Distributed Certificate-Chain Discovery

Stefan Schwoon  Hao Wang  Somesh Jha  Thomas Reps

Universität Stuttgart  University of Wisconsin-Madison
Authorization Problem

For a given security policy $P$ with respect to a resource $R$, can principal $A$ access $R$?

- Straightforward in a centralized environment
- But real-world is not centralized
  - Resources/services are located in different administrative domains
  - No centralized authority—policies cross domains!
  - Privacy concerns—users may not want to reveal too much information
Cross-Domain Authorization

Q1: Should Alice be allowed to access R in domain UW?  
Q2: If so, prove it!

Centralized Solution

- Assume a centralized authority
- Does not deal with privacy concerns
Solution: Distributed Certificate-Chain Discovery

- Based on two technologies
  - SPKI/SDSI—a trust-management language
  - WPDS—Weighted Pushdown Systems
- Employs a **distributed algorithm** to find certificate chains
  - Previous approaches use centralized algorithms
    - SPKI/SDSI, RT₀, etc.
- Addresses privacy issue—does not reveal sensitive information
- Scalable
  - Tested in a simulated environment with up to 1,600 certificates
Why Use Weighted Pushdown Systems?

- WPDS technology enables a **distributed** solution for the authorization problem
  - WPDS reachability algorithm uses an automaton to summarize knowledge \(\Rightarrow\) synopsis of SPKI/SDSI proof
  - To send a relevant proof fragment, ship an automaton fragment
- Addresses shortcomings of previous SPKI/SDSI work
  - A proof may consist of **multiple** certificate chains
  - Original approach of Rivest et al. only capable of finding **single-chain** proofs
  - Addresses privacy concerns
Status

- A prototype has been built and tested
  - Uses a SPKI/SDSI library to manage certificates
  - Uses the WPDS Library to perform proof search
  - Distributed algorithm coordinates interactions between multiple domains
DoD Interests

SBIR: AF03-095:
Cross-domain user identity and credential management
- Maintain organizational namespace consistency
- Enable information-system managers to effectively deal with the rapid consolidation and turnover of personnel within mission critical force package

SBIR: AF04-094:
XML Guard
- Investigate cross-domain guarding advancement opportunities made possible by the rapid growth of XML technologies

SBIR: N05-085:
Cross-Domain Document-Based Collaboration
- Develop technologies that enable secure cross-domain collaboration technologies
  - Secure and certifiable sharing and editing of composite documents containing sensitive information
  - Span multiple security levels
Outline

- Introduction
- SPKI/SDSI Background
- Distributed Certificate-Chain Discovery Using WPDS
# Cross-Domain Authorization

<table>
<thead>
<tr>
<th>Issues</th>
<th>Existing Approaches: SPKI/SDSI, $RT_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Policy Management</strong></td>
<td>How to manage certificates when there are multiple administrative domains?</td>
</tr>
<tr>
<td><strong>Policy Enforcement</strong></td>
<td>How to prove that one is allowed to access a resource?</td>
</tr>
</tbody>
</table>

Requires all certificates to be sent to a single site.
Our Focus: SPKI/SDSI

- Simple Public Key Infrastructure (SPKI)/Simple Distributed Security Infrastructure (SDSI)
  - A trust-management system that addresses cross-domain authorization
- Two components:
  - Principals
    - Resource owners, users, databases, etc.
    - Represented by their public keys, e.g. $K_{NSF}, K_{ONR}, K_{CS}$
  - Certificates
    - Security policy = set of certificates
    - No need for a centralized authority!
      - Any principal can issue a certificate
      - Each certificate specified and signed by the issuing principal
SPKI/SDSI Name Certificates

- Format: (Key, Name, Subject, Validity)
  - Meaning: Subject is a member of the group known (to Key) as “Name”
  - For convenience: Key Name → Subject

- Map public keys to meaningful (local) names
  - Alice is a faculty member in CS: $K_{CS\text{ faculty}} \rightarrow K_{Alice}$
  - Bob is one of Alice’s students: $K_{Alice\text{ student}} \rightarrow K_{Bob}$

- Declares membership relation across domains
  - $K_{Alice\text{ friend}} \rightarrow K_{Charlie\text{ enemy}}$
  - $K_{UW\text{ faculty}} \rightarrow K_{CS\text{ faculty}}$
SPKI/SDSI Authorization Certificates

- **Format:** (Key, Subject, Delegation, Tag, Validity)
  - **Meaning:** Key grants right “Tag” to Subject
  - **For convenience:** Key → Subject Delegation

