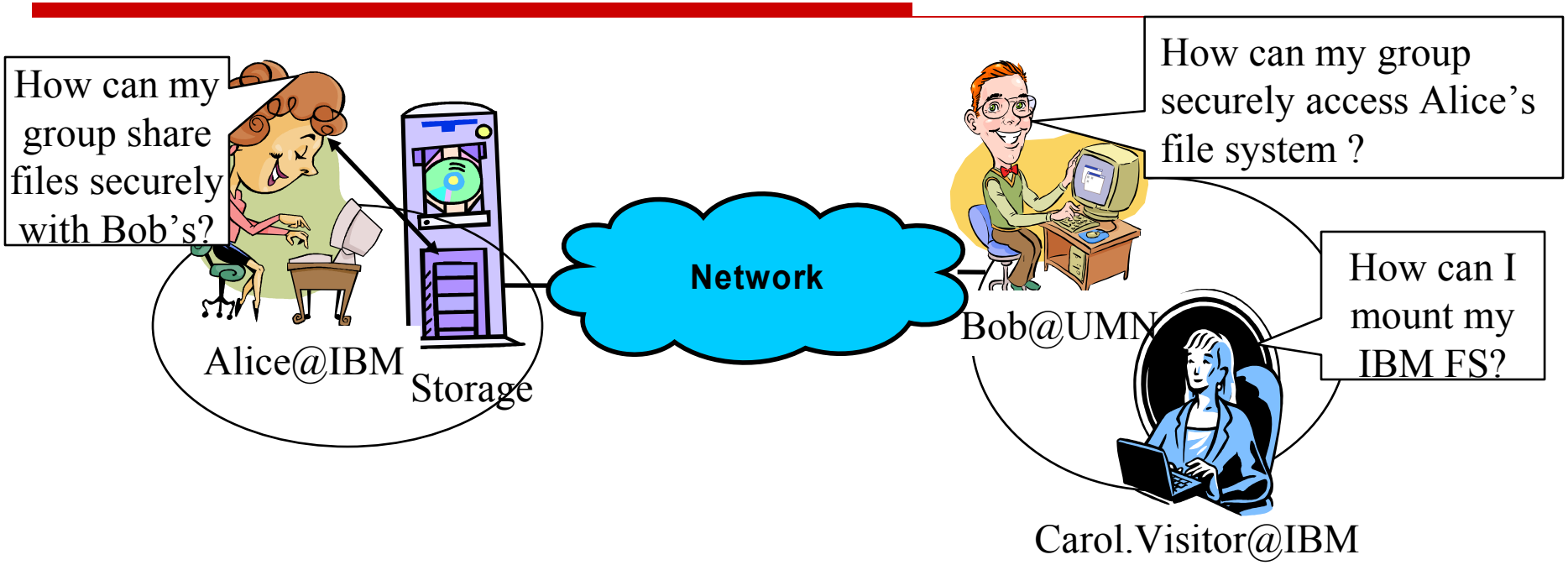
SGFS: Secure, Flexible, and Policy-based Global File Sharing

Vishal Kher
Eric Seppanen
Cory Leach
Yongdae Kim
{*vkher,seppanen,leach,kyd*}@cs.umn.edu
University of Minnesota

Motivation for Network attached Intelligent Storage Devices (NISD)

- Autonomous
  - Utilize available CPU power to perform operations
  - Security, block mgmt, search/indexing, remote query execution
- Improved data sharing
  - Devices can manage meta-data; systems need to handle only naming and location management
- Improved Scalability
  - Clients can directly interact with the devices
- Cost-constrained embedded environment
  - CPU and mem. resources not as powerful as a typical file server

Challenges for Cross-domain Sharing



- Two main challenges
  - Allowing legitimate users to share files across domains without administrative interference
  - Providing consistent file system image irrespective client machine

