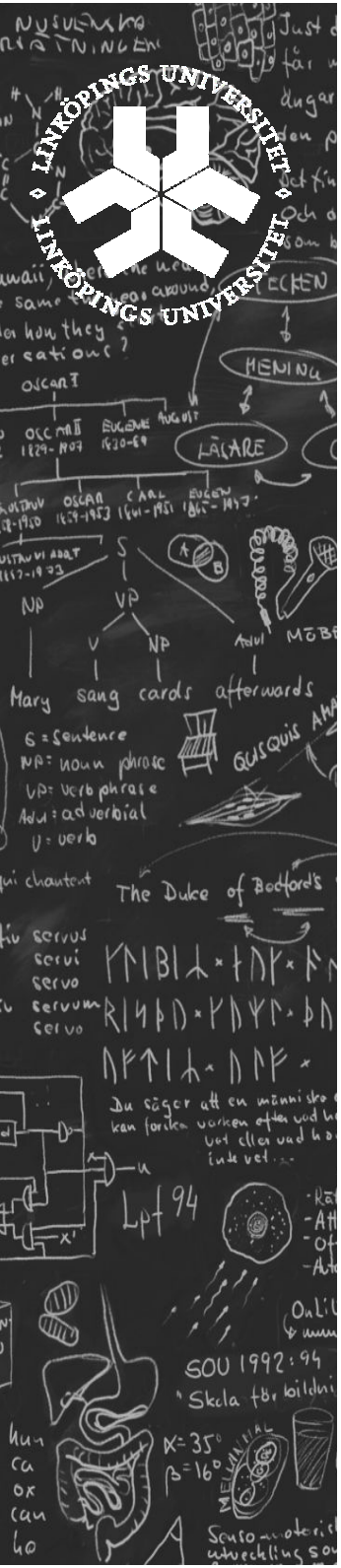

Privacy in the Semantic Web: What Policy Languages Have to Offer

Claudiu Duma, Almut Herzog, Nahid Shahmehri

Department of Computer and Information Science
Linköping University
Sweden

LiU expanding reality

Motivation

- Users are required to disclose credentials
  - E.g. A service provider grants user Alice access to a resource if Alice has a valid credit card
- Users need to control the exposure of their sensitive information
- Policies could provide the solution
  - E.g. Alice's policy: *Disclose the credit card only to BBB certified services*.
- How good are existing policy languages in expressing users' privacy requirements?

In this paper

- A privacy-centric, independent comparison of six current and prominent policy languages
  - **Protune** – a rule-based policy language
  - **Rei** – an ontology-based policy language
  - **APPEL** – a policy language for interacting with P3P privacy policies
  - **Ponder** – an object-oriented policy language
  - **Trust-X** – an XML-based language for trust negotiation
  - **KeyNote** – an authorization language for distributed systems

Overview

- Evaluation
  - Criteria
  - Method
  - Results
  - Summary
- Conclusions
- Future work

Evaluation Criteria

- "*Privacy is the claim of individuals, groups and institutions to determine for themselves, <u>when</u>, <u>how</u> and to <u>what extent</u> information about them is communicated to others*"

  by Alan Westin, Privacy and Freedom

- Criteria derived from user needs to:
  - Classify and label sensitive objects
    - *Type of classification*
    - *Granularity of objects*
  - Control to whom and to which extent the sensitive objects are disclosed
    - *Access control*
    - *Minimal information disclosure*
    - *Mutual exclusiveness*
    - *Sensitive policies*
    - *Push control*
    - *Usage control*

Evaluation Method

- One scenario for each criterion.
- If the scenario can be expressed and encoded in a language => the language fulfills the corresponding criterion

Minimal Information Disclosure

- Objects that are least sensitive will be selected first for disclosure

  If she can choose, Alice would rather disclose her birth certificate than her driver's license to prove her age.

Minimal Information Disclosure (cont')

- Protune: meta-attributes

```
birth_certificate.sensitivity:low
driver_licence.sensitivity:high
negotiator.selection_method:order(sensitivity)
```

- Ponder: sets of domain expressions and policy types

```
type auth+ preferType (target t, domain tsub) {
    subject s = *;
    target t->select(t1|t1.hasDomain(tsub));
    action  read();
}
inst auth+ preferBirthCert = preferType(
    /credential/birthcertificate
    + /credential/driverlicense,
    /credential/birthcertificate);
```

Minimal Information Disclosure (cont')

- Trust-X can only partially implement
  - Supports sensitivity for credentials …
    - as a measure of how many attributes that credential has
  - … but **not** for attributes and concepts

- Rei, KeyNote, and APPEL can not implement
  - Rei could attach sensitivities to credentials, but cannot select the credential based on sensitivity.

