# An Architecture for a Secure Service Discovery Service

Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, Randy H. Katz

University of California, Berkeley

{*czerwin, ravenben, hodes, adj, randy*}*@cs.berkeley.edu*

April 27, 2014

# Overview

1. Introduction
    - Motivation
    - Goals
2. Design Concepts
3. Architecture
4. Wide Area Support

# Outline

1. Introduction
   - Motivation
   - Goals

# Motivation

- Large scale deployment of networks and devices
- Cheaper networks and network-enabled devices

# Goals

- Locate a service out of thousands
- Secure and trusted services with minimum client intervention
- Repository of (running) service descriptions
- Hierarchical load-balancing and recovery

# Outline

- Motivation
- Goals

2 Design Concepts

# Design Concepts

- Annoucement-based Information Dissemination
  Use of perodic multicast annoucements for recovery, bootstrapping and updating. Suitable for eventual consistency.
- Hierarchical Organisation
  If a server is overloaded a child node is started. Downwards cascading recovery for several server failures.
- XML Service Descriptions
  Flexibility, validation ability and backward compatibility.
- Privacy and Authentication
  Hybrid cryptography: symmetric and asymmetric cryptography. Principals and component's public keys assure authentication.

# Design Concepts

- Annoucement-based Information Dissemination
  Use of perodic multicast annoucements for recovery, bootstrapping and updating. Suitable for eventual consistency.
- Hierarchical Organisation
  If a server is overloaded a child node is started. Downwards cascading recovery for several server failures.
- XML Service Descriptions
  Flexibility, validation ability and backward compatibility.
- Privacy and Authentication
  Hybrid cryptography: symmetric and asymmetric cryptography. Principals and component's public keys assure authentication.

# Design Concepts

- Annoucement-based Information Dissemination
  Use of perodic multicast annoucements for recovery, bootstrapping and updating. Suitable for eventual consistency.
- Hierarchical Organisation
  If a server is overloaded a child node is started. Downwards cascading recovery for several server failures.
- XML Service Descriptions
  Flexibility, validation ability and backward compatibility.
- Privacy and Authentication
  Hybrid cryptography: symmetric and asymmetric cryptography.
  Principals and component's public keys assure authentication.

# Design Concepts

- Annoucement-based Information Dissemination
  Use of perodic multicast annoucements for recovery, bootstrapping and updating. Suitable for eventual consistency.
- Hierarchical Organisation
  If a server is overloaded a child node is started. Downwards cascading recovery for several server failures.
- XML Service Descriptions
  Flexibility, validation ability and backward compatibility.
- Privacy and Authentication
  Hybrid cryptography: symmetric and asymmetric cryptography. Principals and component's public keys assure authentication.

# Outline

- Motivation
- Goals

3 Architecture

# SDS Server

- Global multicasts authenticated messages
- Authenticated advertisements contain:
  - Certificate Authority and Capabilities Manager contact
  - Address for sending service announcements
  - Service annoucement rate
- Aggregate rate set by administrators
- Overloaded servers reaching a given threshold start another server
- Failure handled individually or cascading through the hierarchical organisation
- Privacy and authentication possible through the *secure one-way service broadcast*

# Services

1. Continously listen on the global multicast channel for SDS server announcements
2. Multicast its service descriptions to the appropriate channel/frequency
3. Set appropriate capabilities by contacting the Capabilities Manager

# Services

1. Continously listen on the global multicast channel for SDS server announcements
2. Multicast its service descriptions to the appropriate channel/frequency
3. Set appropriate capabilities by contacting the Capabilities Manager

# Services

1. Continously listen on the global multicast channel for SDS server announcements
2. Multicast its service descriptions to the appropriate channel/frequency
3. Set appropriate capabilities by contacting the Capabilities Manager

# Certificate Authority

1. Clients contact CAs for retrieving the principal's certificate
2. Stores encryption key certificates and the principal's certificates
3. The CA's public key is public
4. The encryption key certificate is used by the client to communicate with the principal

