Attribute-based Interactions in a Distributed AAI: The PAPI Experience

Diego R. López
Rodrigo Castro-Rojo
RedIRIS

Trustbus03 – Prague, September 2003

Outline

- What is PAPI
  - Components and protocols
- Current usage scenarios
  - How user requirements drive system evolution
- Enhancing user experience
  - Authentication interface
  - Seamless access
- Extending the trust fabric
  - Authorization mechanisms
  - Access token rotation
- Current status

What is PAPI

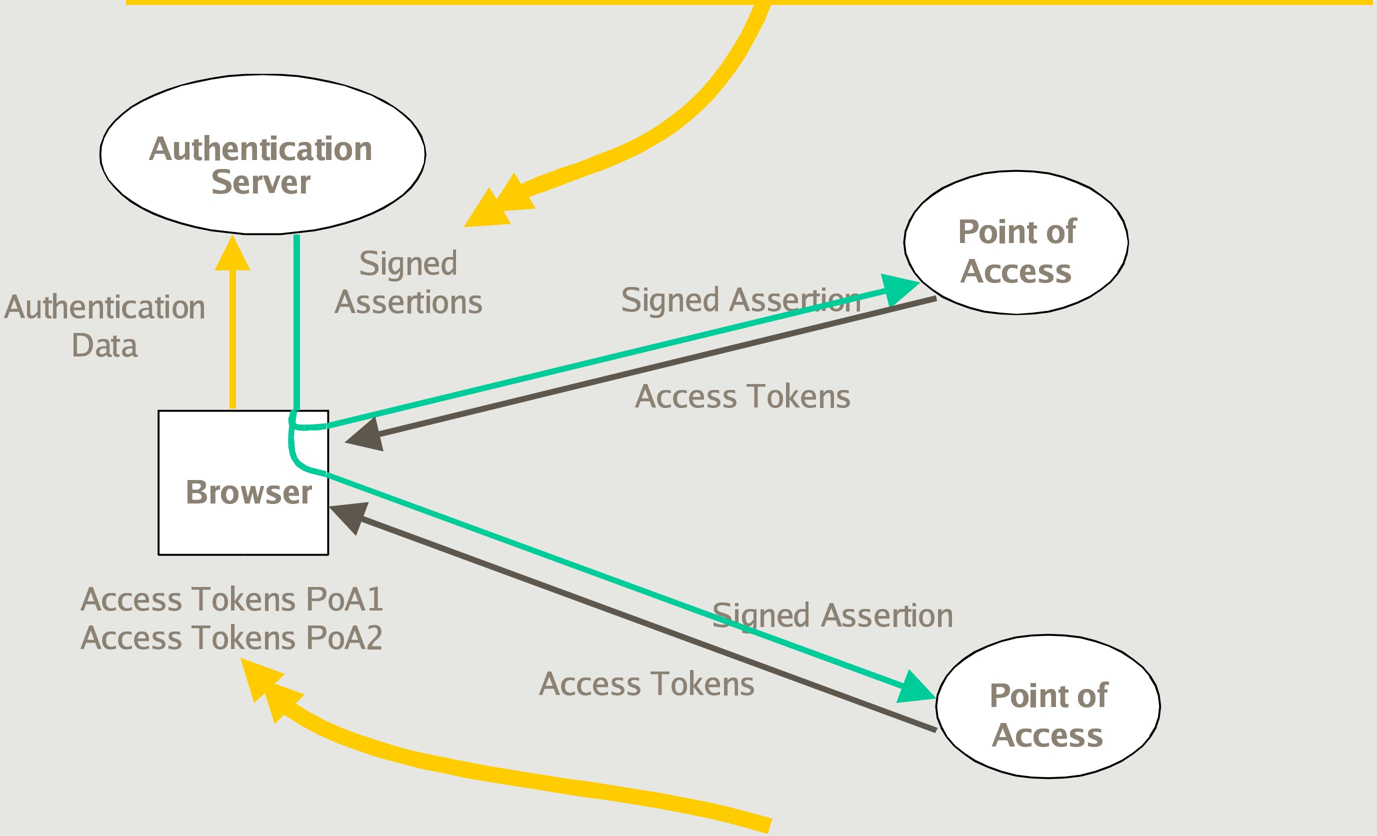
- PAPI is a distributed access control system for Internet information resources
  - Usable for intra- an inter-realm scenarios
  - Based on the federated administration and active privacy principles
  - Based on standard HTTP procedures and public key cryptography
- Is the only system able to support federated authN/authZ currently in production
  - Pursuing interoperability with other similar intiatives

