

# 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 30, 2014

# Overview

- 1 Introduction
  - Motivation
  - Goals
- 2 Design Concepts
- 3 Architecture
- 4 Wide Area Support
- 5 System Performance
- 6 Conclusion

## 1 Introduction

- Motivation
- Goals

# Motivation

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

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

- Motivation
- Goals

## 2 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.

- Announcement-based Information Dissemination  
Use of periodic multicast announcements 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.



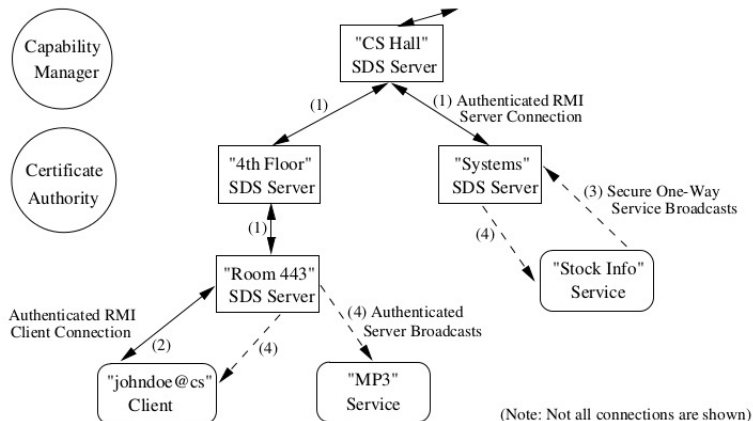
- Announcement-based Information Dissemination  
Use of periodic multicast announcements for recovery, bootstrapping and updating. Suitable for eventual consistency.
- Hierarchical Organisation  
If a server is overloaded a child node is started. Downwards cascading recovery for several server failures.
- 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.

- Announcement-based Information Dissemination  
Use of periodic multicast announcements 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.

- Motivation
- Goals

## 3 Architecture

# SDS Architecture



- Global multicasts authenticated messages
- Authenticated advertisements contain:
  - Certificate Authority and Capabilities Manager contact
  - Address for sending service announcements
  - Service announcement 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*

- ① Continuously 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

- 1 Continuously 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

- 1 Continuously 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
- 3 Generates, stores and distributes appropriate capabilities

# Capabilities Manager

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

# Capabilities Manager

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

# Secure Communications



- **Authenticated Server Announcements**

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

- **Authenticated Server Announcements**
  - Readable by all clients
  - Non-forgable

- **Authenticated Server Announcements**

- Readable by all clients
- Non-forgable
- Reply attack resistant (timestamps)

- **Authenticated Server Announcements**

- Readable by all clients
- Non-forgable
- Reply attack resistant (timestamps)

- **Secure One-Way Service Description Announcements**

Hybrid public/symmetric key system: a packet is sufficient for describing a service which will be decrypted by the SDS server

- **Authenticated Server Announcements**

- Readable by all clients
- Non-forgable
- Reply attack resistant (timestamps)

- **Secure One-Way Service Description Announcements**

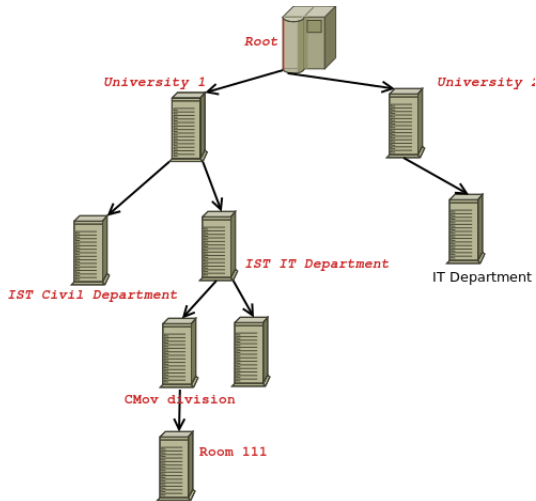
Hybrid public/symmetric key system: a packet is sufficient for describing a service which will be decrypted by the SDS server

- **Authenticated RMI**

A handshake establishes the symmetric key for the session between client and SDS servers and between pairs of SDS servers

- Motivation
- Goals

## 4 Wide Area Support





# Wide Area Support

- 1 Hierarchies built based upon query criteria:

## ① Hierarchies built based upon query criteria:

- Administrative domain

## ① Hierarchies built based upon query criteria:

- Administrative domain
- Network topology

## ① Hierarchies built based upon query criteria:

- Administrative domain
- Network topology
- Physical location

## ① Hierarchies built based upon query criteria:

- Administrative domain
- Network topology
- Physical location

## ② Aggregate service description (lossy)

## ① Hierarchies built based upon query criteria:

- Administrative domain
- Network topology
- Physical location

## ② Aggregate service description (lossy)

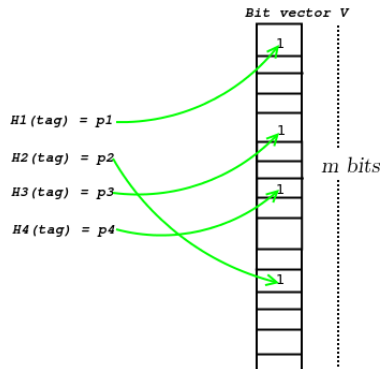
## ③ Use aggregation tables for routing queries

# Lossy aggregation & query routing

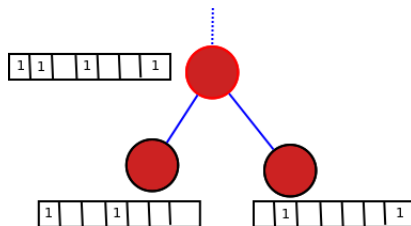
- Hash values of tag subsets of service descriptions used as summary
- Algorithm:
  - ① When adding: compute description tag subset, insert into Bloom Filter table
  - ② When querying: compute tag subsets, examine corresponding entries in Bloom Filter table for possible matches
- Limitations:
  - Computation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



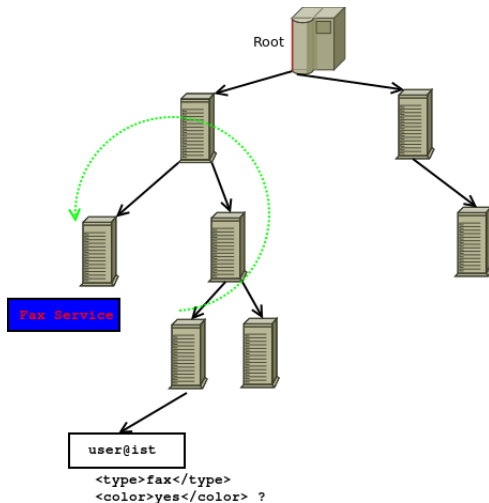
# Bloom filters



# Index aggregation & routing



# Index aggregation & routing



- Motivation
- Goals

## 5 System Performance

<i>Name</i>	<i>Time</i>
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

<i>Files</i>	<i>ms / query</i>
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

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

Table 4: Secure Query Latency Breakdown



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

- Motivation
- Goals

## 6 Conclusion

# Conclusion

Work still needed on:

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

# Questions ?