PPL & Friends:
Privacy Policies in PrimeLife
The whole idea

- Forms & natural-language privacy policies are so 20th century

- Want automation for
  - Requesting attribute values
  - Filling out attribute values
  - 3rd party attestation of attributes
  - Interpreting privacy policies
Policy languages

- Need for standardized, machine-interpretable languages to express
  - what information server needs
  - how server will treat this information
  - what information user is willing to give away to whom
  - how user expects her information to be treated

- Policy languages vs. protocol languages
Overview of this talk

- Privacy policy language: P3P
- Access control policy language: XACML
- Attribute & identity claims protocol language: SAML
- PrimeLife Policy Language (PPL)
P3P

- Platform for Privacy Preferences (P3P)
- Developed by World Wide Web Consortium (W3C)
- Standard protocol language for sites to describe their privacy policies
- Enables development of user agents that
  - match P3P policies against user’s preferences
    - Internet Explorer 6: cookies preferences tab
    - AT&T/CMU Privacy Bird: preferences in APPEL language
  - summarize/visualize P3P policies
- Critiques
  - almost exclusively used for cookies management
  - too complex to create/read P3P policy
  - lack of enforcement, consumer/industry/browser support
P3P data flow

User (Agent) Web Server

- request policy reference file (@well-known location, e.g., example.com/w3c/p3p.xml)
- policy reference file
- request policy
- policy
- match policy against preferences
- inform user
- request web page
- web page
P3P policy format

- Legal entity & contact info of web server
- Ability for users to access their own data
- Applicable law, court, and sanctions in case of disputes
- Advantages to user of accepting policy
  - Indicates no identifiable data will be collected
  - Purposes of data collection and usage
  - Recipients of collected data
  - Kind of retention: none, business-practice, indefinite, ...
- Which (categories of) data will be transferred or inferred
**XACML**

- **XACML**
  - eXtensible Access Control Markup Language, developed by OASIS
  - Industry standard specifying
    - XML-based access control policy language
    - XML-based access request/response protocol language
    - processing model
  - Access decisions based on attributes of
    - Subject (e.g., username, role)
    - Protected resource (e.g., file name, URL, content,…)
    - Action (e.g., read, write,…)
    - Environment (e.g., date, time,…)
  - Extension points: can define new attributes, data types, functions, obligations, rule/policy combining algorithms
XACML data flow

1. policy

Policy Administration Point (PAP)

2. access request

access requester

3. request

Policy Decision Point (PDP)

4. request notification

5. attribute queries

6. attribute query

context handler

7a. subject attributes

7a. subject attributes

7c. resource attributes

7d. environment attributes

Policy Information Point (PIP)

8. attribute

9. resource content

resource

10. attributes

11. response context

12. response

Policy Enforcement Point (PEP)

13. obligations

obligations service
XACML policy format

Built-in rule/policy combining algorithms:
- Permit overrides
- Deny overrides
- First applicable
- Only one applicable

“Permit” or “Deny”
Restricts the applicability of this rule
Restrictions on subject who performs request
Restrictions on protected resource
Restrictions on requested action
Restrictions on environment attributes
Arbitrary other restrictions (boolean function over atts)

Obligations PEP has to adhere to when access granted
SAML

- Security Assertion Markup Language, by OASIS
- Protocol language for communicating
  - user authentication information
  - user attribute information
  - authorization decision information
- Main use cases
  - Single sign-on (SSO)
  - Identity federation
  - Attribute provision
- SAML profile of XACML
  - Translate SAML attributes into XACML attributes
  - Use SAML to carry XACML policies, authorization decisions,…
SAML single sign-on

Identity Provider (IdP) → User
1. authenticate

User → Service Provider (SP)
2. access resources
SAML identity federation

1. obtain pseudonym for jdoe@SP1=johnd@SP2
2. login as jdoe@SP1=johnd@SP2
SAML attribute provision

1. obtain assertion for status="GoldMember"
2. show assertion status="GoldMember"
SAML assertion format

Assertion with unique identifier and time of issuing
SAML authority making the claim
XML signature of assertion by issuer
Principal that is subject of the assertion
Constraints on use of assertion (time, audience,…)
Abstract statement type
Statement that (and how) user authenticated at IdP
Statement that subject has specified attribute values.
PrimeLife Policy Language (PPL) scenario

Personal Data (PD)
Non-certified
Certified: cards

Data Subject

- request resource
- request personal data

Data Controller

- resource
- personal data
- credential

Issuer

- request credential

Downstream Data Controller

- request personal data
- personal data
- resource

Resources
Non-personal content, services,…
Collected personal data
Main features of PPL

- Card-based access control
  - attributes grouped in cards
  - technology independence
  - privacy friendly
    - reveal attributes vs. prove conditions
    - support anonymous credentials (Identity Mixer, U-Prove)
- Integrated data handling
  - two-sided detailed data handling preferences/policies
  - automated matching procedure
  - extensible vocabularies
  - downstream usage
- Policy sanitization
- Based on existing standards: XACML & SAML
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:
Resource PD1:
• DHPrefs1: ...
Resource cred.PD2:
• ACP2: ...
• DHPrefs2: ...

Generic Policy:
GDHPrefs: ...