The components of PAPI

- The Authentication Server (AS)
  - Provides users with a (local) single authentication point
- The Point of Access (PoA)
  - Performs actual access control by means of temporary cryptographic tokens, encoded as HTTP cookies
- The Group-wide Point of Access (GPoA)
  - Combines a group of PoAs with similar access policies
  - Intended to simplify AS-PoA interactions and PoA operation

The PAPI base protocol

K_{privAS}(assertionsOnUser+ASID+destination+signatureTime+TTL)



Hcook: K_H(userHash + destination + expirationTime + randomBlock)

Lcook: K_L(destination+creationTime)

Current usage scenarios

- Single sign-on for corporate applications
  - One-step authentication for any Internet-available resource (internal or external)
- Single sign-on for remote services
  - Most popular use in university libraries and consortia
  - Has required the development of proxy elements
- Inter-realm access
  - Keep user identity data inside the user's realm
  - Simplify management of collaborative systems
  - Initial step for digital identity services

The evolution of PAPI

- Users value single sign-on above other features
  - Simpler and clearer user interfaces
  - Deep linking
  - Extension to other services
- Organizations require finer control once they realize the potential of attribute-based access control
  - Generalized intranet access
  - Cost reduction in subscription services
  - Better usage statistics
  - New ways to establish usage/access agreements
  - Personalization

The authentication interface – 1

- The PAPI AS is a general framework able to code assertions and send them to PoAs
  - It offers an open interface for modules that:
    - Establish user identity
    - Retrieve user attributes
  - There are modules supporting LDAP, SQL and ad-hoc databases, IMAP, POP3, X.509 and Kerberos
    - Specific modules have also been developed
- Different assertions can be sent to each PoA
  - Assertion templates are used to define the values and attributes sent
  - The assertion template to be used for a given PoA can be specified up to the indivisual user level

The authentication interface – 2

- Assertions are sent as part of an URL requesting an specific HTML object inside an HTML page
  - The PoA sends a different object depending on whether the request is authorized or not
  - Any HTML element with an *src* attribute
    - Images, frames, scripts, CSS stylesheets,...
- There are practical limits in URL size
  - Ad-hoc syntaxes, agreed between AS and PoAs
    - URNs can be used to formalize them
  - Assertion content is a reference to attributes
    - Direct: SAML queries from PoA to AS
    - Indirect: LDAP reference to attribute certificate

Seamless access – Deep linking

- A common term in the digital library jargon
  - Implies the ability of following any URL to an information retrieval system
  - From a user perspective, one of the major drawbacks of AA systems in comparison to simple IP-based access control
- A PAPI PoA incorporates
  - Elements to keep state among redirections caused by AA interactions
    - Including a POST method handler
  - Interfaces to seamless authorize users:
    - The GPoA interface inside the local trust realm
    - The WAYF interface to select and query ASes

Seamless access – Other services

- Web-based services can be easily integrated within PAPI in a authorization stack
  - The PAPI access module sets authentication data once access tokens are read
  - Other modules can use their own procedures
  - WebCT and VRVS are examples of this
- A very common user request is one-step authentication for any service
  - Experiments in access to external services by means of browser helper applications based on Kerberos
  - Upon authorization, the PoA generates a Kerberos ticket and includes it in the access tokens

Authorization mechanisms – 1

- A PoA can receive an assertion from:
  - An AS contacting as result of user authentication
  - A parent GPoA in response to a attribute query
  - An AS in response to a attribute query
- PoAs can make their authorization decisions according to:
  - Local *filters*
    - ❐ Regular expressions applied to attribute strings contained in assertions
    - ❐ Formalizable to some extent by means of URNs
  - A query to an external authorization engine
    - ❐ Transform assertion data into a format accepted by the engine

Authorization mechanisms – 2

- External authorization engines allow:
  - Richer semantics in authorization decisions
    - ❒ Finer control
  - The application of policies
    - ❒ Simplify and rationalize administration
  - Better formalization of the process
    - ❒ Set the ground for a full PMI
- Current implementation
  - Based on the native SPOCP protocol
  - A SAML interface is under development
    - ❒ Openess
    - ❒ Extensibility

