A Semantic Privacy-Preserving Model for Data Sharing and Integration

Yuh-Jong Hu Jiun-Jan Yang
{hu, 98753036}@cs.nccu.edu.tw

Emerging Network Technology(ENT) Lab.
Department of Computer Science
National Chengchi University, Taipei, Taiwan

May-25-2011

International Conference on
Web Intelligence, Mining, and Semantics (WIMS'11)

Part I

Research Goals

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
  - heterogeneity of the data sources
  - relation between the global schema and the data sources
  - limitations on the mechanisms for access the data sources
  - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
  - heterogeneity of the data sources
  - relation between the global schema and the data sources
  - limitations on the mechanisms for access the data sources
  - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
  - heterogeneity of the data sources
  - relation between the global schema and the data sources
  - limitations on the mechanisms for access the data sources
  - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
    - heterogeneity of the data sources
    - relation between the global schema and the data sources
    - limitations on the mechanisms for access the data sources
    - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
  - heterogeneity of the data sources
  - relation between the global schema and the data sources
  - limitations on the mechanisms for access the data sources
  - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
    - heterogeneity of the data sources
    - relation between the global schema and the data sources
    - limitations on the mechanisms for access the data sources
    - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
  - heterogeneity of the data sources
  - relation between the global schema and the data sources
  - limitations on the mechanisms for access the data sources
  - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Motivations

- Large enterprises spend a great deal of time and money on data sharing and integration [3].
- Semantic web technologies provide a possible solution.
- But it is a very complicated research problem because [11]:
  - heterogeneity of the data sources
  - relation between the global schema and the data sources
  - limitations on the mechanisms for access the data sources
  - queries processing expressed on the global schema
- We further exploit data protection issue besides data sharing.

Research Goals and Achievements

1. Providing a virtual platform for data sharing and protection in multiple servers with relational database systems.
2. Representing and enforcing semantics-enabled policies as a combination of ontology and rule.
3. Using a combination of semantics-enabled policies for data sharing and protection in multiple servers.
4. Ensuring soundness and completeness of query rewriting services in a semantic privacy-preserving model.

Research Goals and Achievements

1. Providing a virtual platform for data sharing and protection in multiple servers with relational database systems.
2. Representing and enforcing semantics-enabled policies as a combination of ontology and rule.
3. Using a combination of semantics-enabled policies for data sharing and protection in multiple servers.
4. Ensuring soundness and completeness of query rewriting services in a semantic privacy-preserving model.

Research Goals and Achievements

1. Providing a virtual platform for data sharing and protection in multiple servers with relational database systems.
2. Representing and enforcing semantics-enabled policies as a combination of ontology and rule.
3. Using a combination of semantics-enabled policies for data sharing and protection in multiple servers.
4. Ensuring soundness and completeness of query rewriting services in a semantic privacy-preserving model.

Research Goals and Achievements

1. Providing a virtual platform for data sharing and protection in multiple servers with relational database systems.
2. Representing and enforcing semantics-enabled policies as a combination of ontology and rule.
3. Using a combination of semantics-enabled policies for data sharing and protection in multiple servers.
4. Ensuring soundness and completeness of query rewriting services in a semantic privacy-preserving model.

Related Work

Cited Papers

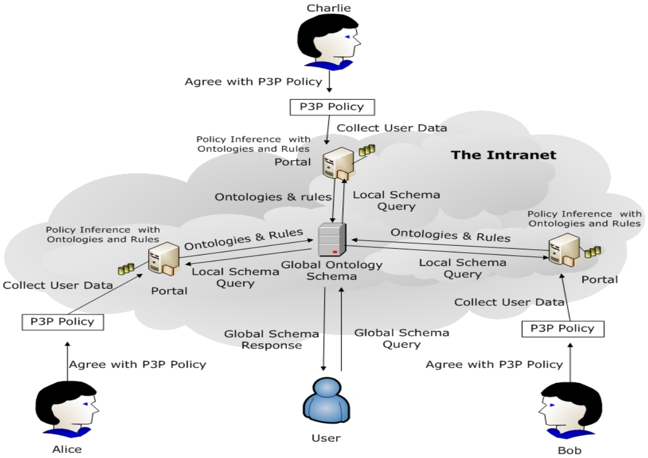
- Data integration for relational DB
  A. Halevy, et al. [18] [19] [27]
- Data integration with Description Logic (DL)
  D. Calvanses et al. [8] [9] [10]
- Privacy-preserving data integration and sharing
  C. Clifton et al. [12]
- Data usage control model
  J. Park and R. T. Sandhu [32]
- Access control policies and languages in open environments
  S. Jajodia et al. [22]
- A privacy policy model for enterprise
  G. Karjoth [24] [25]
- A KRDB perspective for the semantic web

  F. Goasdoué and M.-C. Rousset [15]

Part II

A Privacy-Preserving Model

A Semantic Privacy Protection Model



Formal Privacy Protection Policy

1. A formal policy ($\mathcal{FP}$) is a declarative expression executed in a computer system for a human legal norm without semantic ambiguity.
2. An $\mathcal{FP}$ is created from a policy language ($\mathcal{PL}$), and $\mathcal{PL}$ is shown as a combination of ontology and rule languages.
3. An $\mathcal{FP}$ is composed of ontologies $\mathcal{O}$ and rules $\mathcal{R}$, where ontologies are created from an ontology language and rules are created from a rule language.
4. A formal protection policy ($\mathcal{FPP}$) is an $\mathcal{FP}$ that aims at representing and enforcing resource protection principles, where the structure of resources is modeled as ontologies $\mathcal{O}$ but the resources protection is shown as rules $\mathcal{R}$.

Formal Privacy Protection Policy

1. A formal policy ($\mathcal{FP}$) is a declarative expression executed in a computer system for a human legal norm without semantic ambiguity.
2. An $\mathcal{FP}$ is created from a policy language ($\mathcal{PL}$), and $\mathcal{PL}$ is shown as a combination of ontology and rule languages.
3. An $\mathcal{FP}$ is composed of ontologies $\mathcal{O}$ and rules $\mathcal{R}$, where ontologies are created from an ontology language and rules are created from a rule language.
4. A formal protection policy ($\mathcal{FPP}$) is an $\mathcal{FP}$ that aims at representing and enforcing resource protection principles, where the structure of resources is modeled as ontologies $\mathcal{O}$ but the resources protection is shown as rules $\mathcal{R}$.

Formal Privacy Protection Policy

1. A formal policy ($\mathcal{FP}$) is a declarative expression executed in a computer system for a human legal norm without semantic ambiguity.
2. An $\mathcal{FP}$ is created from a policy language ($\mathcal{PL}$), and $\mathcal{PL}$ is shown as a combination of ontology and rule languages.
3. An $\mathcal{FP}$ is composed of ontologies $\mathcal{O}$ and rules $\mathcal{R}$, where ontologies are created from an ontology language and rules are created from a rule language.
4. A formal protection policy ($\mathcal{FPP}$) is an $\mathcal{FP}$ that aims at representing and enforcing resource protection principles, where the structure of resources is modeled as ontologies $\mathcal{O}$ but the resources protection is shown as rules $\mathcal{R}$.