Overall Goal

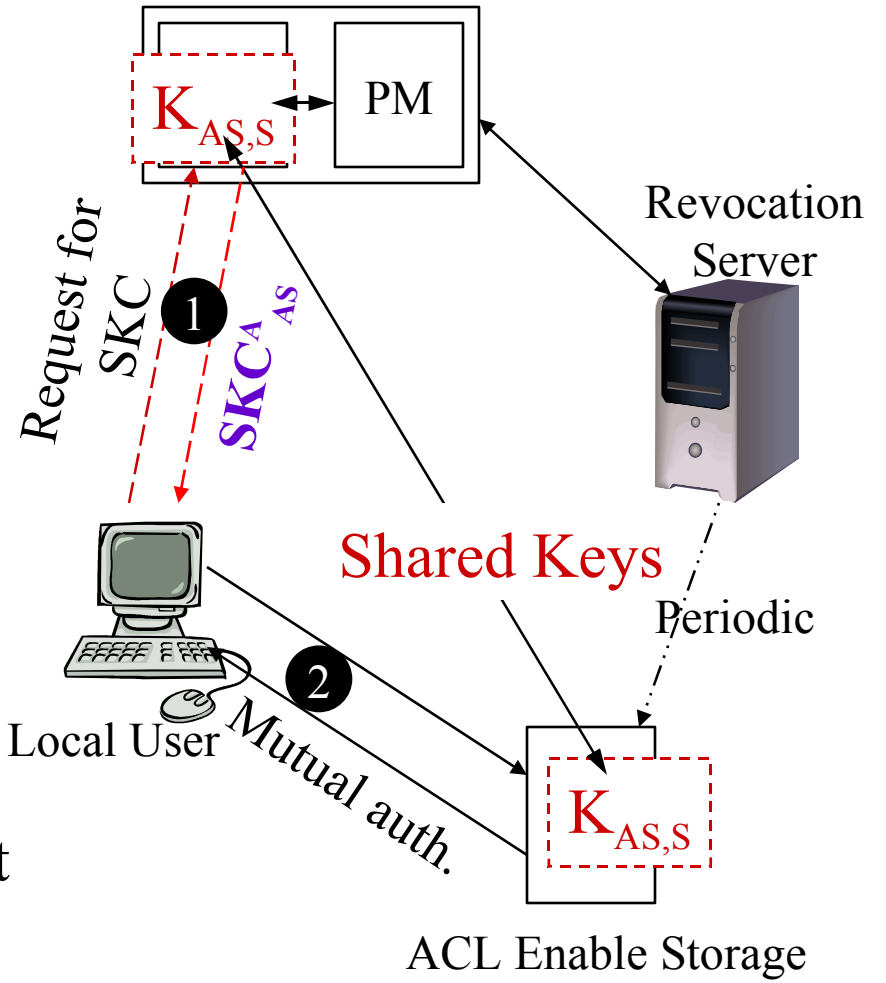
- Developing new mechanisms to allow cross-domain file sharing in the presence of NISD
  - Secure
  - Efficient

