



UNIVERSITÀ DEGLI STUDI DI MILANO

Identity Management for Web-based Services

Marco Cremonini, Ernesto Damiani,
Sabrina De Capitani di Vimercate, Pierangela Samarati

*Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Crema (CR)*

CASSIS'05



The Problem

- ▶ All current approaches to Web-based Services Access Control rely on ***vocabularies*** that are shared among all the parties involved, and ***declarative policies*** specifying who is allowed to do what.
 - ▷ Suitable for corporate Intranets.
- ▶ ***Server-side prevalence***
 - ▷ While credentials are held by both sides, AC policy is server-side only.
 - ▷ Requestors must disclose all their credentials hoping to match the service owner's access policy.



Protecting Services on the Open Web

- ▶ On the Open Web, parties may make connections and interact without being previously known to each other.
 - ▷ Before any meaningful interaction starts, a certain level of **trust** must be established.
 - ▷ **Uncontrolled disclosing** client credentials may lead to profiling and privacy violations.
- ▶ **Trust sharing process** is bi-directional.
 - ▷ Both parties may have sensitive information that they are reluctant to disclose until the other party has proved to be trustworthy at a certain level.



Negotiated Access Control-1

Negotiated AC differs from traditional identity-based access control in the following aspects:

- ▶ **Trust** between two strangers (requestor and service provider) is established based on parties' properties
 - ▷ Proven through *negotiated disclosure of digital credentials or zero-knowledge proofs*.
- ▶ Every party can define **release policies** to protect sensitive resources.
 - ▷ Resources can include services accessible over the Internet, RBAC roles credentials, policies, and capabilities in capability-based systems.



Negotiated Access Control-2

- ▶ Policies describe what **properties** each party must **demonstrate** (e.g., ownership of a driver's license issued by an EU country) in order to gain access to a resource.
- ▶ The parties negotiate directly without involving trusted third parties, other than credential issuers. Since both parties have policies, peer-to-peer negotiation is appropriate for Web Services on the Open Web.
 - ▷ Instead of carrying out a one-shot authorization and authentication process, ***trust is established incrementally*** through a sequence of bilateral ***credential disclosure***.



Negotiation protocol

- ▶ A **negotiation process** is triggered when one party requests to access a resource owned by another party.
 - ▷ E.g., a remote requestor tries to access a Web-based service.
- ▶ The goal of a negotiation between a requestor and a service provider resp. holding policies P_r and P_s is:
 - ▷ Finding a sequence of resources $(C_{1.x}, \dots, C_{k.x}, , S_s)$ ($C_{i.x}$: credential belonging to party x , S : service), such that when credential $C_{i.x}$ is disclosed, its release policy has been satisfied by credentials disclosed earlier in the sequence.
 - ▷ E.g. $C_{2.s}$ is released iff policy P_s includes a rule like “disclose $C_{2.s}$ if $C_{1.r}$ has been provided by requestor”).
- ▶ The use of release policies together with a negotiation process seems to be the most promising approach to providing privacy-aware access to services on the Open Web.



Privacy Issues

► Privacy issues

PKI does not provide a comprehensive solution for avoiding unauthorized disclosure of personal information.

► Digital Identity Management System (privacy-aware)

New solutions (management of partial identity) with support of privacy related features: privacy, minimal disclosure, anonymity support, legislation support.



Nyms and Partial Identities

► Digital Identity

▷ Nyms

▷ Partial Identities

Non-disjoint concepts.

Nyms give users *different identities* to use when interacting with other parties in *different environments*.

Behind a nym, strong authentication tools such as tokens, smart cards, digital certificates, or biometrics associate individuals with their true digital identities.



