Quadrāsis™

Federation of Security Policy across Authorization Domains

Fred Dushin <*fadushin@hi.com*>
Quadrāsis, a business unit of Hitachi Computer Products (America), Inc.

March 29, 2001

Overview

- Problem Statement
- Privilege Attribute Mapping (PAM)
  - Standards Support
  - Administration
- Challenges
  - Delegation
  - Enforcement

Today's Organizations...

- ...strive towards centralized management of user and policy stores
- ...require autonomy in their design of security policy
- ...require scalability of security policy through use of roles, groups, privileges, etc.
- ...may require both a "push" and a "pull" model for privilege propagation
- ...but may have distributed applications deployed across organizational boundaries (B2B commerce, M&A, collaboration, etc.)

Problem Statement

- CSIv2 specification provides clear definition of a wire protocol and token formats for exchange of authorization data (e.g., X.509 Attribute Certificates, etc.)
- No guidance with respect to *meaning* of privileges issued in different authorization domains

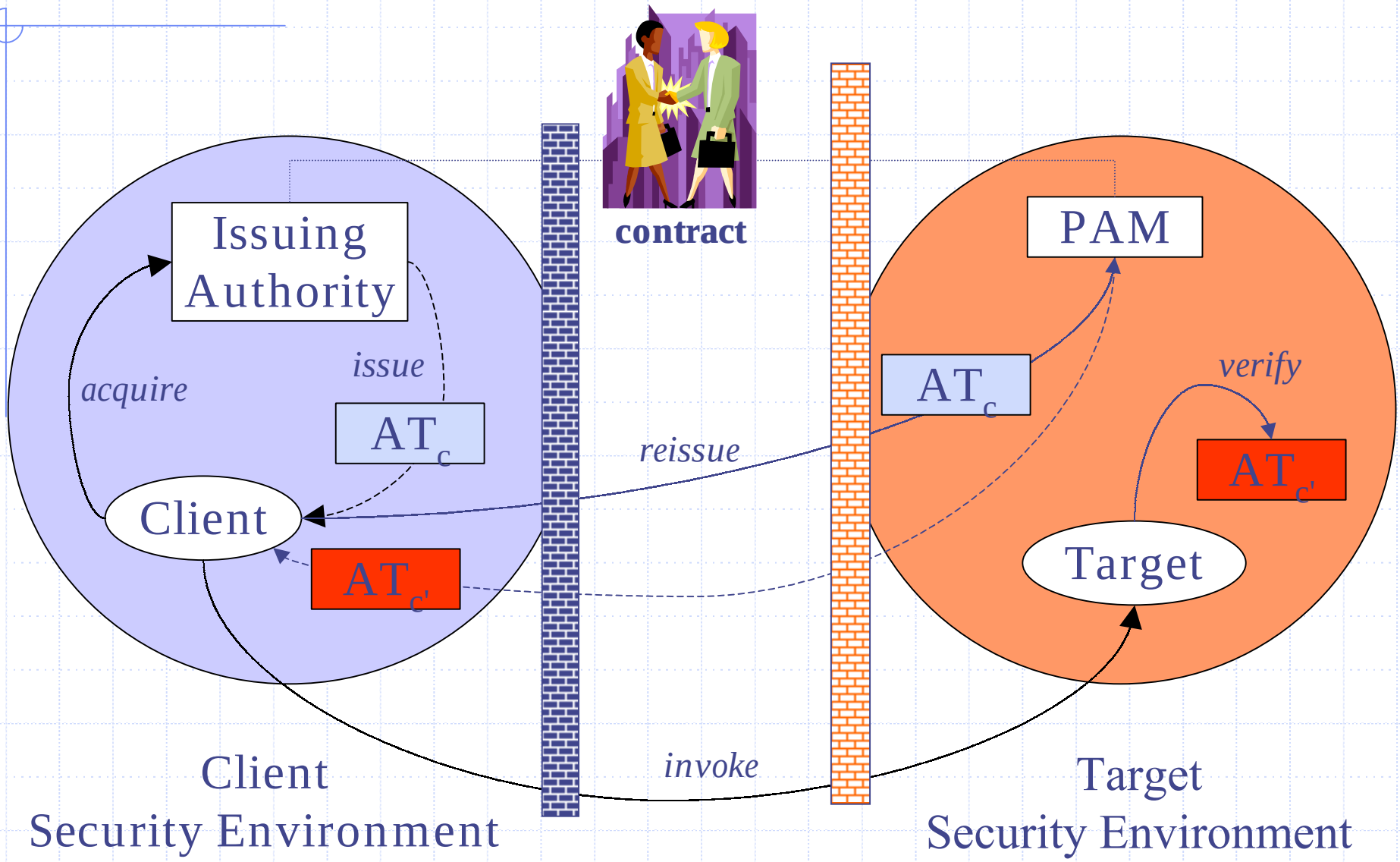
Constraints for a Solution

- No cross-domain administration
- Minimize unnecessary Communication
  - Firewalls are a *real problem*
- Any solution should scale well
- Administration should be
  - ...as simple as possible to maintain
  - ...lend itself well to analysis
- Interoperability would be a fortunate side-effect, not necessarily a goal

Privilege Attribute Mapping (PAM)

- Maps privileges granted in the client's authorization domain to privileges in the target authorization domain
- Mapping can be be made from
  - Client Security Service – prior to pushing privileges (ATLAS scenario)
  - Target Security Service – after secure association establishment, but before authorization decision
- Can be used as an authorization technology bridge

PAM Scenario (Client-side)



PAM Scenario (Target-side)

contract

Issuing Authority

issue

acquire

AT_c

Client

PAM

reissue

AT_c

$AT_{c'}$

Target

invoke

Client Security Environment

Target Security Environment

PAM interface

```
// IDL
interface PAM {

        CSI::AuthorizationElement
        reissue_token(
                in CSI::AuthorizationElement auth_token )
        raises
                UnknownIssuer,
                UnsupportedElementType,
                InvalidToken;

        typedef sequence<CSI::AuthorizationElement>
                IssuerList;

        IssuerList
        get_trusted_issuers();
};
```

