Shared resources in RELOAD as a primitive for coordinating group communication

draft-knauf-p2psip-share-00

Alexander Knauf
Gabriel Hege
Thomas Schmidt
Matthias Wählisch

alexander.knauf@haw-hamburg.de, hege@fhtw-berlin.de, {t.schmidt,waehlisch}@ieee.org

Outline

1. Problem Statement and Objectives

2. Overview Shared Resources

3. Access Control

4. Application Scenarios

5. Conclusion & Outlook

Problem Statement

Why do we need Shared Resources in RELOAD?

- Standard access control mechanisms are not sufficient for controlled write access by multiple peers
- Simplest way: USER-MATCH policy and certificate with same user name for all peers
  - Need to contact enrollment server → infeasible
  - Need to distribute private key/secrets/certificate
  - No individual revocation
- Use cases:
  - conference registration, message board, SSM source announcement, ...

Objectives

- Single resource to be writable by a well defined group of peers
  - Without contacting enrollment server
  - Allow revocation
- Optionally: more relaxed resource naming scheme
- Define some primitives for other Usages to build upon

Shared Resources - Overview

- RELOAD Resource (Kind) for which multiple peers have write access
- Resource Owner: has access by some (standard) policy (e.g., USER-MATCH)
- Resource Owner grants access using an Access Control List (ACL)
- ACL is stored under the same Resource-ID

  → on the same peer
- Write permission may be further delegated

  → Chain of delegations in ACL

Access Control Policies

- For the Owner:
  - Standard policy (e.g., USER-MATCH)
    - or relaxation thereof: USER-PATTERN-MATCH
  - Allows the Owner to store the ACL
- For other peers:
  - USER-CHAIN-ACL
- Enforced by the storing peer, but independently verifiable

Access Control List

- Stored under the same Resource Name as the Shared Resource
- Contains delegations from_user → to_user
- Users in the ACL may write the Shared Resource
- Chain of signed delegations may be independently verified

```
struct {
  opaque resource_name<0..2^16-1>;
  KindId kind;
  opaque from_user<0..2^16-1>;
  opaque to_user<0..2^16-1>;
  Boolean allow_delegation;
} AccessListData;
```

Revocation of Write Permission

Revocation is simple:

- Invalidate corresponding delegation in ACL
  - set exists=false
- Succeeding delegations also invalidated
- Owner can revoke the whole list by deleting the root entry

Access Control List – Example

<table>
<tr><th colspan="3">Access List</th></tr>
<tr><th>#</th><th>Array Entries</th><th>Signature</th></tr>
<tr><td>0</td><td>Kind:1234 from:Owner -> to:Owner ad:1</td><td>signed by Owner</td></tr>
<tr><td>1</td><td>Kind:1234 from:Owner -> to:Alice ad:1</td><td>signed by Owner</td></tr>
<tr><td>2</td><td>Kind:1234 from:Alice -> to:Bob ad:0</td><td>signed by Alice</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>42</td><td>Kind:4321 from:Owner -> to:Owner ad:1</td><td>signed by Owner</td></tr>
<tr><td>43</td><td>Kind:4321 from:Owner -> to:Carol ad:0</td><td>signed by Owner</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
</table>

Application Scenarios (1): Distributed Conferencing (DisCo)

- The impulse for developing ShaRe
- A distributed conferencing Usage for RELOAD
- Tightly coupled SIP conference
- Focus functionality is transparently distributed among multiple peers, which act as a single focus instance
- All focus peers of a conference register under a single URI
- The conference initiator grants the focuses write access to the conference registration

`draft-knauf-p2psip-disco`

Application Scenarios (2): AMT-Relay Registration

- Usually AMT-Relays are discovered via anycast
  - Without anycast other means are necessary
- A Shared Resource in a RELOAD overlay could be used:
  - AMT-Relays register themselves at a well known location
  - AMT-Gateways lookup Relays and choose the closest one
- Multi hop tunnels allow traffic aggregation:
  - Possibly optimization of tunnel trees using information from RELOAD

Application Scenarios (3): Discovery of Tunnel Endpoints

- When no AMT-functionality is available (mobile) clients may need to establish tunnels
- Tunnel endpoints register themselves in a RELOAD overlay in a Shared Resource
- Applications with a built-in RELOAD stack can use this to discover an (optimal) endpoint

Application Scenarios (4):
SSM Source Announcement

- Problem in SSM: finding out which sources are available
- Common solutions: broadcast announcement (e.g., Bayeux) or
  out of band communication
- ShaRe can be used to announce available sources for a group
  - E.g., Stored under a Resource ID derived from the group's address
- The group creator initially registers the resource and delegates write permission to permitted sources
- The group can be extended as long as one source with delegation permission is active

Application Scenarios (5): Distributed Tracker

- Similar to Distributed Conferencing
- But instead of focus peers, instances of a distributed tracker register themselves in a Shared Resource

Conclusion & Outlook

- Defined primitives to allow coordinated shared writing of a RELOAD resource
- Can be used for service announcement in moderately sized groups
- Now we need some drafts using these primitives ;-) (see `draft-knauf-p2psip-disco-02`)

Thank you for your attention!

Any Questions?