Peertrust

Daniel Olmedilla
L3S Research Center

Trust on the Web - Developers Day
Thirteenth International World Wide Web Conference
New York, USA, 22nd May. 2004
Outline

- Introduction
- Trust Negotiation
- PeerTrust Language
- How PeerTrust Works
- Architecture
- Further Work & References
Outline

- Introduction
  - Trust Negotiation
  - PeerTrust Language
  - How PeerTrust Works
  - Architecture
  - Further Work & References
Traditional Access Control for Decentralized Systems

Assumption: I already know you---you have a local account!
Outline

- Introduction
- Trust Negotiation
- PeerTrust Language
- How PeerTrust Works
- Architecture
- Further Work & References
Trust Negotiation

- Trust is based on parties’ properties
- Every party can define access control policies to control outsiders’ access to their sensitive resources
- Establish trust iteratively and bilaterally by the disclosure of certificates and by requests for certificates
Step 1: Alice requests a service from Bob

Step 2: Bob discloses his policy for the service

Step 3: Alice discloses her policy for VISA

Step 4: Bob discloses his BBB credential

Step 5: Alice discloses her VISA card credential

Step 6: Bob grants access to the service
Outline

- Introduction
- Trust Negotiation
- PeerTrust Language
- How PeerTrust Works
- Architecture
- Further Work & References
PeerTrust Language

Based on guarded distributed logic programs

Examples:

E-Learn:
freeEnroll(Course, Requester) @ Requester →
policeOfficer(Requester) @ 'California State Police' @ Requester,
rdfType(Course, 'http://.../elena#Course'),
language(Course, 'es'),
creditUnits(Course, X),
X <= 1.

Alice:
policeOfficer('Alice Smith') @ 'California State Police' @ Requester →
member(Requester) @ 'Better Business Bureau' @ Requester
| signedBy ['California State Police'].
Outline

- Introduction
- Trust Negotiation
- PeerTrust Language
- How PeerTrust Works
- Architecture
- Further Work & References
How It Works

To gain access to a resource, I must satisfy its access control policy, which describes
- Which certificates must be presented to gain access to it
- Who is responsible for obtaining and presenting these certificates

To obtain (access to) a certificate, I must satisfy its access control policy, which specifies ... --and so on, recursively--

This leads to the distributed construction of a **certified proof** tree
- Root represents access to the grid resource
- Leaves represent disclosed credentials / signed facts
- Internal nodes represent heads of policies / signed rules, children represent literals in their bodies
- Inference rules govern allowable edges in the tree
Outline

- Introduction
- Trust Negotiation
- PeerTrust Language
- How PeerTrust Works
- Architecture
- Further Work & References
Trust Agent Architecture

- **Inference Engine**
- **Policies**
- **Ontologies**
- **Metadata**
- **Credentials**
- **Negotiation module**
- **Strategy evaluator**
- **Credential verification**
- **Interface**

TRUST AGENT
Trust Agent Architecture Impl.

Diagram:
- **Peertrust policies**
- **RDFS**
- **RDF instances**
- **Minerva Prolog**
- **Negotiation module**
- **Strategy evaluator**
- **Credential verification**
- **Interface**

Connections:
- X.509 + ext
- TRUST AGENT

SSL connections:
- Input/Output
Negotiation module algorithm
Trust Negotiation among Peers on the Web

Designed a policy language to express trust negotiation
- Delegation, policy protection, negotiation strategies
- Based on guarded distributed logic programs

Developed a run-time system for automated trust negotiation
- Based on Prolog meta interpreter embedded as Java library in Applet / Server (WWW) or Peer-to-Peer environment

Use RDF in policies
- Use of metadata information into policies
- E.g. access(Resource, Requester) ← dcCreator(Resource, Requester)

RuleML import/export facility
- Policies encoded in RuleML
Network Diagram
Outline

- Introduction
- Trust Negotiation
- PeerTrust Language
- How PeerTrust Works
- Architecture

- Further Work & References
Further work

- User-friendly parser for policies
- Integrate XSB Prolog inference engine (opensource)
- Loop detection during negotiation
- Extensions to
  - Semantic Web Services
  - GRID
- Cashing of credentials
References

Project page:
- [http://www.learninglab.de/peertrust/](http://www.learninglab.de/peertrust/)

Reports
  No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web
  European Semantic Web Symposium, May 2004, Heraklion, Greece

- J.Basney, W.Nejdl, D.Olmedilla, V.Welch, M.Winslett
  Negotiating Trust on the Grid
Questions?