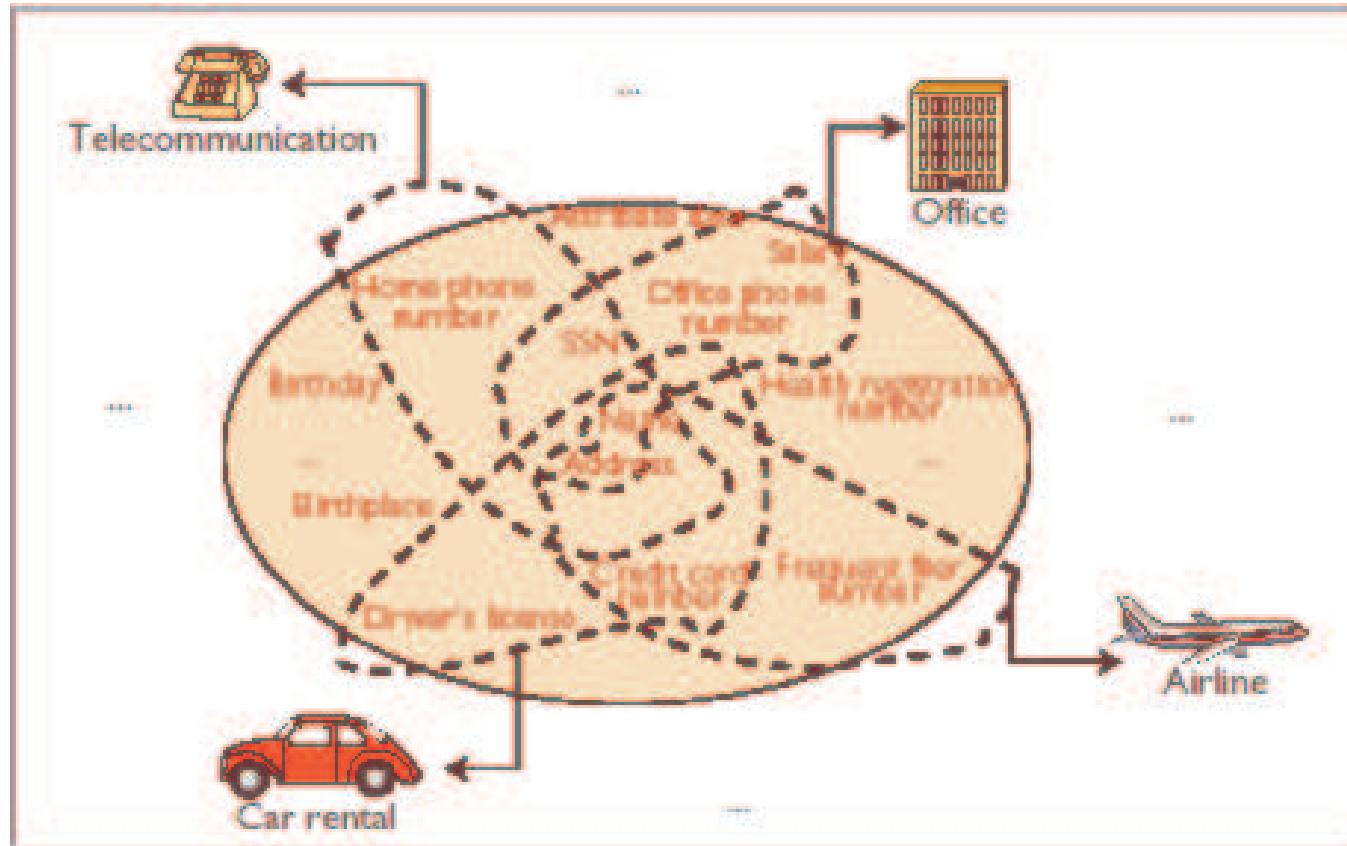
Partial Identities

Partial identities are any subset of the properties associated with users (such as name, age, credit-card number, or employment) that the user can select for interacting with other parties.

A partial identity can be *named* or *unnamed*, which means it might or might not be related to the user's true identity.



Partial Identities



Examples of partial identities. Each dashed line delimits a subset of the user's attributes that can be used as a partial identity when interacting with a party such as an airline or a car rental company.



Multiple and Dependable Digital Identity (MDDI) : Requirements

► **Reliability and dependability**

Protect users against forgery and related attacks while also guaranteeing to other parties (such as suppliers and brokers in an ebusiness transaction) that the users can meet transaction-related obligations.