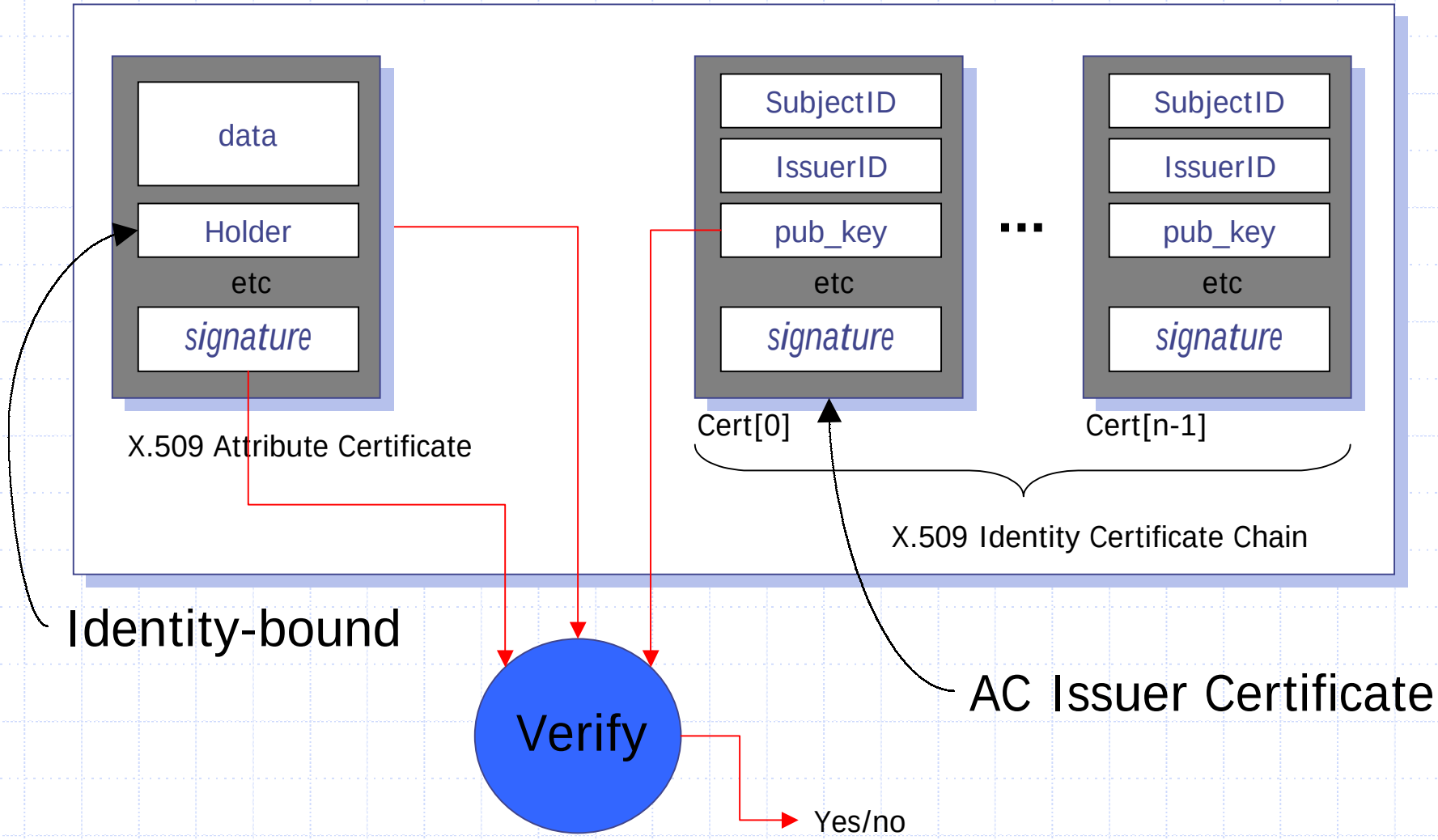
Standards Support

- CSIv2
  - Offers application-level protocol for pushing client credentials through GIOP ServiceContext
  - Neutral with respect to authorization technology (though clear intention is for use of X.509 AttributeCertificates)
  - Support for stateful and stateless connections

Standards Support

- X.509 Attribute Certificates
  - Provide verifiable privilege attributes of an invoking identity
  - More transient that X.509 Identity Certificates
  - Issuing Authority that grants privileges are encapsulated in a X.509 Identity Certificate Chain
  - Support a binding mechanism between the holder of the certificate and the authenticated identity (via a naming mechanism or via a digest of an identity certificate)

Attribute Certificate Chain



PAM Administration

Privilege Attribute Mapping represented by a table:

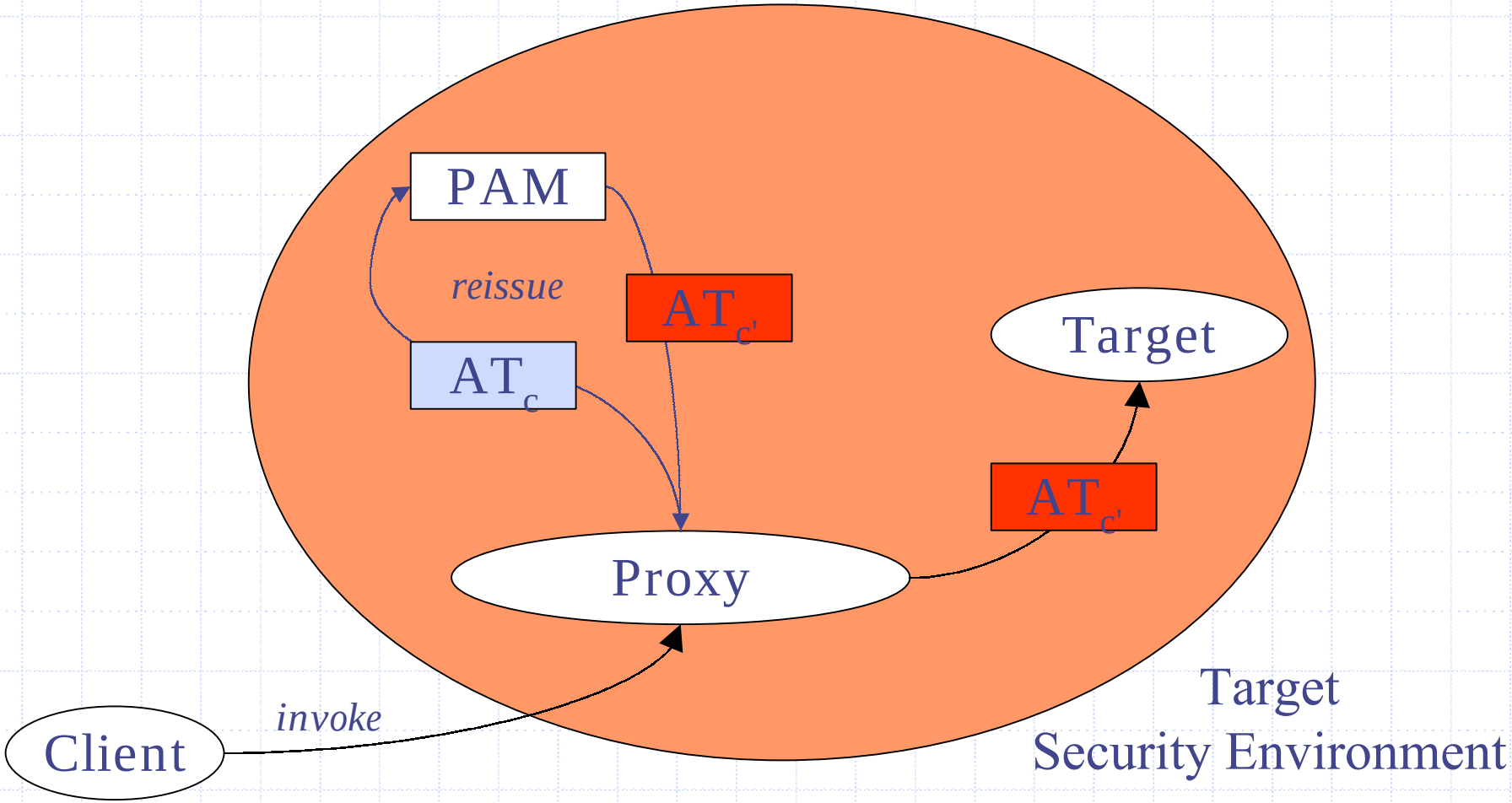
PAM for Boston offices of Acme.com

Client Domain	**Privilege**	**grant**
London Office	HR,VP	HR
	HR	guest_HR
California Office	Eng,projX	projX
	IT	IT
Any_trusted		intranet
Any		web

PAM Administration (cont'd)

- Allows use of wildcard rules to allow policy for default scenarios
  - **Any_trusted** Client comes from any trusted authorization domain not enumerated in table
  - **Any** Client comes from an untrusted authorization domain
- Negation poses interesting challenges
  - Not always feasible to grant a privilege based on *absence* of an attribute
- PAM table constitutes the contractual agreement between authorization domains

Delegation Scenario



Delegation

- If a PAM might receive an AT with embedded proxy attributes
  - These proxy attributes enumerate the entities the *client* endorses to act on the client's behalf
- PAM maps embedded proxies to entities in the target's domain *the client trusts*
  - From an administrative point of view, making this mapping scalable is difficult

PAM Challenges

- Amount of trust in client organization's issuing authority may be prohibitive
  - I.e., no enforcement mechanisms to assure Target Security Service that client issuing authority correctly issued authorization tokens
- Scalability (as number of contractual agreements grows)
- Still some gaps in the spec
  - Semantics of authorization tokens (ordering, duplication, etc.)

Conclusions

- A PAM provides a mechanism for implementing federation
- CSIv2 and X.509 Attribute Certificates provide a framework