Formal Privacy Protection Policy

1. A formal policy ($\mathcal{FP}$) is a declarative expression executed in a computer system for a human legal norm without semantic ambiguity.
2. An $\mathcal{FP}$ is created from a policy language ($\mathcal{PL}$), and $\mathcal{PL}$ is shown as a combination of ontology and rule languages.
3. An $\mathcal{FP}$ is composed of ontologies $\mathcal{O}$ and rules $\mathcal{R}$, where ontologies are created from an ontology language and rules are created from a rule language.
4. A formal protection policy ($\mathcal{FPP}$) is an $\mathcal{FP}$ that aims at representing and enforcing resource protection principles, where the structure of resources is modeled as ontologies $\mathcal{O}$ but the resources protection is shown as rules $\mathcal{R}$.

Semantic Mapping from Local to Global Schema

Definition (Semantic Mapping: GAV, LAV, and GLAV)

- Global-As-View(GAV): Each concept in the global schema is expressed in terms of query over the data sources.
- Local-As-View(LAV): Defining each concept in the data sources as a view over the global schema [10] [26].
- Global-Local-As-View(GLAV): Allowing flexible schema definitions independent of the particular details of the data sources [14] [30].

Semantic Mapping from Local to Global Schema

Definition (Semantic Mapping: GAV, LAV, and GLAV)

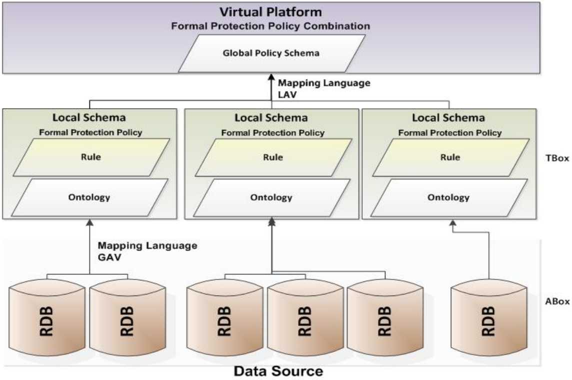
- Global-As-View(GAV): Each concept in the global schema is expressed in terms of query over the data sources.
- Local-As-View(LAV): Defining each concept in the data sources as a view over the global schema [10] [26].
- Global-Local-As-View(GLAV): Allowing flexible schema definitions independent of the particular details of the data sources [14] [30].

Semantic Mapping from Local to Global Schema

Definition (Semantic Mapping: GAV, LAV, and GLAV)

- Global-As-View(GAV): Each concept in the global schema is expressed in terms of query over the data sources.
- Local-As-View(LAV): Defining each concept in the data sources as a view over the global schema [10] [26].
- Global-Local-As-View(GLAV): Allowing flexible schema definitions independent of the particular details of the data sources [14] [30].

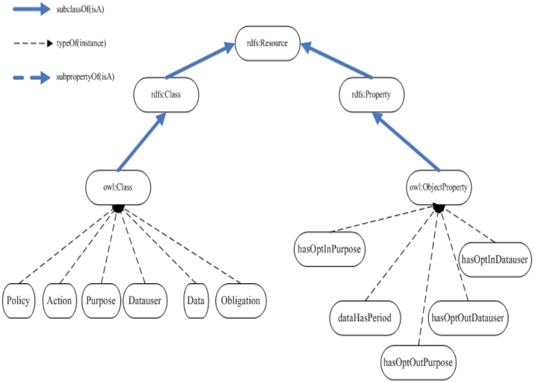
A Semantic Privacy-Preserving Model



A Partial Ontology of FIPs

PROPERTY AND CLASS IN FIPS ONTOLOGY

- $\top \sqsubseteq \forall$ hasOptInPurpose.Data, $\top \sqsubseteq \forall$ hasOptInPurpose^{-}.Purpose
- $\top \sqsubseteq \forall$ hasOptInDatauser.Data, $\top \sqsubseteq \forall$ hasOptInDatauser^{-}.Datauser

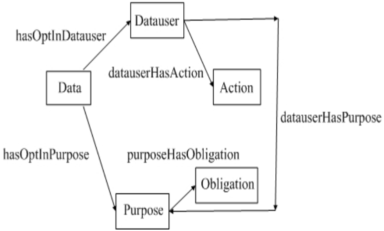


Data Request Services

SWRL Rules for Data Request

- FIP's five attributes ($?d$, $?p$, $?du$, $?a$, $?o$) for each data request service
- An initial feasible parameter input set is $\mathcal{FS} = input(?du, ?r, ?p)$, and output dataset is $output(?d, ?o)$ for pattern-matching and subject-based data requests
- $\texttt{hasOptInPurpose.Data}(?data) \wedge \texttt{hasOptInPurpose}^{-}\texttt{.Purpose}(?purpose)$
  $\longrightarrow \texttt{hasOptInPurpose}(?data, ?purpose) \longleftarrow (1)$
- $\texttt{hasOptInDatauser.Data}(?data) \wedge \texttt{hasOptInDatauser}^{-}\texttt{.Datauser}(?datauser)$

  $\longrightarrow \texttt{hasOptInDatauser}(?data, ?datauser) \longleftarrow (2)$



Formal Privacy Protection Policy (conti.)

1. A privacy protection policy shown as an $\mathcal{FPP}$ is a combination of ontologies and rules, where DL-based ontologies provide data sharing, while LP-based rules provide data query and protection.
2. A formal policy combination ($\mathcal{FPC}$) in a global policy schema ($\mathcal{GPS}$) allows data sharing as an integration of $\mathcal{FP}$ from a variety of servers.
3. a formal protection policy combination ($\mathcal{FPPC}$) allows data sharing and protection from $\mathcal{FPC}$

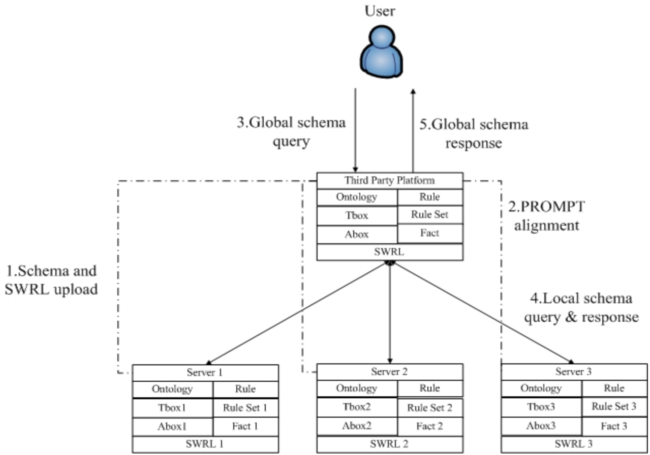
Formal Privacy Protection Policy (conti.)

