a that is adjacent to $D_1 = D$, the composite attribute B of M_{DB} must be B_1. The only-if part can be shown by a symmetric argument.

For the time bound, first consider Step 1 of PAOMatch. Let $\mathcal{D}$ be the set of composite classes in $\mathcal{C}$, and let $\mathcal{B}$ be the set of composite attributes in $\mathcal{RA}$. Weighted bipartite matching is executed at Step 1.4 for each $D \in \mathcal{D}$ and for each $B \in \mathcal{B}$. Let $|D|$ (resp. $|B|$) denote the number of component classes of D (resp. component attributes of B). Then one execution of Step 1.4 takes $O((|D| + |B|)^3)$ time. The total time of Step 1.4 is bounded by $\sum_{D\in\mathcal{D},B\in\mathcal{B}}(|D|+|B|)^3 \leq (\sum_{D\in\mathcal{D}}|D|+\sum_{B\in\mathcal{B}}|B|)^3 = E^3$. For Step 2, bipartite matching is performed on $G_{\sigma^a,\beta}$, which has N nodes. Thus Step 2 takes $O(N^3)$ time. Step 3 can be done in $O(N + E)$ time. □

5.2 Combination Risk Class and Inhibitor

Now consider combination risk classes. A combination risk class D_r is a composite class having component classes $C_1, \ldots, C_k$, where C_i is a class in $\mathcal{C}$ or component-composite classes, and the risk values $r_f(D_r)$ and $r_p(D_r)$ are given. These risk values are applied when and only when all of D_r's component classes are selected in an augmented matching M^a. Thus combination risk classes can express high-risk combination of privacy attributes.

Let us consider similarity score maximization where tolerable maximum limits are imposed on f- and/or p-risk values, as we discussed in Section 4.1. If D_r exceeds the risk limit, selecting all the component classes of D_r should be prohibited in the matching. Now let $\mathcal{D}_r$ be the subset of combination risk classes such that $D_r \in \mathcal{D}_r$ exceeds a given risk limit. We need to design an algorithm that finds a maximum matching that avoids selecting all the component classes for each $D_r \in \mathcal{D}_r$. To solve this problem, we introduce a *combination inhibitor* $Inh(D_r)$, which is a supplementary graph constructed by the algorithm CombInhibitor, shown in Figure 5.

Let us reconsider the running example, and assume that p-risk limit $m_p = 4$ is given. Then `rn&bn` is the only combination risk class in Figure 1 that should be inhibited. CombInhibitor adds a component inhibitor for `rn&bn` to the augmented matching graph. In Figure 3, the inhibitor node is labeled as `!rn&ad`. The combination inhibitor works as follows: The dashed edges attached to the inhibitor node have the highest weight in the graph. Therefore, if both *real name* and `shopping.address` are selected in a matching M, we can always make another matching M' by replacing one of the matching edges, say, the one adjacent to `real name`, with (`real name`, `!rn&ad`). Then M' should have a score higher than M. Therefore maximum matching will give us a solution that avoids simultaneously selecting `real name` and

For each combination risk class $D_r \in \mathcal{D}_r$, do:

1. Add the following bipartite subgraph $Inh(D_r) = (V_h, U_h, E_h)$ to the augmented matching graph $G^a_{\sigma,\beta}(\mathcal{C}, \mathcal{RA}, E)$. Let $C_1, \ldots, C_k$ be the component classes of D_r.

1.1 The node set V_h equals the component classes $\{C_1, \ldots, C_k\}$, and the other node set U_h equals the singleton set $\{A_h\}$ containing a newly introduced *inhibitor node* A_h.

1.2 The edge set E_h consists of k edges $(C_1, A_h), \ldots, (C_k, A_h)$, where each edge has an equal weight w_h such that w_h is any fixed value higher than the maximum similarity score found in $G_{\sigma,\beta}(\mathcal{C}, \mathcal{RA}, E)$.