# Certificate Authority

1. Clients contact CAs for retrieving the principal's certificate
2. Stores encryption key certificates and the principal's certificates
3. The CA's public key is public
4. The encryption key certificate is used by the client to communicate with the principal

# Certificate Authority

1. Clients contact CAs for retrieving the principal's certificate
2. Stores encryption key certificates and the principal's certificates
3. The CA's public key is public
4. The encryption key certificate is used by the client to communicate with the principal

# Certificate Authority

1. Clients contact CAs for retrieving the principal's certificate
2. Stores encryption key certificates and the principal's certificates
3. The CA's public key is public
4. The encryption key certificate is used by the client to communicate with the principal

# Capabilities Manager

1. Contacted by services
2. Services specify an ACL for principals principals
3. Generates, stores and distributes appropriate capabilities

# Capabilities Manager

1. Contacted by services
2. Services specify an ACL for principals principals
3. Generates, stores and distributes appropriate capabilities

# Capabilities Manager

1. Contacted by services
2. Services specify an ACL for principals principals
3. Generates, stores and distributes appropriate capabilities

# Secure Communications

# Secure Communications

- **Authenticated Server Annoucements**

# Secure Communications

- **Authenticated Server Annoucements**
    - Readable by all clients

# Secure Communications

- **Authenticated Server Annoucements**
  - Readable by all clients
  - Non-forgeable

# Secure Communications

- **Authenticated Server Annoucements**
    - Readable by all clients
    - Non-forgeable
    - Reply attack resistant (timestamps)

# Secure Communications

- **Authenticated Server Annoucements**
  - Readable by all clients
  - Non-forgeable
  - Reply attack resistant (timestamps)
- **Secure One-Way Service Description Annoucements**
  Hybrid public/symmetric key system: a packet is sufficient for describing a service which will be decrypted by the SDS server

# Secure Communications

- **Authenticated Server Annoucements**
  - Readable by all clients
  - Non-forgeable
  - Reply attack resistant (timestamps)
- **Secure One-Way Service Description Annoucements**
  Hybrid public/symmetric key system: a packet is sufficient for describing a service which will be decrypted by the SDS server
- **Authenticated RMI**
  A handshake establishes the symmetric key for the session between client and SDS servers and between pairs of SDS servers

# Outline

- Motivation
- Goals

4 Wide Area Support

# Wide Area Support

# Wide Area Support

1. **Hierarchies built based upon query criteria**:

# Wide Area Support

1. **Hierarchies built based upon query criteria**:
    - Administrative domain

# Wide Area Support

1. **Hierarchies built based upon query criteria**:
    - Administrative domain
    - Network topology

# Wide Area Support

1. **Hierarchies built based upon query criteria**:
   - Administrative domain
   - Network topology
   - Physical location

# Wide Area Support

1. **Hierarchies built based upon query criteria**:
    - Administrative domain
    - Network topology
    - Physical location
2. **Aggregate service description (lossy)**

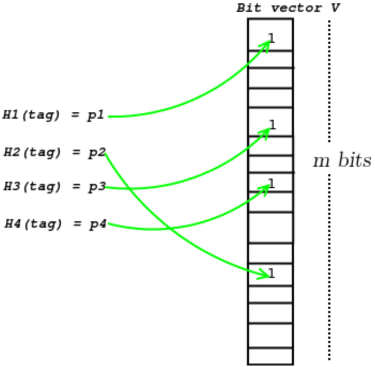
# Wide Area Support

1. **Hierarchies built based upon query criteria**:
   - Administrative domain
   - Network topology
   - Physical location
2. **Aggregate service description (lossy)**
3. **Use aggregation tables for routing queries**

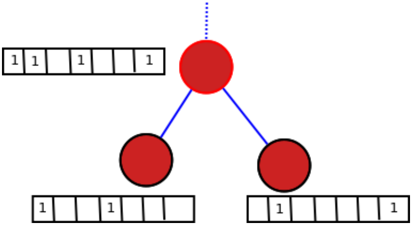
# Lossy aggregation & query routing

- Hash values of tag subsets of service descriptions used as summary
- Algorithm:
    1. When adding: compute description tag subset, insert into Bloom Filter table
    2. When querying: compute tag subsets, examine corresponding entries in Bloom Filter table for possible matches
- Limitations:
    - Computation required: fewer subset hashes
    - Space required: use bloom filters