1. A privacy protection policy shown as an $\mathcal{FPP}$ is a combination of ontologies and rules, where DL-based ontologies provide data sharing, while LP-based rules provide data query and protection.
2. A formal policy combination ($\mathcal{FPC}$) in a global policy schema ($\mathcal{GPS}$) allows data sharing as an integration of $\mathcal{FP}$ from a variety of servers.
3. a formal protection policy combination ($\mathcal{FPPC}$) allows data sharing and protection from $\mathcal{FPC}$

Formal Privacy Protection Policy (conti.)

1. A privacy protection policy shown as an $\mathcal{FPP}$ is a combination of ontologies and rules, where DL-based ontologies provide data sharing, while LP-based rules provide data query and protection.
2. A formal policy combination ($\mathcal{FPC}$) in a global policy schema ($\mathcal{GPS}$) allows data sharing as an integration of $\mathcal{FP}$ from a variety of servers.
3. a formal protection policy combination ($\mathcal{FPPC}$) allows data sharing and protection from $\mathcal{FPC}$

A $\mathcal{VP}$ for Ontology Merging and Rule Integration



A Perfect Ontology Alignment

Definition (A Perfect Ontology Alignment)

- A perfect ontology alignment between $\mathcal{T}_i$ in $\mathcal{O}_i$ and $\mathcal{T}_j$ in $\mathcal{O}_j$ via a mapping (uid, e_i, e_j, n, ρ) and merging satisfied the following:
  - $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ entity names are either for describing the root class or for property which corresponding to the privacy protection concepts and relations.
  - A numeric confidence measure n is always equal 1.
  - ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathcal{T}_i$ and $\mathcal{T}_j$ schemas.
- A mapping language $\mathcal{ML}$ semantically links a global policy schema $\mathcal{GPS}$ to multiple local policy schemas $\mathcal{LPS}s$.

A Perfect Ontology Alignment

Definition (A Perfect Ontology Alignment)

- A perfect ontology alignment between $\mathcal{T}_i$ in $\mathcal{O}_i$ and $\mathcal{T}_j$ in $\mathcal{O}_j$ via a mapping (uid, e_i, e_j, n, ρ) and merging satisfied the following:
  - $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ entity names are either for describing the root class or for property which corresponding to the privacy protection concepts and relations.
  - A numeric confidence measure n is always equal 1.
  - ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathcal{T}_i$ and $\mathcal{T}_j$ schemas.
- A mapping language $\mathcal{ML}$ semantically links a global policy schema $\mathcal{GPS}$ to multiple local policy schemas $\mathcal{LPS}s$.

A Perfect Ontology Alignment

Definition (A Perfect Ontology Alignment)

- A perfect ontology alignment between $\mathcal{T}_i$ in $\mathcal{O}_i$ and $\mathcal{T}_j$ in $\mathcal{O}_j$ via a mapping (uid, e_i, e_j, n, ρ) and merging satisfied the following:
  - $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ entity names are either for describing the root class or for property which corresponding to the privacy protection concepts and relations.
  - A numeric confidence measure n is always equal 1.
  - ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathcal{T}_i$ and $\mathcal{T}_j$ schemas.
- A mapping language $\mathcal{ML}$ semantically links a global policy schema $\mathcal{GPS}$ to multiple local policy schemas $\mathcal{LPS}s$.

A Perfect Ontology Alignment

Definition (A Perfect Ontology Alignment)

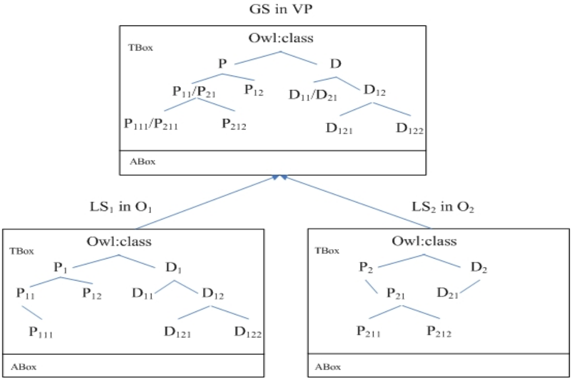
- A perfect ontology alignment between $\mathcal{T}_i$ in $\mathcal{O}_i$ and $\mathcal{T}_j$ in $\mathcal{O}_j$ via a mapping (uid, e_i, e_j, n, ρ) and merging satisfied the following:
  - $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ entity names are either for describing the root class or for property which corresponding to the privacy protection concepts and relations.
  - A numeric confidence measure n is always equal 1.
  - ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathcal{T}_i$ and $\mathcal{T}_j$ schemas.
- A mapping language $\mathcal{ML}$ semantically links a global policy schema $\mathcal{GPS}$ to multiple local policy schemas $\mathcal{LPS}s$.

A Perfect Ontology Alignment

Definition (A Perfect Ontology Alignment)

- A perfect ontology alignment between $\mathcal{T}_i$ in $\mathcal{O}_i$ and $\mathcal{T}_j$ in $\mathcal{O}_j$ via a mapping (uid, e_i, e_j, n, ρ) and merging satisfied the following:
  - $e_i \in \mathcal{T}_i$ and $e_j \in \mathcal{T}_j$ entity names are either for describing the root class or for property which corresponding to the privacy protection concepts and relations.
  - A numeric confidence measure n is always equal 1.
  - ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathcal{T}_i$ and $\mathcal{T}_j$ schemas.
- A mapping language $\mathcal{ML}$ semantically links a global policy schema $\mathcal{GPS}$ to multiple local policy schemas $\mathcal{LPS}s$.

Class Alignment for Ontology Merging



A Perfect Rule Integration

Definition (A Perfect Rule Integration)

- A datalog rule is a $\mathcal{CQ}$ of the form: $v_i \leftarrow conj_i(\overrightarrow{x}_i)$ [9].
- A datalog rule $\mathtt{r}_i$ in the $\mathcal{R}_i$ of $\mathcal{FP}_i$ is: $\mathcal{H} \longleftarrow \mathcal{B}_1 \wedge \mathcal{B}_2 \wedge, \cdots, \wedge \mathcal{B}_n$, where $\mathcal{H}$, the query results (or views).
- A perfect datalog rules integration is: $\exists \mathtt{r}_i \in \mathcal{RS}_i$ in $\mathcal{FP}_i$, for data sharing and protection without causing rules conflict with $\exists \mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i$, $\lambda_i \in \underset{i}{\diamond}\mathcal{O}_i$.
- Avoid conditions as: (*Incomplete*)$\exists \mathtt{r}_i \models \lambda_i \Rightarrow \exists \mathtt{r}'_i \not\models \lambda_i$ and (*Unsound*)$\exists \mathtt{r}_i \not\models \lambda_i \Rightarrow \exists \mathtt{r}'_i \models \lambda_i$.