Fig. 5. CombInhibitor($\mathcal{D}_r, G^a_{\sigma,\beta}$): Adding combination inhibitors

`shopping.address`. Thus we have succeeded in preventing p-risk value from exceeding 4. Formally, we have the following property:

Theorem 3. *Suppose that a matching graph $G_{\sigma,\beta} = (\mathcal{C}, \mathcal{RA}, E)$ is augmented with the combination inhibitor $Inh(D_r)$ for each $D_r \in \mathcal{D}_r$, where $\mathcal{D}_r$ is a subset of combination risk classes of $G_{\sigma,\beta}$. Then a maximum matching M of $G_{\sigma,\beta}$ always includes an edge of $Inh(D_r)$ for any $D_r \in \mathcal{D}_r$. Thus there is no maximum matching that includes all the component classes of D_r.*

Proof. (omitted due to space limitation)

By the above theorem, just adding combination inhibitor $Inh(D_r)$ to the matching graph can prevent application of the exceeded risk values of D_r. The supplementary subgraphs introduced by combination inhibitors have a maximum total size equal to the size of combination risk classes. Thus adding combination inhibitors multiplies the graph size only by a constant factor. We also note that the total maximum weight includes the weight of inhibitor edges given by w_{inh} = (the number of inhibitors)* w_h. Thus we need to subtract w_{inh} from the matching weight w_M to obtain the actual total similarity score.

5.3 Finding Optimum Matching

We use the following monotonicity in matching solutions for searching on risk values.

Lemma 1. *Let M^a_1 be a matching of graph $G_{\sigma,\beta}$ such that M^a_1 satisfies risk limits r_f and r_p. Then there is a matching M^a_2 such that M^a_2 satisfies risk limits $r'_f > r_f$ and $r'_p > r_p$, and $\sigma^a(M^a_1) \leq \sigma^a(M^a_2)$.*

Proof. It is obvious that M^a_1 remains a matching under the weaker limits of r'_f and r'_p. Thus at least M^a_1 satisfies the condition of M^a_2 of the lemma. □

Let F (resp. P) be the number of distinct f-risk (resp. p-risk) values appearing in $G^a_{\sigma,\beta}$. If we are using 5-digit risk values, then we have $F \leq 5$ and $P \leq 5$. Figure 6 shows matching algorithms for the optimization problems defined in Section 4.1.

Algorithm MaxSim($G^a_{\sigma,\beta}, m_f, m_p$) /* Similarity score maximization under risk limit */
Input Augmented bipartite graph $G^a_{\sigma,\beta}$, maximum f-risk m_f, and maximum p-risk m_p, where m_f and/or m_f may be ∞. /*
Output Maximum augmented matching M^a of $G^a_{\sigma,\beta}$ such that $r_f(M^a) \leq m_f$ and $r_p(M^a) \leq m_p$.
1. Remove from $G^a_{\sigma,\beta}$ classes C such that $r_f(C) > m_f$ or $r_p(C) > m_p$.
2. Let $\mathcal{D}_r$ be the set of combination risk classes D_r such that $r_f(D_r) > m_f$ or $r_p(D_r) > m_p$ holds. Execute CombInhibitor($\mathcal{D}_r, G^a_{\sigma,\beta}$).
3. Apply PAOMatch to $G^a_{\sigma,\beta}$ to obtain M^a.