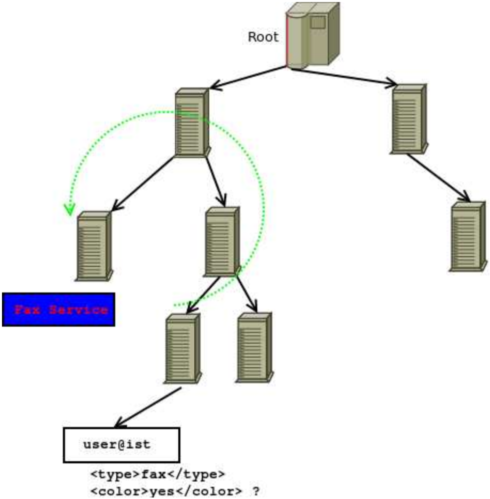
# Bloom filters



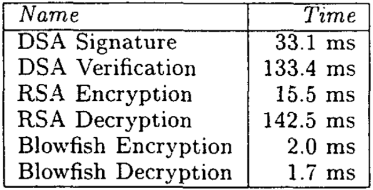
# Index aggregation & routing



# Index aggregation & routing



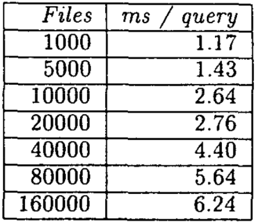
# System Performance



| *Name* | *Time* |
|---|---|
| DSA Signature | 33.1 ms |
| DSA Verification | 133.4 ms |
| RSA Encryption | 15.5 ms |
| RSA Decryption | 142.5 ms |
| Blowfish Encryption | 2.0 ms |
| Blowfish Decryption | 1.7 ms |

Table 1: Timings of cryptographic routines

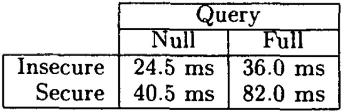
# System Performance



| *Files* | *ms / query* |
|---:|---:|
| 1000 | 1.17 |
| 5000 | 1.43 |
| 10000 | 2.64 |
| 20000 | 2.76 |
| 40000 | 4.40 |
| 80000 | 5.64 |
| 160000 | 6.24 |

Table 2: XSet Query Performance

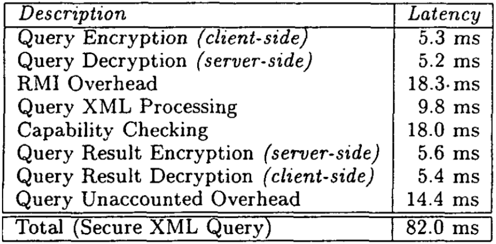
# System Performance



| | Query | |
|---|---|---|
| | Null | Full |
| Insecure | 24.5 ms | 36.0 ms |
| Secure | 40.5 ms | 82.0 ms |

Table 3: Query Latencies for Various Configurations

# System Performance



| *Description* | *Latency* |
|---|---|
| Query Encryption *(client-side)* | 5.3 ms |
| Query Decryption *(server-side)* | 5.2 ms |
| RMI Overhead | 18.3 ms |
| Query XML Processing | 9.8 ms |
| Capability Checking | 18.0 ms |
| Query Result Encryption *(server-side)* | 5.6 ms |
| Query Result Decryption *(client-side)* | 5.4 ms |
| Query Unaccounted Overhead | 14.4 ms |
| Total (Secure XML Query) | 82.0 ms |

Table 4: Secure Query Latency Breakdown

# Related Work

- **DNS & Globe**
- **Condor Classads**
- **JINI**
- **Service Location Protocol**

# Conclusion

Work still needed on:

- **Wide area implementation**
- **Benchmarking**
- **Ninja infrastructure necessary to evaluate**

# Questions ?