- Administration is flexible, while still manageable
- Not without challenges
  - Authorization-token based delegation
  - Scalability
  - Enforcement

References

- CSIv2 Draft Adopted Specification
  - ptc/01-01-05
- ATLAS: The Authorization Token Layer Acquisition Service RFP
  - *orbos/00-12-17*
- An Internet Attribute Certificate Profile for Authorization
  - draft-ietf-pkix-ac509prof-06.txt
  - http://www.ietf.org

Appendix - CSIv2

```
// IDL
module CSI {
    typedef unsigned long AuthorizationElementType;

    struct AuthorizationElement {
        AuthorizationElementType      the_type;
        sequence<octet>               the_element;
    };

    typedef sequence<AuthorizationElement> AuthorizationToken;

    const AuthorizationElementType X509AttributeCertChain = 1;
};
```

Appendix - X.509 Attribute Certificates

```
AttributeCertificate := SEQUENCE {
        acinfo                  AttributeCertificateInfo,
        signatureAlgorithm      AlgorithmIdentifier,
        signatureValue          BIT STRING
}

AttributeCertificateInfo := SEQUENCE {
        version                 AttCertVersion DEFAULT v1,
        holder                  Holder,
        issuer                  AttCertIssuer,
        signature               AlgorithmIdentifier,
        serialNumber            CertificateSerialNumber,
        attrCertValidityPeriod AttCertValidityPeriod,
        attribute               SEQUENCE of Attribute,
        issuerUniqueID          UniqueIdentifier OPTIONAL,
        extensions              Extensions OPTIONAL
}
```