Algorithm MaxSimCombinedRisk($G^a_{\sigma,\beta}, m_c$) /* Similarity score maximization under combined-risk limit */
Input: Augmented bipartite graph $G^a_{\sigma,\beta}$, and maximum combined-risk m_c.
Output: Maximum augmented matching M^a of $G^a_{\sigma,\beta}$ such that $cr(r_f(M^a), r_p(M^a)) \leq m_c$.
1. Assume that $F < P$. Otherwise in the following steps swap F with P, and swap m_f with m_p.
2. For each value r_f in F do:
2.1 Compute maximum r_p such that $cr(r_f, r_p) \leq m_c$ holds.
2.2 Call MaxSim($G^a_{\sigma,\beta}, r_f, r_p$).
3. Report matching M^a that had maximum score at 2.2

Algorithm MinRisk($G^a_{\sigma,\beta}, w_{min}$) /* Risk minimization under minimum similarity score w_{min}. */
1. Assume that $F < P$. Otherwise in the following steps swap F with P, and swap m_f with m_p.
2. For each value r_f in F do:
2.1 Perform binary search on p-values to find minimum r_p such that result M^a of MaxSim($G^a_{\sigma,\beta}, r_f, r_p$) has score no smaller than w_{min}.
3. Report the matching found in Step 2.1 such that its combined risk r is minimum.

Algorithm MaxCombined($G^a_{\sigma,\beta}$) /* Combined score maximization. */
1. For each f-risk value p_f and for each p-risk value p_f, do:
1.1 Call MaxSim($G^a_{\sigma,\beta}, r_f, r_p$).
2. Report matching M^a that had maximum combined score $\sigma(M^a)/r_c(M^a)$ in Step 1.1.

Fig. 6. Algorithms for maximum matching under given risk constraints

Theorem 4. *Let F (resp. P) be the number of distinct f-risk (resp. p-risk) values appearing in augmented matching graph $G^a_{\sigma,\beta}$. Let R be $F + P$, and let N be the number of nodes and E be the number of* `partOf` *links in $G^a_{\sigma,\beta}$. The following holds:*

1. MaxSim($G^a_{\sigma,\beta}, m_f, m_p$) *solves similarity score maximization in $O(N^3 + E^3)$ time.*
2. MaxSimCombinedRisk($G^a_{\sigma,\beta}, m_c$) *solves similarity score maximization under combined-risk limit m_c in $O((N^3 + E^3)R)$ time.*
3. MinRisk($G^a_{\sigma,\beta}, w_{min}$) *solves risk minimization under minimum similarity score w_{min} in $O((N^3 + E^3)R \log R)$ time.*

4. *MaxCombined*($G^a_{\sigma,\beta}$) *solves combined score maximization in* $O((N^3 + E^3)R^2)$ *time.*

Proof. 1. In Step 1 of MaxSim($G^a_{\sigma,\beta}, m_f, m_p$), classes C that violate the maximum limit m_f or m_p are removed. If these classes C are not removed, it is easy to construct a graph that has a maximum matching violating one of these limits. In Step 2, CombInhibitor introduces combination inhibitors so that by Theorem 3, any matching of $G^a_{\sigma,\beta}$ will not include a combination risk class that violates the limits. If CombInhibitor is not applied, it is easy to construct a graph that has a maximum matching that includes all the component classes of a combination risk class which violates the limits. Thus the matching obtained at Step 3 gives maximum score under the limits m_f and m_p. For the time bound, Step 1 and Step 2 can be done in linear time and increase the size of $G^a_{\sigma,\beta}$ by a factor of a constant. Thus by Theorem 2, Step 3 can be done in $O(N^3 + E^3)$ time.
2. For similarity score maximization under combined-risk limit m_c, testing maximum matching score among every combination of risk values r_f and r_p that satisfy the limit m_c guarantees that there will be no other matching that has a higher score while satisfying m_c. We do not need to test on combinations r'_f and r'_p which have combined risk values less than m_c, since by Lemma 1, matching score satisfying r'_f and r'_p does not exceed the score satisfying $r_f \geq r'_f$ and $r_p \geq r'_p$ such that $cr(r_f, r_p) \leq cr(r_f, r_p) \leq m_c$. For the time bound, MaxSim($G^a_{\sigma,\beta}, r_f, r_p$) is called $F \leq R$ times, which gives the bound $O((N^3 + E^3)R)$.
3. For risk minimization under minimum similarity score w_{min}, it is sufficient to test all the combinations of f-risk and p-risk values that have matching M^a such that $\sigma^a(M^a) \geq w_{min}$ holds. Again by Lemma 1, if a combination of r_f and r_p has a matching score greater than the minimum limit w_{min}, then all the combinations such that $r'_f \geq r_f$ and $r'_p \geq r_p$ also have matching score greater than w_{min}. This property allows us to perform binary search on r_f for each fixed r_p. Thus MaxSim($G^a_{\sigma,\beta}, r_f, r_p$) is called $O(R \log R)$ at Step 2.2 and we have the time bound.
4. For combined score maximization, again it is sufficient to test all the combinations of f-risk and p-risk values to find a matching M^a having maximum combined score $s_c(M^a) = \sigma(M^a)/r_c(M^a)$. Since the combined score $s_c = w/r$ does not have monotonicity, we try MaxSim($G^a_{\sigma,\beta}, r_f, r_p$) for $O(R^2)$ times. □

