PEERTRUST: A LANGUAGE FOR TRUST MANAGEMENT IN P2P SYSTEMS

M. WINSLETT, C. ZHANG

Trust Management in Open Systems

How can I specify my restrictions on access, usage, and information redissemination, and tell whether my conditions are satisfied?

How can we describe our purpose of access to Alice’s info/service, find the credentials she wants, control what she does with info we release to her, and limit info Alice pushes us?

PeerTrust Solution Approach

- Logic-based policy language for specifying purpose-based access, release, redissemination, and exposure control in distributed systems
- Formal semantics and efficient runtime system that support local autonomy during trust establishment

Key contributions: proof-carrying logical signatures on assertions: simple, unified treatment of purpose, exposure control, redissemination control, and access control

Purpose of Access Versus Access Control

- To meet legislative and corporate requirements, I need:
  - Distributed, scalable, credential-based trust management for my legacy and trust-aware services
  - Ways to specify and reason about purpose of access
  - "sticky" policies to control access, release, and redissemination of information

PeerTrust in Action

- Label indicates the purpose of the query (in this case, the response’s recipient)
- Logical signature indicates the authority vouching for this certificate

Proof-Carrying Logical Signatures

- UUIC delegates student identification to the registrar office:
  - (isStudent(X) ← isStudent(X) @ [Registrar] @ UUIC)
- The registrar office issues Alice a digital student ID:
  - isStudent(Alice) @ [Registrar]

Logical Signature: isStudent(Alice) @ UUIC
- (isStudent(X) ← isStudent(X) @ [Registrar] @ UUIC)
- Logical equality:
  - isStudent(Alice) @ UUIC

The book store gives a special discount to UUIC students.

Bob, do you have an Amazon gift certificate that I can use?
-Sorry, Bob. I do not have one but it cannot be given to Alice.

Amazon pushed a gift certificate to Bob. On arrival, her filters added her own label to the certificate: (isFriend("Alice") @ Amazon [recipient = "Alice"] @ [Registrar])

Bob’s filters rewrite the query to use "GIFTCertificate(s) @ Amazon [recipient = "Alice"]"
- Amazon will process the request and Bob receives the certificate.