A Distributed Credential Management System for SPKI-based Delegation Scenarios

Óscar Cánovas
<ocanovas@um.es>

University of Murcia (Spain)

1st Annual PKI Research Workshop
Overview

- Introduction
- Motivation
- Naming Management System
- Authorization Management System
- Some implementation details
- Conclusions and future work
Introduction

✓ Questions about identity have been (partially) solved by the X.509 standard, but we have to determine what the identities should be allowed to do

✓ Digital certificates can also contain information about authorization (in fact, about anything)
  ✓ Examples: SPKI/SDSI, X.509 Attribute Certificates, KeyNote

✓ There are several proposals making use of SPKI certificates to provide authorization services to different environments: CORBA, WWW, physical access control...
Most of those SPKI scenarios are based on delegation
- Resource controllers have small ACLs delegating access to some particular public keys (authorities)

Application-dependent approaches try to answer:
- How do I encode a certification request?
- How do I submit the certification request?
- How do the authorities specify and enforce the authorization policies? (i.e. who is able to obtain a particular authorization?)

In complex environments, simple command-line (and off-line) applications do not seem to be the right approach
Motivation (II)

- It is necessary to address the problems related to scalability and interoperability.

- **DCMS (Distributed Credential Management System)**
  - DCMS defines: requests, policies, and entities
  - DCMS is divided into:
    - NMS: SPKI ID Certificates
    - AMS: SPKI Attribute and Authorization Certificates
  - Entities exchange authorization information using the AMBAR Protocol (similar protocols are valid too)
  - Main goal of DCMS: to be application-independent
A particular scenario

- Physical access control based on RBAC and SPKI
- We use special devices named TICA
  - Located at the entrances of buildings
  - They can establish their access control options
  - Users make use of their smart cards
- TICAs delegate authorization management to particular authorization authorities by means of:
  - ACLs entries (propagation activated)
  - Authorization certificates (propagation activated)
A particular scenario (II)

- Delegation structure
  - TICAs create authorization certificates
  - AAs create attribute certificates (relationship Role-Permission)
  - NAs create role membership certificates (relationship User-Role)

- Authorization loop
A particular scenario (III)

Use of DCMS:
- Principals can request SPKI certificates to gain access
- Trusted service access points (SAP)
- DCMS provides an encoding for certification requests and authorization policies
- Requests and certificates are exchanged using AMBAR
Naming Management System (NMS)

- NMS is responsible for certification operations related to SPKI ID certificates

- This type of certificates can be used to:
  - link a name to a particular public key (principal)
  - define group membership

- NMS can be especially useful when authorization is based on groups of principals
  - NMS can be used by the principals in order to obtain an ID certificate for group $G$

- ID certificates are issued by naming authorities (NA)
NMS Entities

- Requestors:
  - They create certification requests
  - Additional certificates can be also attached to the requests

- Two types of requestors:
  - Demanding an ID certificate for a public key
  - Demanding an ID certificate for a name (subgroups)
NMS Entities

✓ Service Access Points (SAP):
  ✓ Requestors use SAPs to submit the certification requests
  ✓ Several advantages:
    ➢ Naming authorities can be protected
    ➢ They “know” the appropriate naming authorities
    ➢ Public terminals placed at buildings or departments
NMS Entities

✓ Naming Authorities (NA):
  ✓ NAs are controlled by authorization policies
  ✓ In DCMS, those policies are implemented using SPKI ACLs
  ✓ Use of certificate chain discovery methods
    - Input: request, additional certificates, ACL
    - Output: data used to generate the new certificate
NMS requests and ACL entries

NMS s-expressions:

- There is no need for a new syntax (we use the certificate struct.)
- Main differences:
  - N can be a (* prefix) form or a (* set) form
  - P can make reference to several principals (* set Q S T)
  - valid is making reference to the intended validity period
  - The request is signed by the requestor, not by the issuer
NMS requests and ACL entries

ACL entries (authorization policy):
- The tag specifies which entities can obtain ID certificates
- R makes reference to the valid certificate requestors
  - Requestors can be relying parties different from certificate subjects
- certificates can be requested during the period specified by valid

\[ acl \\
  \begin{array}{l}
  \text{(entry)} \\
  \quad \text{(subject } R) \\
  \quad \text{(propagate)} \\
  \quad \text{(tag ...)} \\
  \quad \text{(valid ...)} \\
  \end{array} \]
A Distributed Credential Management System for SPKI-based Delegation Scenarios

Oscar Canovas
PKI RW 2002

NMS example

(sequence
  (tag
    (cert-request
      (issuer
        (name morpheus-pk Nebuchadnezzar))
      (subject neo-pk)
    )
  )
  (signature ...)
)

Request and additional certificate

(cert
  (issuer
    (name morpheus-pk Nebuchadnezzar))
  (subject trinity-pk)
)

ACL

(acl
  (entry
    (subject
      (name morpheus-pk Nebuchadnezzar))
    (tag
      (cert-request
        (issuer
          (name morpheus-pk Nebuchadnezzar))
        (subject
          (* set neo-pk trinity-pk switch-pk))
      )
    )
  )
)

ACL
Authorization Management System

✓ AMS is responsible for the certification operations related to SPKI Attribute and Authorization certificates

✓ NMS and AMS are based on similar entities:
  ✓ Requestors and SAPs are also part of AMS
  ✓ NAs are replaced by AAs (Authorization Authorities)

✓ S-expressions for requests and ACLs are similar to those defined for NMS (including propagation and tags)

✓ There are also two types of requestors:
  ✓ Requestors of authorization certificates
  ✓ Requestors of attribute certificates
AMS. Attribute Certificates

✓ Attribute Certificates: Authorization → Name
  ✓ The name might make reference to a group name

✓ “Who must the requestor of an attribute certificate be?”
  ✓ It depends on the authorization policies (no inherent policies)

✓ Valid requestors range from group members to specific role managers

✓ Group management can be greatly simplified using specific administrators (role managers)
AMS. Role Managers

- We need to encode statements like:
  - Psion-AA authorizes the Role Manager RM to request attribute certificates granting the set of permissions \texttt{tag-A} for group \texttt{Nebuchadnezzar} defined by Morpheus

```
(acl
  (entry
    (subject RM-pk)
    (tag
      (cert-request
        (issuer psion-pk)
        (subject (name morpheus-pk Nebuchadnezzar))
        (tag tag-A)
      )
    )
  )
)
```
A Distributed Credential Management System for SPKI-based Delegation Scenarios

AMS. Role Managers

Request

ACL
DCMS. Implementation

- AMBAR was implemented using Intel CDSA 3.14

- DCMS is being implemented also using CDSA
  - Graphical User Interface (QT libraries)
  - Red Hat Linux 7.1

- Several applications:
  - DCMS tag constructors
  - ACL Management
  - Authorities
  - Service Access Points
A Distributed Credential Management System for SPKI-based Delegation Scenarios

DCMS. Implementation
Conclusions and Future Work

- DCMS provides:
  - Certification requests (s-expressions)
  - Authorization policies (SPKI ACLs)
  - Architectural elements

- Mechanisms for scalability:
  - We keep inherent policies to minimum
  - It is possible to specify sets of certificates (issued on demand)
  - We make a clear distinction between requestors and subjects

- Future work:
  - Certificate Storage and Certificate Revocation