- **Grants access permission to other principals**
  - e.g. Bob can read Prof. Alice’s homework directory:
    - Directly:
      \[ K_{Alice} \rightarrow HW \rightarrow K_{Bob} \]
    - Indirectly — via 1 or more name certificates:
      \[ K_{Alice} \rightarrow students \rightarrow K_{Bob} \]

- **May delegate rights to other principals**
  \[ K_{NSF} \rightarrow K_{EDU} programs \]
Certificate-Chain

- An authorization proof is a chain of certificates

\[ K_{UW(R)} \xrightarrow{K_{UW}} K_{LS} \xrightarrow{R} K_{CS} \xrightarrow{K_{CS}} K_{Alice} \]
Algorithms for Certificate-Chain Discovery

- Previous certificate-chain-discovery algorithms require all certificates to be sent to a single site
  - Defeats the purpose of having cross-domain security policies
  - No privacy! Each site must reveal its certificates

- This work
  - Distributed algorithm for certificate-chain discovery
Outline

- Introduction
- SPKI/SDSI background
- Distributed Certificate-Chain Discovery Using WPDS
Distributed Certificate-Chain Discovery—How?

- Exploit relationships among certificates
  - Who has related certificates?

- Map SPKI/SDSI certificate-chain problem to Weighted Pushdown System (WPDS) domain
  - Ship automaton fragments to different sites
  - Different sites collaborate on proof
Exploit Certificate Relationships

Cross-site certificates
Weighted Pushdown System (WPDS)

- Pushdown System (PDS), plus
  - Weights on transition rules
- Three components
  - States: \{\sigma_1, \sigma_2, \sigma_3\}
  - Stack symbols: \{A, B, C, D\}
  - Transition rules with weights:
    - \(<\sigma_1, A> \xrightarrow{w_1} <\sigma_2, \varepsilon>\)
    - \(<\sigma_1, A> \xrightarrow{w_2} <\sigma_2, B>\)
Map SPKI/SDSI to WPDS

SPKI/SDSI Certificates

$K_{UW} \xrightarrow{R} K_{LS}$ faculty

SPKI/SDSI Certificate Chain

$K_{UW} \xrightarrow{R} K_{LS}$ faculty

$K_{LS}$ faculty $\rightarrow K_{CS}$ faculty

$K_{CS}$ faculty $\rightarrow K_{Alice}$

WPDS Transition Rules

$\langle K_{UW}, \_ \rangle \xrightarrow{R} \langle K_{LS}, \text{faculty}, \_ \rangle$

WPDS Run

R

Rule 1

K_{LS}

K_{CS}

Rule 2

K_{Alice}

Rule 3

Automaton State

Goal

Initial Automaton

10/27/2005

Hao Wang (hbwang@cs.wisc.edu)
Distributed Certificate-Chain Discovery Using WPDS
Distributed Certificate-Chain Discovery Using WPDS

- Two approaches, derived from the Generalized Pushdown Reachability (GPR) problems in WPDS:
  - Generalized Pushdown Successor (GPS)
    - Distributed Post*
  - Generalized Pushdown Predecessor (GPP)
    - Distributed Pre*
R is accessible to faculty members in the college of LS.

Faculty members of Bio and CS are faculty members of LS.

Alice is a faculty member in CS.
Distributed Post*

Start search (Req. ID)

Register (Req. ID)
Distributed Post*

\[ \begin{align*} 
\langle K_{UW}, □ \rangle & \rightarrow \langle K_{LS}, \text{faculty}, □ \rangle 
\end{align*} \]
Distributed Post*
Distributed Post*

UW

LS

BIO

CS

\( K_{UW} \)

\( K_{LS} \)

\( K_{BIO} \)

\( K_{CS}, K_{Alice} \)

Alice, Req. ID

R

ε

faculty

faculty

faculty

faculty

\( K_{LS} \)

\( K_{CS} \)

\( K_{BIO} \)

 Req. ID

 Req. ID

 Req. ID

 Alice, Req. ID
Distributed Post*
Preserving Privacy

Only knows req. ID
Only knows who the client is
Does not know who request it
Does not know what client is accessing
Multiple Certificate Chains

- In real world, a proof may consist of **multiple** certificate chains
  - Previous work assumes one certificate chain
- Our approach addresses this issue
  - WPDS enables us to solve the problem—using semirings
WPDS and Multiple Certificate Chains

Alice requests for \{read, write\}

WPDS Semiring Combine operation

Alice

\[ \oplus = \cup \]
Future Work

- **Performance enhancement**
  - Use caching to reduce response time
    - Especially for long certificate chains
  - Network optimization—piggyback messages

- **Termination**
  - How to determine whether all possible paths have been exploited and terminate the search early?
END

Questions and comments?