Design Goals

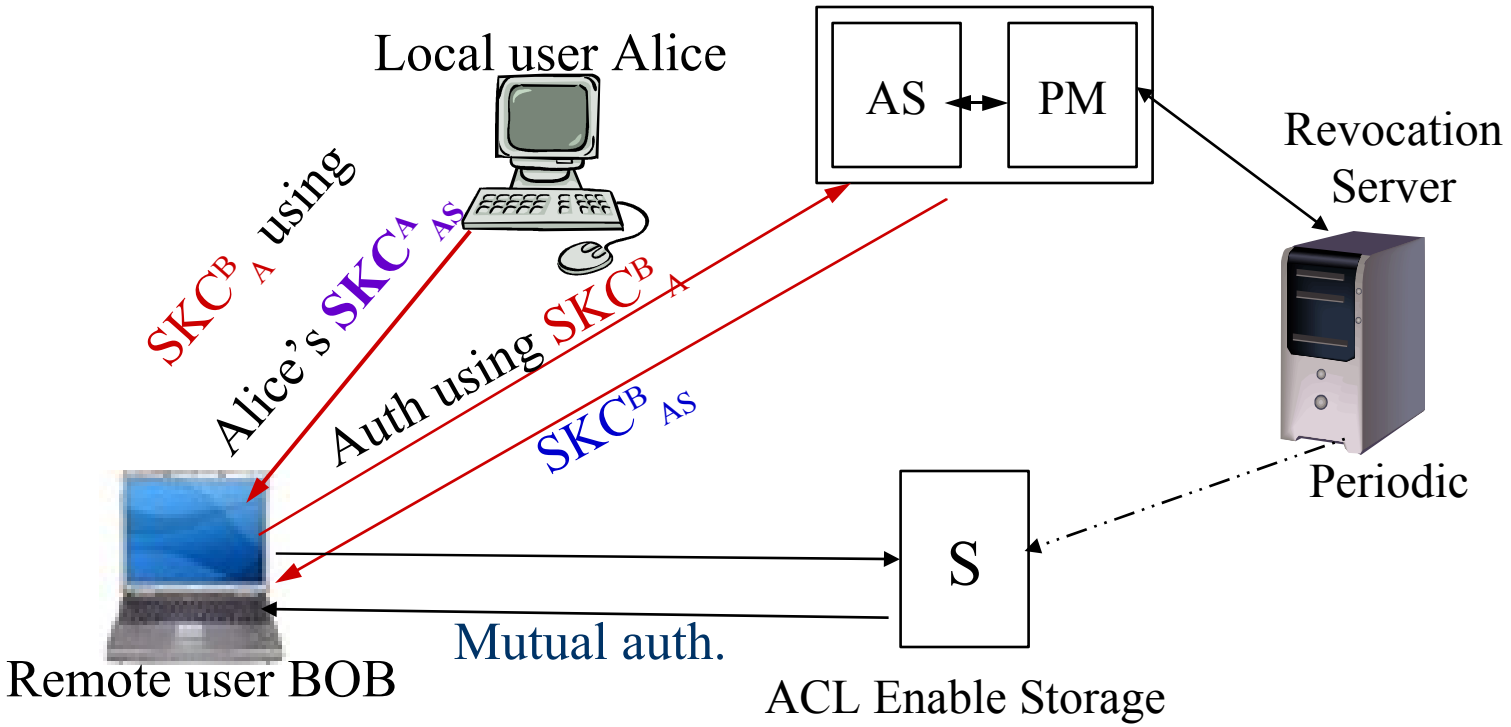
- Minimal administrative interference
  - User should be able to grant access to other users; sharing should not be restricted only to “joined “ domains
- Global file system
  - Users should be able to access files the same way from any machine
- Secure data access
  - Authorized access, encrypted and authenticated data transfer
- Minimal cryptographic overhead on NISD
  - Minimal impact on NISD performance and functionalities
- Flexible policy support
  - Satisfy various environments and various requirements

SGFS Overview

- Trusted Entities
  - Authentication Server (AS)
  - Policy Manager (PM)
  - Storage Device (S)
  - Revocation Server
- Symmetric Key Certificate (SKC)
  - Mimic X.509 attribute certificate
  - Gives power to use different access control models
  - Exploit existing PKI policy managers