A Perfect Rule Integration

Definition (A Perfect Rule Integration)

- A datalog rule is a $\mathcal{CQ}$ of the form: $v_i \leftarrow conj_i(\overrightarrow{x}_i)$ [9].
- A datalog rule $\mathtt{r}_i$ in the $\mathcal{R}_i$ of $\mathcal{FP}_i$ is: $\mathcal{H} \longleftarrow \mathcal{B}_1 \wedge \mathcal{B}_2 \wedge, \cdots, \wedge \mathcal{B}_n$, where $\mathcal{H}$, the query results (or views).
- A perfect datalog rules integration is: $\exists \mathtt{r}_i \in \mathcal{RS}_i$ in $\mathcal{FP}_i$, for data sharing and protection without causing rules conflict with $\exists \mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i$, $\lambda_i \in \underset{i}{\diamond}\mathcal{O}_i$.
- Avoid conditions as: (*Incomplete*)$\exists \mathtt{r}_i \models \lambda_i \Rightarrow \exists \mathtt{r}'_i \not\models \lambda_i$ and (*Unsound*)$\exists \mathtt{r}_i \not\models \lambda_i \Rightarrow \exists \mathtt{r}'_i \models \lambda_i$.

A Perfect Rule Integration

Definition (A Perfect Rule Integration)

- A datalog rule is a $\mathcal{CQ}$ of the form: $v_i \leftarrow conj_i(\overrightarrow{x}_i)$ [9].
- A datalog rule $\mathtt{r}_i$ in the $\mathcal{R}_i$ of $\mathcal{FP}_i$ is: $\mathcal{H} \longleftarrow \mathcal{B}_1 \wedge \mathcal{B}_2 \wedge, \cdots, \wedge \mathcal{B}_n$, where $\mathcal{H}$, the query results (or views).
- A perfect datalog rules integration is: $\exists \mathtt{r}_i \in \mathcal{RS}_i$ in $\mathcal{FP}_i$, for data sharing and protection without causing rules conflict with $\exists \mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i$, $\lambda_i \in \underset{i}{\diamond}\mathcal{O}_i$.
- Avoid conditions as: (*Incomplete*)$\exists \mathtt{r}_i \models \lambda_i \Rightarrow \exists \mathtt{r}'_i \not\models \lambda_i$ and (*Unsound*)$\exists \mathtt{r}_i \not\models \lambda_i \Rightarrow \exists \mathtt{r}'_i \models \lambda_i$.

A Perfect Rule Integration

Definition (A Perfect Rule Integration)

- A datalog rule is a $\mathcal{CQ}$ of the form: $v_i \leftarrow conj_i(\overrightarrow{x}_i)$ [9].
- A datalog rule $\mathtt{r}_i$ in the $\mathcal{R}_i$ of $\mathcal{FP}_i$ is: $\mathcal{H} \longleftarrow \mathcal{B}_1 \wedge \mathcal{B}_2 \wedge, \cdots, \wedge \mathcal{B}_n$, where $\mathcal{H}$, the query results (or views).
- A perfect datalog rules integration is: $\exists \mathtt{r}_i \in \mathcal{RS}_i$ in $\mathcal{FP}_i$, for data sharing and protection without causing rules conflict with $\exists \mathtt{r}'_i \in \underset{i}{\odot} \mathcal{R}_i$, $\lambda_i \in \underset{i}{\diamond} \mathcal{O}_i$.
- Avoid conditions as: (*Incomplete*)$\exists \mathtt{r}_i \models \lambda_i \Rightarrow \exists \mathtt{r}'_i \not\models \lambda_i$ and (*Unsound*)$\exists \mathtt{r}_i \not\models \lambda_i \Rightarrow \exists \mathtt{r}'_i \models \lambda_i$.

Part III

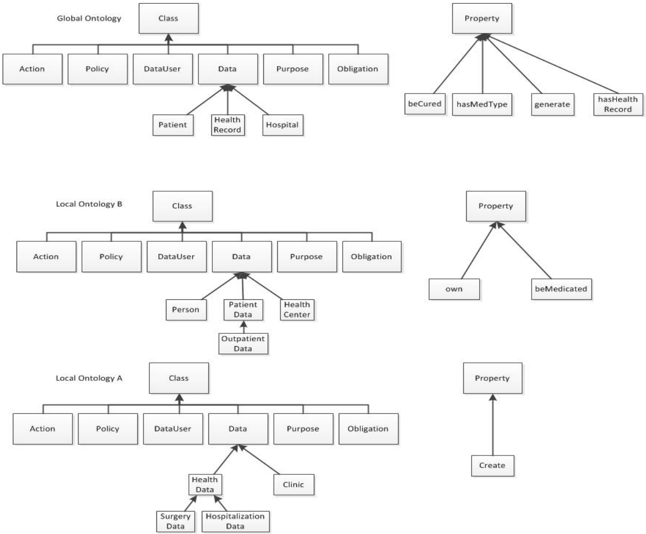
An EHRs Sharing and Protection Scenario

An EHRs Sharing and Protection Scenario

Example (Scenario Description)

Under the data protection law, two hospitals, `A` and `B`, have allowed to share their patients' Electronic Health Records (EHRs) after patients give their consents for the medication purpose. A patient was hospitalized in the hospital `A` for a surgery. After that, this patient went to the hospital `B` for an outpatient medication. A physician in the hospital `B` was authorized to query this patient's sharable EHR at the $\mathcal{VP}$ collected from hospital `A` and hospital `B`'s RDB data sources.

An EHRs Sharing and Protection Scenario(conti.)



An EHRs Sharing and Protection Scenario(conti.)

The hospital A's local ontology schema

- Class: Clinic and HealthData with subClass SurgeryData and HospitalizationData
- Property: create with domain Hospital and range HealthData
- $T \sqsubseteq \forall$ create.Hospital
- $T \sqsubseteq \forall$ create^{-}.HealthData

The hospital B's local ontology schema

- Class: Person, HealthCenter, and PatientData with subClass OutPatientData
- Property: own, beMedicared with domain and range:
- $T \sqsubseteq \forall$ own.Person, $T \sqsubseteq \forall$ own^{-}.PatientData.
- $T \sqsubseteq \forall$ beMedicated.Person,
- $T \sqsubseteq \forall$ beMedicated^{-}.HealthCenter.

An EHRs Sharing and Protection Scenario(conti.)