6 Conclusion

In this paper, we proposed the concept of privacy attribute ontology for identity management involving complex attributes and identities. Our ontology model realizes risk evaluation of matching attributes, and the algorithms presented in this paper solve maximum similarity matching under various types of risk constraints.

Acknowledgment

This work is in part supported by the Grant-in-Aid for Scientific Research of JSPS (Japan Society for the Promotion of Science) (#18300031), and Strategic International Cooperative Program of JST (Japan Science and Technology Agency). The work of

Gail-J. Ahn was partially supported by the grants from US National Science Foundation (NSF-IIS-0242393) and the US Department of Energy Early Career Principal Investigator Award (DE-FG02-03ER25565).

References

1. Microsoft Developer Network (MSDN) CardSpace page, http://msdn.microsoft.com/CardSpace
2. Cohen, W., Ravikumar, P., Feinberg, S.: A Comparison of String Metrics for Matching Names and Records. In: Proc. KDD Workshop on Data Cleaning and Object Consolidation (2003)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability - A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
4. Kagal, L., Finin, T.W., Joshi, A.: A Policy Based Approach to Security for the Semantic Web. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 402–418. Springer, Heidelberg (2003)
5. Jutla, D.N., Bodorik, P.: Sociotechnical Architecture for Online Privacy. IEEE Security & Privacy 3(2), 29–39 (2005)
6. Japan Network Security Association, Surveys on Information Security Incidents (in Japanese) (2006), http://www.jnsa.org/result/2006/pol/insident/070720/
7. Kolari, P., Li Ding, S., Ganjugunte, L., Kagal, A.J., Finin, T.: Enhancing Web Privacy Protection through Declarative Policies. In: Proc. IEEE Workshop on Policy for Distributed Systems and Networks(POLICY 2005) (June 2005)
8. Kuhn, H.W.: The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly 2, 83–97 (1955)
9. Li, W.-S., Clifton, C.: SEMINT: a Tool for Identifying Attribute Correspondences in Heterogeneous Database Using Neural Networks. Data Knowledge Eng. 33(1), 49–84 (2000)
10. Liberty Alliance Project Homepage, http://www.projectliberty.org/
11. OpenID Foundation, http://openid.net/
12. OWL Web Ontology Language Overview, W3C Recommendation 10 (February 2004), http://www.w3.org/TR/owl-features/
13. Patel, C., Supekar, K., Lee, Y.: OntoGenie: Extracting Ontlogy Instances from WWW. In: Proc. Huaman Language Technology for the Semantic Web and Web Services, ISWC 2003 (2003)
14. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, W3C Working Group Note (November 13, 2006)
15. Udrea, O., Getoor, L., Miller, R.J.: Leveraging Data and Structure in Ontology Integration. In: Proc. ACM SIGMOD 2007, pp. 449–460 (2007)
16. WordNet — a Lexical Database for the English Language, Princeton University, http://wordnet.princeton.edu/