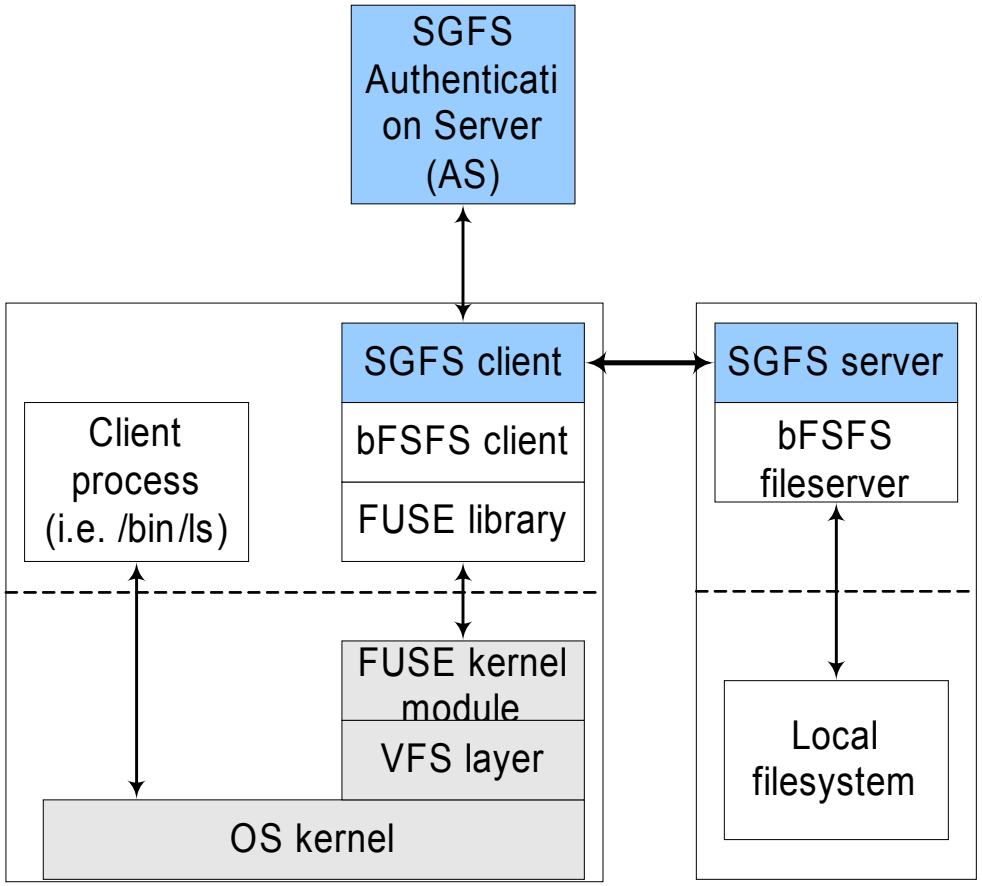
User-to-user Delegation



- Delegation using symmetric key certificates
- Server does not verify chains
- No public-key operations on servers

Prototype Implementation

- SGFS runs in user space and supports generic API
- Currently SGFS runs on top of bFSFS and maskFS
- Key management
  - SKC are stored securely
  - Encrypted using user's public key
- Tool support for users
  - Create and securely manage keys
  - Delegate access rights
- Modularity
  - SGFS interface is independent of FS; any FS can use SGFS



Summary

- Low performance impact on the storage server
  - Symmetric key cryptography - lesser overhead
- Storage server is simple
  - Check whether a client has a valid key or not
  - Perform access control
- User mobility
  - User can store access keys on a smart card, or USB
  - Encrypt with keys public key and move to other machine
- Secure access
- Eliminate central point of failures
  - AS is contacted only once. Files are unavailable only if the storage server is down.

Design Requirements

- User-to-user delegation without administrative interference
- No PKI and certificate chain verification on NISD
  - Minimize computation and communication overheads
- No central point of failures
- Seamless access to files; support user mobility
- Eliminate overhead of resolving remote group names
  - Users should not have to list remote group names on local ACLs
- Support for various access control models
  - UNIX, RBAC
- Centralized policy management
- Ease of management

Traditional Solutions

- Traditional solution for cross-domain sharing - create accounts
  - Requires interaction with system administrators: not flexible
- Kerberos
  - No user-to-user delegation
  - Administrative overhead setup realms
- User-to-user delegation using PKI
  - Storage devices have to verify a chain of certificates
  - Computation overhead as well communication overhead since verifying it might require traversing trust hierarchies

Status

- Completed
  - Architecture design
  - Design of security and key management protocols
  - File system design
  - Implementation of user-level file system layered on FUSE
  - Implementation of security protocols
- Future
  - Performance evaluation
  - Design and implementation of revocation server
  - Auto mounting and global naming
  - Using appropriate policy engines

Research Goals

- This research focuses on developing new mechanisms to allow secure and *efficient cross-domain file sharing in the presence of NISD*
- Minimal administrative interference
  - User should be able to grant access to other users; sharing should not be restricted only to “joined “ domains
- Global file system
  - Users should be able to access files the same way from any machine
- Secure data access
  - Authorized access, encrypted and authenticated data transfer
- Minimal cryptographic overhead on NISD
  - Minimal impact on NISD performance and functionalities
- Flexible policy support
  - Satisfy various environments and various requirements