► **Controlled information disclosure**

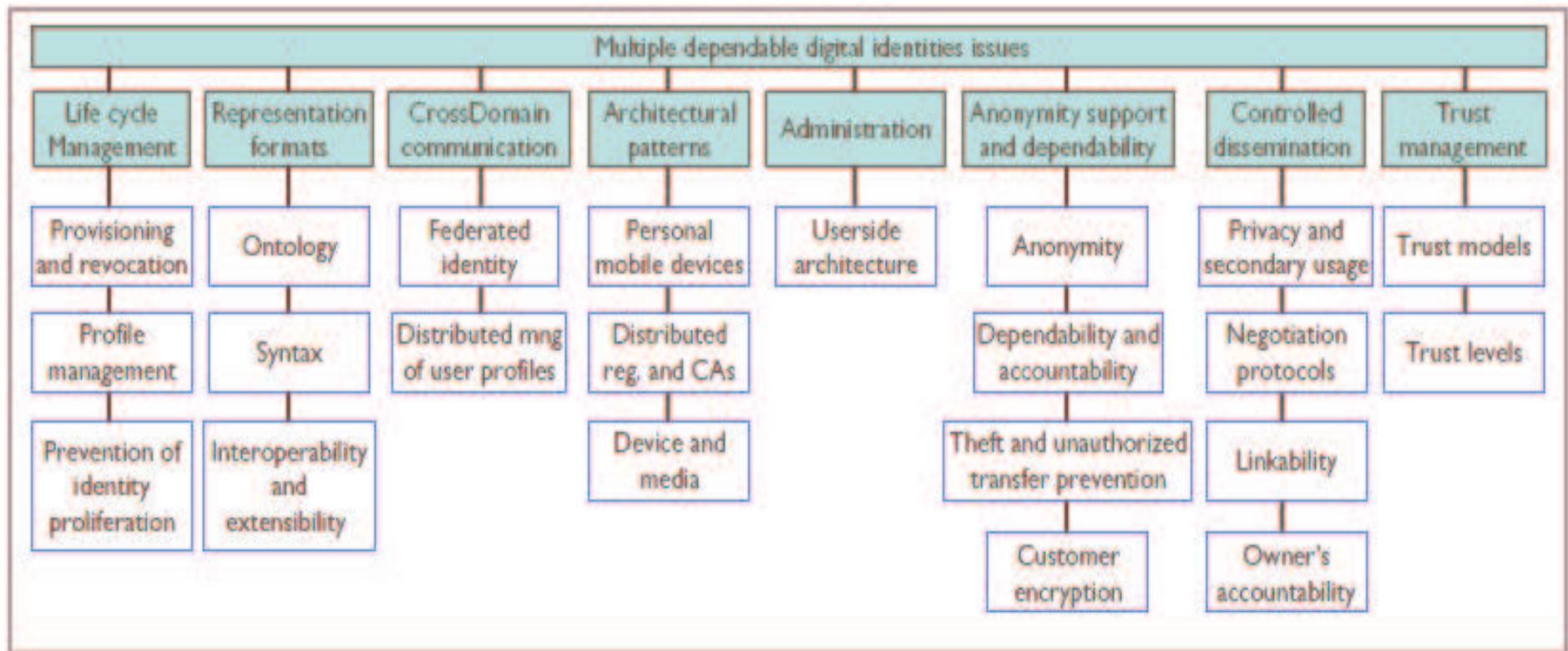
Users must have control over which identity to use in specific circumstances, as well as over its secondary use and the possible replication of any identity information revealed in a transaction.

► **Mobility support**

The mobile computing infrastructure must be able to take into account its own peculiarities (such as limited bandwidth and display size) to apply MDDI technology successfully.



MDDI System Design Issues



Multiple and dependable digital identity (MDDI) system design issues. The shaded boxes represent the main categories of problems and the clear boxes the specific issues to be addressed.



Identities Life Cycle Management

Open issues:

- ▶ **Provisioning:** users must be given the ability to efficiently obtain/create identities.
- ▶ **Revocation:** Identities may become obsolete and not applicable anymore.
- ▶ **Profile management:** users must be given the ability to manage their own identity information.
- ▶ **Prevention of identity proliferation:** impose soft/hard limit on the number of identities that can be associated with a single individual.



Digital Identities Representation

- ▶ **Ontology:** domain ontology, task ontology. Need to allow for sound reasoning about the identity equivalence and trust propagation.
- ▶ **Identity syntax:** profile-based digital identities and definition of identities on named hierarchical profiles.
 - ▷ **Attribute and credentials:** User profiles can be based on existing directory or certificate standards or by credentials.
 - ▷ **Minimal disclosure:** Privacy-friendly credentials (disclose only what's needed in a transactions).
- ▶ **Identity interoperability and portability:** Identity must be provided in a common interchange format. The identity management service must support extensible mapping between identities.
- ▶ **Identity extensibility:** Unlimited number of attributes that may be associated with an identity.



Cross-domain Identity Communication

- ▶ **Federated identity management support:** need to investigate techniques for identity composition and interchange. Challenge is to balance complete retrieval with privacy.
- ▶ **Distributed profile management:** retrieval of different chunks of identity information.
- ▶ **Distributed update support:** Distribution of profile information and update support.