The hospital A's local ontology schema

- Class: Clinic and HealthData with subClass SurgeryData and HospitalizationData
- Property: create with domain Hospital and range HealthData
- $T \sqsubseteq \forall$ create.Hospital
- $T \sqsubseteq \forall$ create$^-$.HealthData

The hospital B's local ontology schema

- Class: Person, HealthCenter, and PatientData with subClass OutPatientData
- Property: own, beMedicared with domain and range:
- $T \sqsubseteq \forall$ own.Person, $T \sqsubseteq \forall$ own$^-$.PatientData.
- $T \sqsubseteq \forall$ beMedicated.Person,
- $T \sqsubseteq \forall$ beMedicated$^-$.HealthCenter.

An EHRs Sharing and Protection Scenario(conti.)

The $\mathcal{VP}$'s global ontology schema:

- Class: Patient, Hospital, Surgery, and HealthRecord
- Property: beCured, hasHealthRecord, generate:
- $\top \sqsubseteq \forall$ beCured.Patient, $\top \sqsubseteq \forall$ beCured$^-$.Hospital
- $\top \sqsubseteq \forall$ hasHealthRecord.Patient, $\top \sqsubseteq \forall$ hasHealthRecord$^-$.HealthRecord
- $\top \sqsubseteq \forall$ generate.Hospital, $\top \sqsubseteq \forall$ generate$^-$.HealthRecord

An EHRs Sharing and Protection Scenario(conti.)

Views at the $\mathcal{VP}$ from hospital A's local schema:

- def($V1_{\texttt{Clinic}}$) = Hospital
- def($V2_{\texttt{HealthData}}$) = HealthRecord
- def($V3_{\texttt{SuregeryData}}$)=HealthRecord ∧ ∀hasMedType.Surgery
- def($V4_{\texttt{HospitalizationData}}$)=HealthRecord ∧ ∀ hasMedType.Hospitalization
- def($V5_{\texttt{create}}$) = generate

Views at the $\mathcal{VP}$ from hospital B's local schema:

- def($V6_{\texttt{Person}}$) = Patient
- def($V7_{\texttt{HealthCenter}}$) = Hospital
- def($V8_{\texttt{PatientData}}$) = HealthRecord
- def($V9_{\texttt{OutPatientData}}$)=HealthRecord ∧ ∀ hasMedType.OutPatient
- def($V10_{\texttt{beMedicated}}$) = beCured
- def($V11_{\texttt{own}}$) = hasHealthRecrod

An EHRs Sharing and Protection Scenario(conti.)

Views at the $\mathcal{VP}$ from hospital A's local schema:

- $\mathtt{def}(\mathtt{V1}_{\mathtt{Clinic}}) = \mathtt{Hospital}$
- $\mathtt{def}(\mathtt{V2}_{\mathtt{HealthData}}) = \mathtt{HealthRecord}$
- $\mathtt{def}(\mathtt{V3}_{\mathtt{SuregeryData}})=\mathtt{HealthRecord} \wedge \forall \mathtt{hasMedType.Surgery}$
- $\mathtt{def}(\mathtt{V4}_{\mathtt{HospitalizationData}})=\mathtt{HealthRecord} \wedge \forall\ \mathtt{hasMedType.Hospitalization}$
- $\mathtt{def}(\mathtt{V5}_{\mathtt{create}}) = \mathtt{generate}$

Views at the $\mathcal{VP}$ from hospital B's local schema:

- $\mathtt{def}(\mathtt{V6}_{\mathtt{Person}}) = \mathtt{Patient}$
- $\mathtt{def}(\mathtt{V7}_{\mathtt{HealthCenter}}) = \mathtt{Hospital}$
- $\mathtt{def}(\mathtt{V8}_{\mathtt{PatientData}}) = \mathtt{HealthRecord}$
- $\mathtt{def}(\mathtt{V9}_{\mathtt{OutPatientData}})=\mathtt{HealthRecord} \wedge \forall\ \mathtt{hasMedType.OutPatient}$
- $\mathtt{def}(\mathtt{V10}_{\mathtt{beMedicated}}) = \mathtt{beCured}$
- $\mathtt{def}(\mathtt{V11}_{\mathtt{own}}) = \mathtt{hasHealthRecrod}$

An EHRs Sharing and Protection Scenario(conti.)

A datalog query q at the $\mathcal{VP}$:

$\texttt{Patient}(?\texttt{x}) \wedge \texttt{beCured}(?\texttt{x}, ?\texttt{y}) \wedge \texttt{hasHealthRecrod}(?\texttt{x}, ?\texttt{r})$
$\wedge\ \texttt{HealthRecord}(?\texttt{r}) \wedge \texttt{hasMedType}(?\texttt{r}, \texttt{Surgery}) \wedge \texttt{generate}(?\texttt{y}, ?\texttt{r})$
$\longrightarrow \texttt{sqwrl} : \texttt{select}(?\texttt{x}, ?\texttt{r})$

An EHRs Sharing and Protection Scenario(conti.)

q_{va} uses views defined at the $\mathcal{VP}$

$\texttt{V6}_{\texttt{Person}} \wedge \texttt{V10}_{\texttt{beMedicated}} \wedge \texttt{V11}_{\texttt{own}} \wedge \texttt{V9}_{\texttt{OutPatientData}} \wedge \texttt{V5}_{\texttt{create}}$
$\longrightarrow \texttt{sqwrl} : \texttt{select}(?\texttt{x}, ?\texttt{r}) \longleftarrow (\texttt{q}_{\texttt{va}})$

q_{va} is rewritten as a query:

$\texttt{B} : \texttt{Person}(?\texttt{p}) \wedge \texttt{B} : \texttt{beMedicated}(?\texttt{p}, ?\texttt{c}) \wedge \texttt{B} : \texttt{own}(?\texttt{p}, ?\texttt{d})$
$\wedge\ \texttt{B} : \texttt{OutPatientData}(?\texttt{od}) \wedge \texttt{A} : \texttt{create}(?\texttt{h}, ?\texttt{hd})$
$\longrightarrow \texttt{sqwrl} : \texttt{select}(?\texttt{p}, ?\texttt{od})$

An EHRs Sharing and Protection Scenario(conti.)

q_{va} uses views defined at the $\mathcal{VP}$

$\mathtt{V6}_{\mathtt{Person}} \wedge \mathtt{V10}_{\mathtt{beMedicated}} \wedge \mathtt{V11}_{\mathtt{own}} \wedge \mathtt{V9}_{\mathtt{OutPatientData}} \wedge \mathtt{V5}_{\mathtt{create}}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?\mathtt{x}, ?\mathtt{r}) \longleftarrow (\mathtt{q}_{\mathtt{va}})$

q_{va} is rewritten as a query:

$\mathtt{B} : \mathtt{Person}(?\mathtt{p}) \wedge \mathtt{B} : \mathtt{beMedicated}(?\mathtt{p}, ?\mathtt{c}) \wedge \mathtt{B} : \mathtt{own}(?\mathtt{p}, ?\mathtt{d})$
$\wedge\ \mathtt{B} : \mathtt{OutPatientData}(?\mathtt{od}) \wedge \mathtt{A} : \mathtt{create}(?\mathtt{h}, ?\mathtt{hd})$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?\mathtt{p}, ?\mathtt{od})$

An EHRs Sharing and Protection Scenario(conti.)

q_{vb} uses views defined at the $\mathcal{VP}$

$\mathtt{V6}_{\mathtt{Person}} \wedge \mathtt{V10}_{\mathtt{beMedicated}} \wedge \mathtt{V11}_{\mathtt{own}} \wedge \mathtt{V3}_{\mathtt{SuregeryData}} \wedge \mathtt{V5}_{\mathtt{create}}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?x, ?r) \longleftarrow (q_{vb})$

q_{vb} is rewritten as a query:

$\mathtt{B} : \mathtt{Person}(?p) \wedge \mathtt{B} : \mathtt{beMedicated}(?p, ?c) \wedge \mathtt{B} : \mathtt{own}(?p, ?d)$
$\wedge\ \mathtt{A} : \mathtt{SuregeryData}(?sd) \wedge \mathtt{A} : \mathtt{create}(?h, ?hd) \longrightarrow \mathtt{sqwrl} : \mathtt{select}(?p, ?sd)$

An EHRs Sharing and Protection Scenario(conti.)

q_{vb} uses views defined at the $\mathcal{VP}$

$\mathtt{V6}_{\mathtt{Person}} \wedge \mathtt{V10}_{\mathtt{beMedicated}} \wedge \mathtt{V11}_{\mathtt{own}} \wedge \mathtt{V3}_{\mathtt{SuregeryData}} \wedge \mathtt{V5}_{\mathtt{create}}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{select}(?x, ?r) \longleftarrow (q_{vb})$

q_{vb} is rewritten as a query:

$\mathtt{B} : \mathtt{Person}(?p) \wedge \mathtt{B} : \mathtt{beMedicated}(?p, ?c) \wedge \mathtt{B} : \mathtt{own}(?p, ?d)$

$\wedge\ \mathtt{A} : \mathtt{SuregeryData}(?sd) \wedge \mathtt{A} : \mathtt{create}(?h, ?hd) \longrightarrow \mathtt{sqwrl} : \mathtt{select}(?p, ?sd)$

Part IV

Soundness and Completeness

Soundness of Query Rewriting

Theorem (Soundness of Query Rewriting)

After a perfect ontology alignment and a perfect rule integration with $\mathcal{FPPC}$, $\exists \mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$, Under a particular feasible parameter input set $\mathcal{FS}_i$, if $\lambda_j \in \mathcal{O}_i$ is protected by a $\mathcal{FPP}_i$ at each server$_i$, $\forall i$, i.e., $\forall i, r_i \in \mathcal{R}_i \nvDash \lambda_j$, then $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i \nvDash \lambda_j$ for the same $\mathcal{FS}_i$, where λ_j is a protective data set in $\mathcal{O}_i$.

Proof.

(Sketch) If $\mathtt{q(x)}$ is a query over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q_{vi}(x)}$ is a query over $\mathcal{O}_i$ in a *server$_i$*, then we need to prove the statement $\forall x \quad q(x) \longrightarrow \bigsqcup_i q_{vi}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathcal{R}_i \nvDash \lambda_j$, then $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i \nvDash \lambda_j$. The $\mathcal{CQ}$ $\mathtt{q(x)}$ is a query containment of datalog rule $\mathtt{r}'_i$ and the $\mathcal{CQ}$ $\mathtt{q_{vi}(x)}$ is a query containment of datalog rule $r_i \in \mathcal{R}_i$. The statement $\forall x \quad q(x) \longrightarrow \bigsqcup_i q_{vi}(x)$ is true because the local as view (LAV) schema mapping only allow the protected concept λ_j in each *server$_i$* to be connected to the global schema. After using a perfect ontology alignment and a perfect rule integration with a perfect mapping language $\mathcal{ML}$, we avoid the condition: $\exists \mathtt{r}_i \nvDash \lambda_j \Rightarrow \exists \mathtt{r}'_i \models \lambda_j$. $\square$

Soundness of Query Rewriting

Theorem (Soundness of Query Rewriting)

After a perfect ontology alignment and a perfect rule integration with $\mathcal{FPPC}$, $\exists \mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$, Under a particular feasible parameter input set $\mathcal{FS}_i$, if $\lambda_j \in \mathcal{O}_i$ is protected by a $\mathcal{FPP}_i$ at each server$_i, \forall i$, i.e., $\forall i, r_i \in \mathcal{R}_i \nvDash \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i \nvDash \lambda_j$ for the same $\mathcal{FS}_i$, where λ_j is a protective data set in $\mathcal{O}_i$.

Proof.

(Sketch) If $\mathtt{q(x)}$ is a query over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q_{vi}(x)}$ is a query over $\mathcal{O}_i$ in a *server$_i$*, then we need to prove the statement $\forall x \quad q(x) \longrightarrow \bigsqcup_i q_{vi}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathcal{R}_i \nvDash \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i \nvDash \lambda_j$. The $\mathcal{CQ}$ $\mathtt{q(x)}$ is a query containment of datalog rule $\mathtt{r'}_i$ and the $\mathcal{CQ}$ $\mathtt{q_{vi}(x)}$ is a query containment of datalog rule $r_i \in \mathcal{R}_i$. The statement $\forall x \quad q(x) \longrightarrow \bigsqcup_i q_{vi}(x)$ is true because the local as view (LAV) schema mapping only allow the protected concept λ_j in each *server$_i$* to be connected to the global schema. After using a perfect ontology alignment and a perfect rule integration with a perfect mapping language $\mathcal{ML}$, we avoid the condition: $\exists \mathtt{r}_i \nvDash \lambda_j \Rightarrow \exists \mathtt{r'}_i \models \lambda_j$. □

Completeness of Query Rewriting

Theorem (Completeness of Query Rewriting)

After a perfect ontology alignment and a perfect rule integration with $\mathcal{FPPC}$, $\exists \mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$, Under a particular feasible parameter input set $\mathcal{FS}_i$, if $\lambda_j \in \mathcal{O}_i$ is shareable by a $\mathcal{FPP}_i$ at each server$_i, \forall i$, i.e., $\forall i, r_i \in \mathcal{R}_i \models \lambda_j$, then $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i \models \lambda_j$ for the same $\mathcal{FS}_i$, where λ_j is a shareable data set in $\mathcal{O}_i$.

Proof.