Interaction overview

Data Subject

Policy Engine

request specific policy(R)

specific policy matching

Data Controller

Policy Engine

request R, proof(ACP), personal data

R

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

Generic Policy:
GDHP: ...
Card abstraction

- Card contains
  - list of attribute-value pairs
  - pre-evidence: technology-specific meta-data to
    - protect attribute integrity
    - prove card ownership

- Card issuer vouches for attributes wrt owner
  (identity/authority)

- Hierarchy of card types: define attributes contained

- Instantiating technologies: X.509, SAML, CardSpace, OpenID, Kerberos, trusted LDAP, Identity Mixer, U-Prove,…
Card-based access control example

- **Policy:** requirements on owned cards, e.g.,
  - `own p::Passport issued-by admin.ch, fgov.be, governo.it`
  - `own c::Creditcard issued-by visa.com, amex.com`
  - `reveal c.number, c.expdate`
  - `where p.name = c.name ^ p.bdate < today-18Y`
    - `^ c.expdate > today ^ p.expdate > today+1M`

- **Authentication** = *claim* over owned cards + *evidence*, e.g.,
  - `own p::Passport issued-by admin.ch`
  - `own c::Creditcard issued-by visa.com`
  - `reveal c.number = "1234567890"`
  - `reveal c.expdate = "31/12/2012"`
  - `where p.name = c.name ^ p.bdate < 05/05/1992 ^ p.expdate > 05/06/2010`
Features of card-based access control

Access control requirements language supporting

- Privacy preservation
  - for user: *minimal* claim to be disclosed
    (selectively) reveal attribute ↔ predicate satisfied
  - for server: “sanitize” sensitive policies

- Multi-card claims
  but prevent “card mixing” through reference pointer to individual cards

- Technology independence
  but supporting advanced features, esp. anonymous credentials
Authorizations and obligations

- General principle: provide
  - wrapper for user-extensible vocabularies
  - basic pre-defined vocabulary

- Authorizations
  - “use for purpose”
    • user-extensible ontology of purposes,
    • basic pre-defined ontology available
  - “forward under policy” = 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
Obligation & authorization matching

automated matching of **any** two data handling preferences/policies via
“**less permissive than**” relation (≤) defined on

- authorizations, e.g.
  
  use for {delivery} ≤ use for {delivery,marketing}

- triggers, e.g.
  
  trigger at 2010/01/01 ≤ trigger at 2010/12/31

- actions, e.g.
  
  delete firstname, lastname ≤ 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
  
  \[ P_1 = (A_1, O_1) \leq P_2 = (A_2, O_2) \iff A_1 \leq A_2 \land O_1 \leq O_2 \]
Downstream usage matching

- **Proactive matching**

Data Subject

request PII, policy

PII, sticky policy

Prefs

Data Controller

Match

Sticky Pol

Downstream Data Controller

PII, sticky policy'

- **Lazy matching**

Data Subject

request PII, policy

PII, sticky policy

Prefs

Data Controller

Match

Sticky Pol

Downstream Data Controller

PII, sticky policy'
Obligation matching prototype
PPL policy format

Proposed data handling policies for revealed attributes
Requested authorizations
Promised obligations
Preferences how target resource should be treated
Agreed-upon sticky policy for target resource
Card-based access control for target resource
Cards to be presented
Required condition over card attributes
Actions to be performed, e.g., reveal attribute under referenced DHP, sign statement, limited spending,…
PPL claims format

One assertion per card, plus cross-card assertion

Reference to sticky policy associated to attribute value

New statement type to carry sticky policies

New statement type to carry conditions over attributes

New statement type to carry other (non-XML-signature) types of card evidence
PPL data flow
PPL data flow

1. preferences
2. resource request
3. resource request
4. claim request + policies
5. claim request + DH policies
6. claim request + DH policy + claims
7. card claim(s)
8. revealed PII + DH policy + claims
9. DH policy, prefs
10. (mis)matching sticky policy
11. Permit + (mis)matching sticky policy / Deny
12. claim(s) + (mis)matching sticky policies
13. selected claim + sticky policies
PPL data flow

Data Subject

- Resource Store
- Card Store
- Card Handler
- Context Handler
- PDP
- Preference Store
- DH Matching

Data Controller

- Resource Store
- Card Store
- Card Handler
- Context Handler
- PDP
- PAPI
- Policy Store
- DH Matching

Flow:
1. Resource request
2. Claim + sticky policies
3. Permit/Deny
4. Authorizations, downstream policy (from sticky policy)
5. Obligations (from sticky policy)
6. Obligation Enforcement

Notes:
- 14. Resource request, claim + sticky policies
- 15. Resource request, claim + sticky policies
- 16. Permit/Deny
- 17. PII
- 18. Obligations (from sticky policy)
- 19. Authorizations, downstream policy (from sticky policy)
PPL data flow (downstream)

1. PII request + DH policies + claims
2. requested PII + DH policy + claims
3. DH policy, prefs
4. matching sticky policy
5. Permit + sticky policies / Deny
6. PII
7. PII + sticky policies
8. PII
9. obligations (from sticky policy)
10. authorizations, downstream policy (from sticky policy)
Conclusion: main features of PPL

- Card-based access control
  - attributes grouped in cards
  - technology independence
  - privacy friendly
    - reveal attributes vs. prove conditions
    - support anonymous credentials (Identity Mixer, U-Prove)

- Integrated data handling
  - two-sided data handling preferences/policies
  - automated matching procedure
  - extensible vocabularies
  - downstream usage

- Policy sanitization

- Based on existing standards: XACML & SAML
W3C Workshop on Privacy and data usage control

- http://www.w3.org/2010/policy-ws/
- October 4-5, Boston
- Position paper deadline: September 10