Architectural Patterns

Traditional PKI context need to be adapted to trust management on digital identities. Need to scale to large user populations.

- ▶ **Personal mobile devices:** need to efficiently and securely storing identity information on mobiles.
- ▶ **Distributed registration and certification authorities:** provide support for cooperation and interoperability of multiple registration authorities and trust networks.
- ▶ **Devices and media for identity support:** different methods (biometric, smart card, secure mobiles).
- ▶ **Integration with other services:** assuming digital identity is trusted, access control policies should be applied to control access to resources.



Identity Administration

- ▶ **Credential update:** techniques, protocols, and tools for reliable update of credentials and efficient view computation.
- ▶ **Integration with personalization solution** to allow reuse of profiles.
- ▶ **Development of user-side architectures** with identity management tools.



Controlled Dissemination

- ▶ **Privacy and secondary usage control:** identity attributes should be enriched with privacy preferences. Current languages are still in their infancy.
- ▶ **Negotiation protocols** (e.g., avoid cases when identity is released and no service is given in return).
- ▶ **Linkability control** between transactions and different information releases.
- ▶ **Owner's accountability:** need to trust automated agents. MDDI services should provide adequate accountability to users.



Trust management

- ▶ **Control on single sign-on identity disclosure:** SSO approaches delegate to the infrastructure all decisions on identity communication. Solutions are needed enabling users to retain some control on such disclosure.
- ▶ **Trust models** to determine under which conditions a party can trust others for their security and privacy.
E.g., reputation models.
- ▶ **Support of trust levels** for instance non-sensitive information can be provided directly by the user, while for others certificates may be needed.



The PRIME Project



► Objective of the project

PRIME focuses on solutions for privacy-enhancing identity management that supports end-users sovereignty over their private sphere and enterprises privacy-compliant data.

Main characteristics:

- ▷ ***anonymity*** and ***end-user control***
- ▷ ***flexible and expressive access control rules***
- ▷ ***client side restrictions***



Model and Format



► Privacy-aware access control model

New privacy-aware access control model together with an access control protocol for the communication of policies and of identity information among parties.

► Profiles and Ontologies

- ▷ **Profiles** associated with subjects and objects define the name and value of some properties that characterize the subjects and objects.
- ▷ **Ontologies** (Subject and Object) contain terms that can be used to make generic assertions on subjects and objects.



The role of semantics



- ▶ The negotiation process can be guided and errors prevented using **formal specifications** of relations among **credentials** (e.g., equivalence of name field in passport and identity card) and between credentials and **zero-knowledge proofs** (e.g., proving one has an Italian driving licence is equivalent to proving $\text{age} \geq 18$ without disclosing the actual age)
- ▶ Advanced ontology-based **metadata** could do the trick
- ▶ Need for **standardization**



Privacy Policies



- ▶ **Access control policies.** They govern access to data/services managed by the user/server-side party (as in traditional access control).
- ▶ **Release policies.** They govern release of properties/credentials/PII of the party and specify under which conditions they can be disclosed.
- ▶ **Data handling policies.** Specified by the user that decide how his/her personal information must be managed by the counterpart (also called **Sticky policies**).
- ▶ **Sanitized policies.** They provide filtering functionalities on the response to be returned to the counterpart to avoid release of sensitive information related to the policy itself.

The first version of the language (Feb. 05), integrated in PRIME Prototype V1, deals with access control and release policies only .



Access Request



Each request is characterized by:

- ▶ the **Subject** that makes the request, defined as a pair:
 - ▷ **User**: identifier of the human entity (possible *anonymous*) that connected to the system and submitted the request.
 - ▷ **Purpose**: reason for which data are being requested and will be used (e.g., Commercial, Teaching, Research, etc.).
- ▶ the **Action** that is being requested (e.g., read, write, download).
- ▶ the **Object** on which the subject wishes to perform the action.



Access Request: Examples



<tom.smith, research>, read, object1
user *tom.smith* requires to *read object1* for *research* purposes.

<john.doe, _>, read, object1
user *john.doe* with undeclared purpose requires to *read object1*.

<_, _, >, browse, object5
an *anonymous* user with undeclared purpose requires to *browse object5*.