(Sketch) If q(x) is a query over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q}_{\mathtt{vi}}(\mathtt{x})$ is a query over $\mathcal{O}_i$ in a *server$_i$*, then we need to prove the statement $\forall x \quad q(x) \longleftarrow \bigsqcup_i q_{vi}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathcal{R}_i \models \lambda_j$, then $\mathtt{r}'_i \in \underset{i}{\odot}\mathcal{R}_i \models \lambda_j$. The $\mathcal{CQ}$ q(x) is a query containment of datalog rule $\mathtt{r}'_i$ and the $\mathcal{CQ}$ $\mathtt{q}_{\mathtt{vi}}(\mathtt{x})$ is a query containment of datalog rule $r_i \in \mathcal{R}_i$. The statement $\forall x \quad q(x) \longleftarrow \bigsqcup_i q_{vi}(x)$ is true because the local as view (LAV) schema mapping only allows all of the shareable concepts λ_j in each *server$_i$* to be exported to the global schema. After using a perfect ontology alignment method and a perfect rule integration method with a perfect mapping language $\mathcal{ML}$, we avoid the condition: $\exists \mathtt{r}_i \models \lambda_j \Rightarrow \exists \mathtt{r}'_i \not\models \lambda_j$. $\square$

Completeness of Query Rewriting

Theorem (Completeness of Query Rewriting)

After a perfect ontology alignment and a perfect rule integration with $\mathcal{FPPC}$, $\exists \mathcal{GPS} = (\underset{i}{\diamond}\mathcal{O}_i, \underset{i}{\odot}\mathcal{R}_i)$ at the $\mathcal{VP}$, Under a particular feasible parameter input set $\mathcal{FS}_i$, if $\lambda_j \in \mathcal{O}_i$ is shareable by a $\mathcal{FPP}_i$ at each server$_i$, $\forall i$, i.e., $\forall i, r_i \in \mathcal{R}_i \models \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i \models \lambda_j$ for the same $\mathcal{FS}_i$, where λ_j is a shareable data set in $\mathcal{O}_i$.

Proof.

(Sketch) If $\mathtt{q(x)}$ is a query over $\underset{i}{\diamond}\mathcal{O}_i$ at the $\mathcal{VP}$ and $\mathtt{q_{vi}(x)}$ is a query over $\mathcal{O}_i$ in a *server$_i$*, then we need to prove the statement $\forall x \quad q(x) \longleftarrow \bigsqcup_i q_{vi}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathcal{R}_i \models \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathcal{R}_i \models \lambda_j$. The $\mathcal{CQ}$ $\mathtt{q(x)}$ is a query containment of datalog rule $\mathtt{r'}_i$ and the $\mathcal{CQ}$ $\mathtt{q_{vi}(x)}$ is a query containment of datalog rule $r_i \in \mathcal{R}_i$. The statement $\forall x \quad q(x) \longleftarrow \bigsqcup_i q_{vi}(x)$ is true because the local as view (LAV) schema mapping only allows all of the shareable concepts λ_j in each *server$_i$* to be exported to the global schema. After using a perfect ontology alignment method and a perfect rule integration method with a perfect mapping language $\mathcal{ML}$, we avoid the condition: $\exists \mathtt{r}_i \models \lambda_j \Rightarrow \exists \mathtt{r'}_i \not\models \lambda_j$. $\square$

Part V

Conclusion and Future Work

Conclusion

Conclusion

1. A semantic privacy-preserving model provides authorized view-based query over a widespread of autonomous multiple servers.
2. Semantics-enabled privacy protection policies empower the data sharing and access control at the virtual platform.
3. The policy combination is shown as ontology mapping/merging and rule integration.
   - The ontology mapping and merging algorithm creates a global ontology schema at the virtual platform by integrating multiple local ontology schemas for data sharing.
   - The perfect datalog rule integration enforces the data query and protection services.
4. The soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination.

Conclusion

Conclusion

1. A semantic privacy-preserving model provides authorized view-based query over a widespread of autonomous multiple servers.
2. Semantics-enabled privacy protection policies empower the data sharing and access control at the virtual platform.
3. The policy combination is shown as ontology mapping/merging and rule integration.
   - The ontology mapping and merging algorithm creates a global ontology schema at the virtual platform by integrating multiple local ontology schemas for data sharing.
   - The perfect datalog rule integration enforces the data query and protection services.
4. The soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination.

Conclusion

Conclusion

1. A semantic privacy-preserving model provides authorized view-based query over a widespread of autonomous multiple servers.
2. Semantics-enabled privacy protection policies empower the data sharing and access control at the virtual platform.
3. The policy combination is shown as ontology mapping/merging and rule integration.
   - The ontology mapping and merging algorithm creates a global ontology schema at the virtual platform by integrating multiple local ontology schemas for data sharing.
   - The perfect datalog rule integration enforces the data query and protection services.
4. The soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination.

Conclusion

Conclusion

1. A semantic privacy-preserving model provides authorized view-based query over a widespread of autonomous multiple servers.
2. Semantics-enabled privacy protection policies empower the data sharing and access control at the virtual platform.
3. The policy combination is shown as ontology mapping/merging and rule integration.
   - The ontology mapping and merging algorithm creates a global ontology schema at the virtual platform by integrating multiple local ontology schemas for data sharing.
   - The perfect datalog rule integration enforces the data query and protection services.
4. The soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination.

Conclusion

Conclusion

1. A semantic privacy-preserving model provides authorized view-based query over a widespread of autonomous multiple servers.
2. Semantics-enabled privacy protection policies empower the data sharing and access control at the virtual platform.
3. The policy combination is shown as ontology mapping/merging and rule integration.
   - The ontology mapping and merging algorithm creates a global ontology schema at the virtual platform by integrating multiple local ontology schemas for data sharing.
   - The perfect datalog rule integration enforces the data query and protection services.
4. The soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination.

Conclusion

Conclusion

1. A semantic privacy-preserving model provides authorized view-based query over a widespread of autonomous multiple servers.
2. Semantics-enabled privacy protection policies empower the data sharing and access control at the virtual platform.
3. The policy combination is shown as ontology mapping/merging and rule integration.
   - The ontology mapping and merging algorithm creates a global ontology schema at the virtual platform by integrating multiple local ontology schemas for data sharing.
   - The perfect datalog rule integration enforces the data query and protection services.
4. The soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination.