Mutual Exclusiveness

- Control the concurrent release of data objects that might be sensitive together.

  Alice does not want to disclose her Microsoft Most Valuable Employee (MMVE) credential and her Apple Best Developer (ABD) credential to the same party.

- Protune
  - Uses meta-level constraints

```
← credential(X, "Microsoft"),
  credential(Y, "Apple"),
  X.type: "MMVE", Y.type: "ABD"
```

- Trust-X
  - Uses private groups
  - Limitation: a Pg can never be released.

```
Pg = {{MMVE_cred, ABD_cred}, ...}
```

Mutual Exclusiveness (cont')

- Rei and Protune
  - Implement a history of disclosed credentials
  - Rei: consequences

```
action(access_MMVE, [R, MMVE],
    not-disclosed(R,ABD), assert(disclosed(R, MMVE)))
action(access_ABD, [R, ABD],
    not-disclosed(MMVE), assert(disclosed(R, ABD)))
```

  - Ponder: external function that log the disclosed credentials
- APPEL and KeyNote cannot implement

Type of Classification

- Make sensitive objects known and express hierarchies of sensitive objects, semantic equivalence, relationships, and more
  - E.g. equivalent attributes from different credentials

- Results
  - Protune, Rei, and Trust-X: external ontology of resources
  - Ponder: taxonomy- or directory-like structure called domain expressions
  - APPEL: vocabulary of the P3P-standard
  - KeyNote: none

Granularity of Objects

- Express the granularity of sensitive objects

  Alice wants to protect information about her age (concept), her electronic passport (credential), her social security number on her electronic id (credential attribute), and her diary file (regular resource).

- Results
  - In general, driven by the type of classification
  - KeyNote: granularity defined by the application

Access Control

- Control to whom sensitive objects are released

  Alice allows access to her diary only to a psychiatrist.

- Results:
  - All can address this criterion

Sensitive Policies

- Control the release of policies that might themselves be sensitive

  a) Alice discloses the policy protecting her diary only to a doctor.

  b) Alice never disclose the policy protecting her diary.

- Results
  - Protune: “sensitivity” can be applied to policies
  - Trust-X: policy preconditions, but has limitations
  - Rei and Ponder: policies are first class objects
  - KeyNote and APPEL cannot implement
    - But in P3P user policy (preferences) are never disclosed.

Push Control

- Address deadlocks due to sensitive policies that cannot be released

  Psychiatrist Eve will always push her psychiatrist credential when she contacts her patients.

- Results
  - Protune: provisional predicates
  - Rei and Ponder: obligations
  - KeyNote: relies heavily on the application
  - Trust-X and APPEL cannot implement

Usage control

- Control how data should be handled by receiving party

  Requesters that have received information about her diary are obliged by Alice to purge this information from their system within 3 months.

- Results
  - Rei and Ponder: obligations for the receiving party
  - APPEL: can implement, but is constraint by the P3P vocabulary.
  - Protune, Trust-X, and KeyNote can not implement

- Usage control policies must be enforced by the receiving party!

Evaluation Summary

Criterion	Protune	Rei	Trust-X	KeyNote	Ponder	APPEL
Type of classification	Ontology	Ontology	Ontology	None	Taxonomy of domain expressions	P3P taxonomy
Granularity of objects	Provided by ontology	Provided by ontology	Provided by ontology	Provided by application	Provided by taxonomy	Provided by taxonomy
Access Control	Yes	Yes	Yes	Yes	Yes	Yes, restricted by P3P vocabulary.
Minimal information disclosure	Yes	No	Yes, partial	No	Yes	No
Mutual exclusiveness	Yes	Yes	Yes, with exceptions	No	Yes, externally	No
Protect sensitive policies	Yes	Yes	Yes, with exceptions	No	Yes	No
Push contol	Yes	Yes	No	Yes	Yes	No
Usage control	No	Yes	No	No	Yes	Yes, restricted by P3P vocabulary.

Conclusions

- Identity theft and uncontrolled exposure of private information are major risks for users
- How well can policy languages address these risks?
- Results of our scenario-based evaluation
  - Policy languages address quite well the criteria
    - Some with special constructs
    - Some have limitations
  - Minimal information disclosure and usage control still poorly addressed

Future Work

- Usable security
  - Can lay user Alice effectively use these languages to protect their privacy?
  - Evaluate tools for setting policies.