Subject and Object Information



- ▶ Each party's **portfolio** contains properties that the party can use to gain (or offer) services:
 - ▷ **data declarations**: statements issued by the party.
 - ▷ **credentials**: statements issued and signed (i.e., certified) by authorities trusted for making the statements.
- ▶ To refer to specific data in a credential we introduce the concept of **credential term**.
 - ▷ A credential term is an expression of the form
credential name (predicate list)
- ▶ **Users** and **Objects** can be grouped into *groups*.
- ▶ **Ontologies** represent relationships (**part-of** and **is-a**) among attributes and credentials to establish what credentials can be provided to fulfill a declaration or credential request.



Basic elements of the language-1



- ▶ A predicate **declaration** where the argument is a list of predicates of the form **predicate name** (*arguments*) ;
- ▶ A binary predicate **credential** where the first argument is a credential term and the second argument is a public key term. Intuitively, a ground atom **credential** (*c*; *K*) is evaluated to true if and only if there exists a credential *c* verifiable with public key *K*.
- ▶ A set of standard built-in mathematic predicates, such as **equal()**, **greater_than()**, **less_than()**, and so on.



Basic elements of the language-2



- ▶ A set of **location-based** predicates of the form **predicate name** (*arguments*) ;
- ▶ A set of **trusted-based** predicates of the form **predicate name** (*arguments*) ;
- ▶ A set of *non predefined* predicates that evaluate information stored at the site.



Obligation



- ▶ An obligation establishes how a released personal data must be managed by the counterpart.
- ▶ Obligations are associated to release policies and linked to released data.
- ▶ Types of obligations:
 - ▷ **Transactional** Obligation: to be immediately enforced (e.g. delete PII data as soon as the transaction is over)
 - ▷ **Data Retention** and **Handling** Obligation: driven by time-based or specific events (e.g. delete PII data after x days from the reception, or delete PII data after 3 accesses)



Access Control rules - 1



***subject WITH subject-expression CAN action
FOR purpose ON object WITH object-expression
IF conditions FOLLOW obligations***

- ▶ An access is granted if there is satisfaction of at least one of the AC rules that apply to the given request.
- ▶ **Rule structure:**
 - ▷ **subject** identifies the subject to which the rule refers.
 - ▷ **subject-expression** is an expression defining conditions on the subject that must be evaluated on the subject's portfolio (declarations and credentials);



Access Control rules - 2



*subject WITH subject-expression CAN **action**
FOR **purpose** ON **object** WITH **object-expression**
IF conditions FOLLOW obligations*

- ▷ **action** is the action to which the rule refers (e.g., read, write, etc.).
- ▷ **purpose** is the purpose to which the rule refers and represents how the data is going to be used by the recipient.
- ▷ **object** identifies the object to which the rule refers.
- ▷ **object-expression** is an expression defining conditions on the object that must be evaluated on object's data (stored in DB or other repositories).



Access Control rules - 3



*subject WITH subject-expression CAN action
FOR purpose ON object WITH object-expression
IF conditions FOLLOW obligations*

- ▷ **conditions** is a boolean expression of generic conditions that an access request to which the rule applies has to satisfy. For instance, trust properties, or the user's consent to disclose.
- ▷ **obligations** is a boolean expression of obligations that the server must follow when manage the information/data/PII. E.g., all accesses against a certain type of data for a given purpose must be logged.



Examples of rules - 1



- ▶ A registered user who works as a doctor, can read for research the patientXXX-Data with the agreement of the patient.

```
registeredUsers WITH declaration(equal(user.work,  
"doctor")) CAN read FOR research ON patientXXX-Data  
with declaration(equal(object.patient_agreement,  
yes)) IF no-condition FOLLOW no-obligation
```



Examples of rules – 2



- Anybody with $\text{age} > 18$ can book two seat for the movie "Full Metal Jacket" giving a credit card and accepting a contract. Server must delete credit card after the end of transaction.

```
Anonymous WITH declaration(greater than(user.age, 18))  
CAN book FOR no-purpose ON movie with  
declaration(equal(object.title, "Full Metal Jacket"))  
IF sign_Contract() and credential(credit_card, K)  
FOLLOW delete(credit_card)
```



Open problems



► *Policy correctness*

- ▷ Unfortunately, real-world policies tend to be very **complex**.
- ▷ Policy errors could allow outsiders to gain **inappropriate access** to services, possibly inflicting huge and costly damages.

► *Leakage in automated negotiation*

- ▷ Very specific policies may **leak information** about what we want to protect.
- ▷ **Malicious services** may try to get information which is not relevant to the resource the requestor needs to access.