Future Work

Future Work

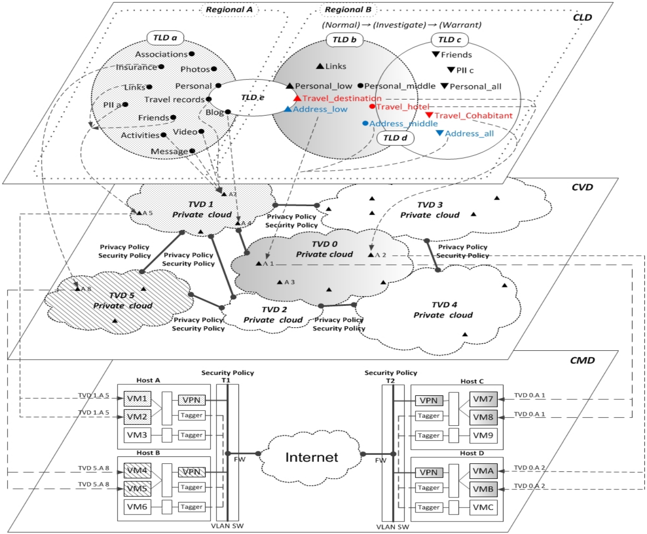
- Modularize and reuse of ontologies for data sharing and protection
- Semantics-enabled policies and framework to enforce information sharing and protection in the cloud: national security vs. privacy protection

Future Work

Future Work

- Modularize and reuse of ontologies for data sharing and protection
- Semantics-enabled policies and framework to enforce information sharing and protection in the cloud: national security vs. privacy protection

A Semantics-enabled Policy Framework in the Cloud



System Demo and Q&A

System Demo. and Q&A

- System Demo.: Jiun-Jan Yang
- Q&A

System Demo and Q&A

System Demo. and Q&A

- System Demo.: Jiun-Jan Yang
- Q&A

A. H. Anderson.
A comparison of two privacy policy languages: EPAL and XACML.
In *Proceedings of the 3rd ACM Workshop on Secure Web Services (SWS'06)*, pages 53–60. ACM, 2006.

I. A. Antón et al.
A roadmap for comprehensive online for privacy policy management.
Comm. of the ACM, 50(7):109–116, July 2007.

A. P. Bernstein and L. M. Haas.
Information integration in the enterprise.
Comm. of the ACM, 51(8):72–79, July 2008.

A. P. Bonatti et al.
An algebra for composing access control policies.
ACM Trans. on Information and Systems Security, 5(1):1–35, February 2002.

P. Bonatti and D. Olmedilla.
Policy language specification, enforcement, and integration. project deliverable D2, working group I2.
Technical report, REWERSE, 2005.

J. d. Bruijn.
RIF RDF and OWL compatibility.
Technical report, W3C, Oct. 2009.

D. Calvanese et al.
Description logic framework for information integration.
In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 2–13. Morgan Kaufmann, 1998.

D. Calvanese et al.
Data integration through $DL - Lite_A$ ontologies.
In *3rd Int. Workshop on Semantics in Data and Knowledge Base (SDKB)*, volume 4925, pages 26–47. Springer, 2008.

D. Calvanese et al.
View-based query answering over description logic ontologies.
In *Proc. of KR-2008*. AAAI Press, 2008.

D. Calvanese and G. D. Giacomo.
Data integration: A logic-based perspective.
AI Magazine, 26(1):59–70, 2005.

D. Calvanses et al.
Description logics for information integration.
In *Computational Logic*, LNAI 2408, pages 41–60. Springer, 2002.

C. Clifton et al.
Privacy-preserving data integration and sharing.
In *Data Mining and Knowledge Discovery*, pages 19–26. ACM, 2004.

J. Euzenat and P. Shvaiko.
Ontology Matching.
Springer-Verlag, 2007.

M. Friedman et al.
Navigational plans for data integration.
In *Proc. of the Sixteen National Conference on Artificial Intelligence (AAAI'99)*, pages 67–73. AAAI/MIT Press, 1999.

F. Goasdoué and M.-C. Rousset.
Answering queries using views: a KRDB perspective for the semantic web.
ACM Trans. on Internet Technology, 4(3):255–288, August 2004.

C. B. Grau et al.
Modular reuse of ontologies: Theory and practice.
Journal of Artificial Intelligence Research, pages 273–318, 2008.

C. B. Grau et al.
OWL2: The next step for OWL.
Web Semantics: Science, Services and Agents on the World Wide Web 3, pages 309–322, 2008.

A. Halevy, A. Rajaraman, and J. Ordille.
Data integration: The teenage years.
In *VLDB'06*, pages 9–16. ACM, 2006.

Y. A. Halevy.
Answering queries using views: A survey.
The VLDB Journal, 10(4):270–294, 2001.

I. Horrocks et al.
OWL rules: A proposal and prototype implementation.
Web Semantics: Science, Services and Agents on the World Wide Web 3, 3(1):23–40, 2005.

Y. J. Hu and H. Boley.
SemPif: A semantic meta-policy interchange format for multiple web policies.
In *2010 IEEE/WIC/ACM Int. Conference on Web Intelligence and Intelligent Agent Technology*, pages 302–307. IEEE, 2010.

S. Jajodia et al.
Flexible support for multiple access control policies.
ACM Trans. on Database Systems, 26(2):214–260, June 2001.

E. Jiménez-Ruiz et al.
Ontology integration using mappings: Towards getting the right logical consequences.
In *ESWC 2009*, LNCS 5554, pages 173–187. Springer, 2009.

G. Karjoth and M. Schunter.
A privacy policy model for enterprises.
In *15th IEEE Computer Security Foundations Workshop (CSFW)*. IEEE, June 2002.

G. Karjoth, M. Schunter, and E. V. Herreweghen.
Translating privacy practices into privacy promises - how to promise what you can keep.
In *POLICY'03*. IEEE, 2003.

M. Lenzerini.
Data integration: A theoretical perspective.
In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pages 233–246. ACM, 2002.

Y. A. Levy.
Logic-based techniques in data integration.
In T. Yu and S. Jajodia, editors, *Logic-based Artificial Intelligence*, pages 1–27. Kulwer, 2001.

P. Mazzoleni et al.
XACML policy integration algorithms.
ACM Trans. on Information and System Security, 11(1), 2008.

B. Motik, U. Sattler, and R. Studer.
Query answering for OWL-DL with rules.
In *3rd International Semantic Web Conference (ISWC) 2004*, LNCS 3298, pages 549–563. Springer, 2004.

A. Nash and A. Deutsch.
Privacy in GLAV information integration.
In *ICDT 2007*, LNCS 4353, pages 89–103. Springer, 2007.

J. M. O'Connor and A. K. Das.
SQWRL: a query language for OWL.
In *OWLED*, volume 529. CEUR, 2009.

J. Park and R. T. Sandhu.
The UCON$_{ABC}$ usage control model.
ACM Trans. on Information and System Security, 7(1):128–174, 2004.

A. Poggi et al.
Linking data to ontologies.
Journal on Data Semantics X, 4900:133–173, 2008.

D. J. Ullman.
Information integration using logical views.
Theoretical Computer Science, 239:189–210, 2000.

S. D. C. d. Vimercati et al.
Access control policies and languages in open environments.
In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*, pages 21–58. Springer, 2007.