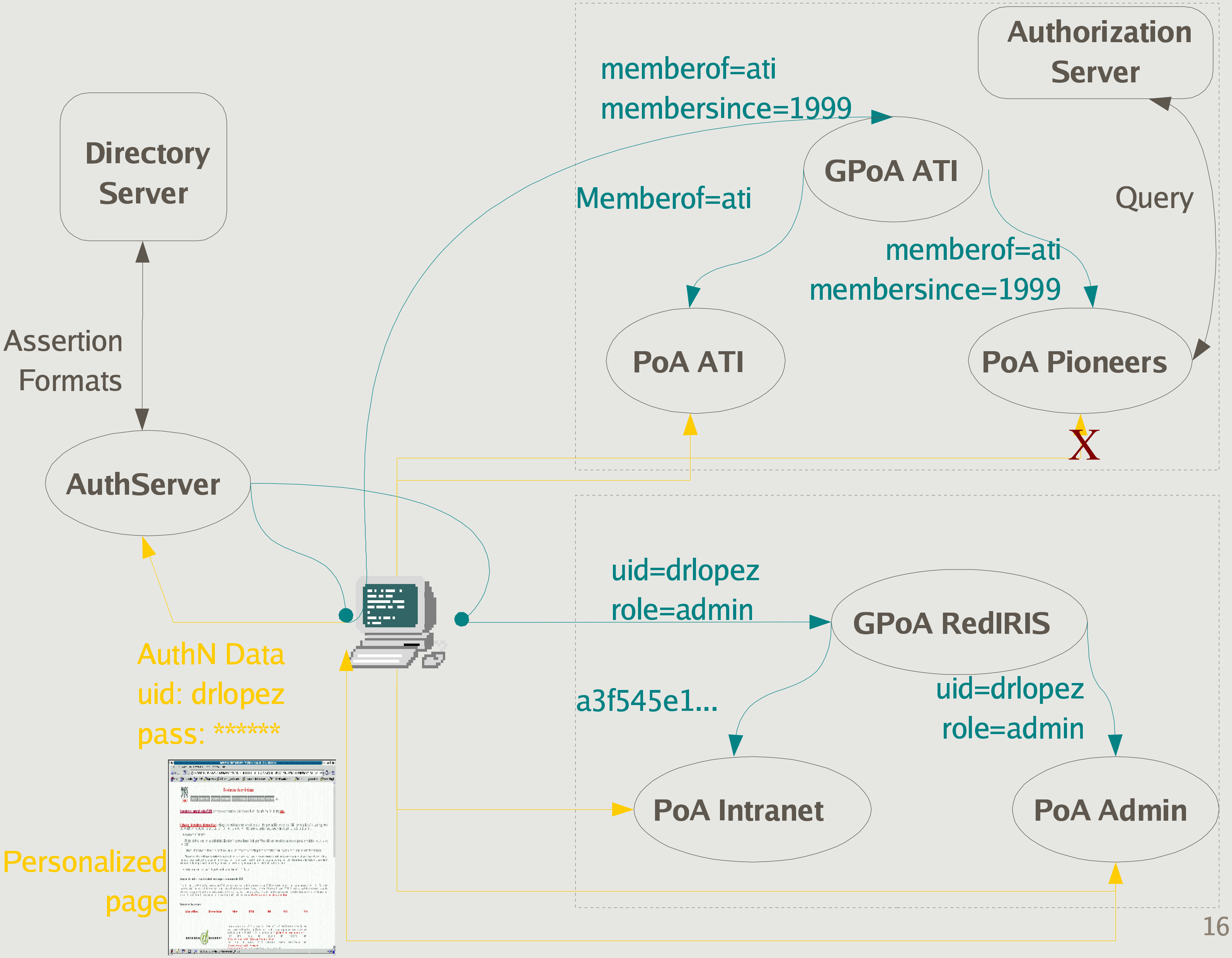
Access Tokens

- PAPI access tokens are implemented as HTTP cookies
  - Encrypted with the PoA symmetric key
- Structure
  - User data
    - ❐ Derived from received assertion(s)
    - ❐ From a Kerberos ticket to a simple nonce
  - Nonce
    - ❐ Stored in a database at the PoA
  - Time to live of the set of cookies
  - Time to live of the authorization
  - Optionally, client IP address

Access Token Rotation

- Cookie based tokens are subject to copy or interception attacks
  - IP address inclusion only mitigates this problem
    - And breaks full mobility
- PAPI PoAs incorporate a token rotation mechanism
  - Token nonces are periodically refreshed
  - Only the token containing the active nonce is valid
- The rotation procedures can be tuned
  - Permit or not persistent authorization
  - Avoid false positives caused by user behavior

Attribute-based authorization



Authorization Server

memberof=ati
membersince=1999

GPoA ATI

Directory Server

Memberof=ati

Query

memberof=ati
membersince=1999

PoA ATI

PoA Pioneers

Assertion Formats

X

AuthServer

uid=drlopez
role=admin

GPoA RedIRIS

AuthN Data
uid: drlopez
pass: ******

a3f545e1...

uid=drlopez
role=admin

PoA Intranet

PoA Admin

Personalized page

Current status

- Version 1.3 in production
  - Available in open source from http://www.rediris.es/app/papi/
  - Several thousands of users
- PAPI2 is under development
  - Redesign of components and protocols
  - Based on Web Services
  - Incorporate PKI usage
- Interoperability with other systems
  - First experiments with Athens DA
  - Coding a Shibboleth origin
  - Planned integration with OGSA