PrimeLife Policy Language
Scenario

Personal Data (PD)
Non-certified
Certified: credentials

Data Subject

request credential
credential

request resource
resource

request personal data
personal data

Data Controller

Resources
Non-personal content, services,…
Collected personal data

Issuer

request personal data
personal data

Downstream Data Controller
Types of policies

**Specific Policy:**
over specific personal data (e.g. birth date)
- **Access control policy (ACP):**
  who can access (e.g. PrivacySeal silver)
- **Data handling preferences (DHPrefs):**
  how is to be treated when revealed
  - **Authorizations** (e.g. marketing purposes, forwarded to PrivacySeal gold)
  - **Obligations** (e.g. delete after ≤2y)

**Generic Preferences:**
DHPrefs over implicitly revealed personal data (e.g. IP address, cookies,...)
- **Authorizations** (e.g. admin purposes)
- **Obligations** (e.g. delete after ≤2y)

**Specific Policy:**
over specific resource (e.g. BuyService)
- **Access control policy (ACP):**
  who can access
  - credentials to possess (e.g. ID card)
  - personal data to reveal (e.g. nationality)
  - conditions to satisfy (e.g. age>18)
- **Data handling policy (DHP):**
  how revealed personal data will be treated
  - **Authorizations** (e.g. marketing purposes, forwarded to PrivacySeal gold)
  - **Obligations** (e.g. delete after 1y)

**Generic Policy:**
DHP over implicitly revealed personal data (e.g. IP address, cookies,...)
- **Authorizations** (e.g. admin purposes)
- **Obligations** (e.g. delete after 1y)
Specific Policy:
Resource PD1:
• DHPrefs1: ...
Resource cred.PD2:
• ACP2: ...
• DHPrefs2: ...

Generic Policy:
GDHPrefs: ...

---

Data Subject

request generic policy

Policy Engine

request specific policy(R)

Policy Engine

request R, proof(ACP), personal data

---

Data Controller

Specific Policy:
Resource R:
• ACP:
  • own cred
  • reveal PD1 under DHP1, cred.PD2 under DHP2
    • where φ
  • DHP1: ...
  • DHP2: ...

Generic Policy:
GDHP: ...

---
Interaction overview: generic policy negotiation

**Data Subject**

- **Specific Policy:**
  - Resource PD1:
    - DHPrefs1: …
  - Resource cred.PD2:
    - ACP2: …
    - DHPrefs2: …

- **Generic Policy:**
  - GDHPrefs: …

**Data Controller**

- **Specific Policy:**
  - Resource R:
    - ACP:
      - own cred
      - reveal PD1 under DHP1, cred.PD2 under DHP2
      - where φ
    - DHP1: …
    - DHP2: …

- **Generic Policy:**
  - GDHP: …

**Policy Engine**

- request generic policy
  - (fixed location, à la P3P safe zone)

- **generic sticky policy**

- GSP = match(GDHP, GDHPrefs)

- GSP ≤ GDHP ?

- request specific policy(R)

- specific policy matching

- request R, proof(ACP), personal data

- R
Interaction overview: specific policy negotiation

Data Subject

Specific Policy:
Resource PD1:
• DHPrefs1: ...

Resource cred.PD2:
• ACP2: ...
• DHPrefs2: ...

Generic Policy:
GDHPrefs: ...

Data Controller

Specific Policy:
Resource R:
• ACP:
  • own cred
  • reveal PD1 under DHP1, cred.PD2 under DHP2
  • where φ
• DHP1: ...
• DHP2: ...

Generic Policy:
GDHP: ...

request generic policy

request policy(R)

satisfy ACP ? (identity selection)

satisfy ACP2 ? SP1 ≤ DHP1 ? SP2 ≤ DHP2 ?

proof(ACP2)

request R, proof(ACP), personal data
Interaction overview: resource request

Data Subject

Specific Policy:
Resource PD1:
• DHPrefs1:
  • Auths1’, Obls1’
Resource cred.PD2:
• ACP2
• DHPrefs2:
  • Auths2’ ⊇ DACP2
  • Obls2’

Generic Policy:
GDHPrefs

Data Controller

Specific Policy:
Resource R:
• ACP:
  • own cred
  • reveal PD1 under DHP1,
    cred.PD2 under DHP2
  • where φ
  • DHP1: …
  • DHP2: …

Generic Policy:
GDHP: …

Policy Engine

request generic policy
generic policy matching

Policy Engine

request specific policy(R)
specific policy matching

Policy Engine

request R, proof(φ), PD1, cred2.PD2

φ, PD1, cred2.PD2 ⇒ ACP ?
check proof(φ)

R
Authorizations and obligations

- General principle: provide
  - wrapper for user-extensible vocabularies
  - basic pre-defined vocabulary
- Authorizations
  - “use for purpose”
    • user-extensible (OWL?) ontology of purposes,
    • basic pre-defined ontology available
  - “forward to ACP” = downstream access control
- Obligations
  - general structure: do action when trigger (from start to end)
  - pre-defined actions:
    • “delete data”
    • “anonymize data”
    • “notify data subject”
    • “write to (secure) log”
  - pre-defined triggers:
    • at time, periodic
    • data access, data deletion
    • data loss, obligation violation
    • aliens landing on earth
automated matching of any two data handling preferences/policies via “less permissive than” relation (\(\leq\)) defined on

- authorizations, e.g.
  
  \[
  \text{use for \{delivery\}} \leq \text{use for \{delivery,marketing\}}
  \]

- triggers, e.g.
  
  \[
  \text{trigger at 2010/01/01} \leq \text{trigger at 2010/12/31}
  \]

- actions, e.g.
  
  \[
  \text{delete firstname, lastname} \leq \text{delete firstname}
  \]

- obligations
  
  \[
  o_1 = (a_1, t_1, v_1) \leq o_2 = (a_2, t_2, v_2) \iff (a_1 \leq a_2) \land (t_1 \leq t_2) \land (v_1 \leq v_2)
  \]

- sets of authorizations and obligations
  
  \[
  O_1 \leq O_2 \iff \forall o_1 \in O_1 \exists o_2 \in O_2 : o_1 \leq o_2
  \]

- data handling policies
  
  \[
  P_1 = (A_1, O_1) \leq P_2 = (A_2, O_2) \iff A_1 \leq A_2 \land O_1 \leq O_2
  \]
Embedding into XACML
Summary

- Privacy enhancements
  - step-wise interaction: generic policy, specific policy, resource
  - reveal attributes vs. prove condition holds
  - two-sided data handling policies/preferences, automated matching
  - user-extensible authorization/obligation vocabularies, basic vocabularies provided

- Credential-based access control
  - attributes grouped in credentials
  - technology independence
  - policy sanitization

- Based on existing standards: XACML & SAML