An IPSec-based Host Architecture for Secure Internet Multicast

R. Canetti, P-C. Cheng, F.Giraud, D. Pendarakis, J.R. Rao, P. Rohatgi,

IBM Research

D. Saha

Lucent Technologies

Motivation

- In today's Internet the need for *efficient* and *secure* multicast communication is growing.
- Most works on designing secure multicast mechanisms concentrate on the global architecture and design of group control entities.
- We present a host architecture for a member in a secure multicast group.

In this talk:

- Background on secure IP multicast:
  - Some applications
  - Security requirements
  - Overall design of secure IP multicast groups (as developed in the IRTF)
- Basic design tenets of host architecture
- Overview of the design
- Outstanding issues

Multicast communication:

Whenever there are multiple recipients

- Typical applications:
  - File and software updates
  - News-feeds
  - Video/audio broadcasts
  - Virtual conferences, town-hall meetings
  - Multiparty video games

Security requirements

- Limiting access to group communication:
  - Long-term secrecy
  - Ephemeral access restriction
- Authentication:
  - Group
  - Source
- Anonymity
- Availability (against denial of service attacks)

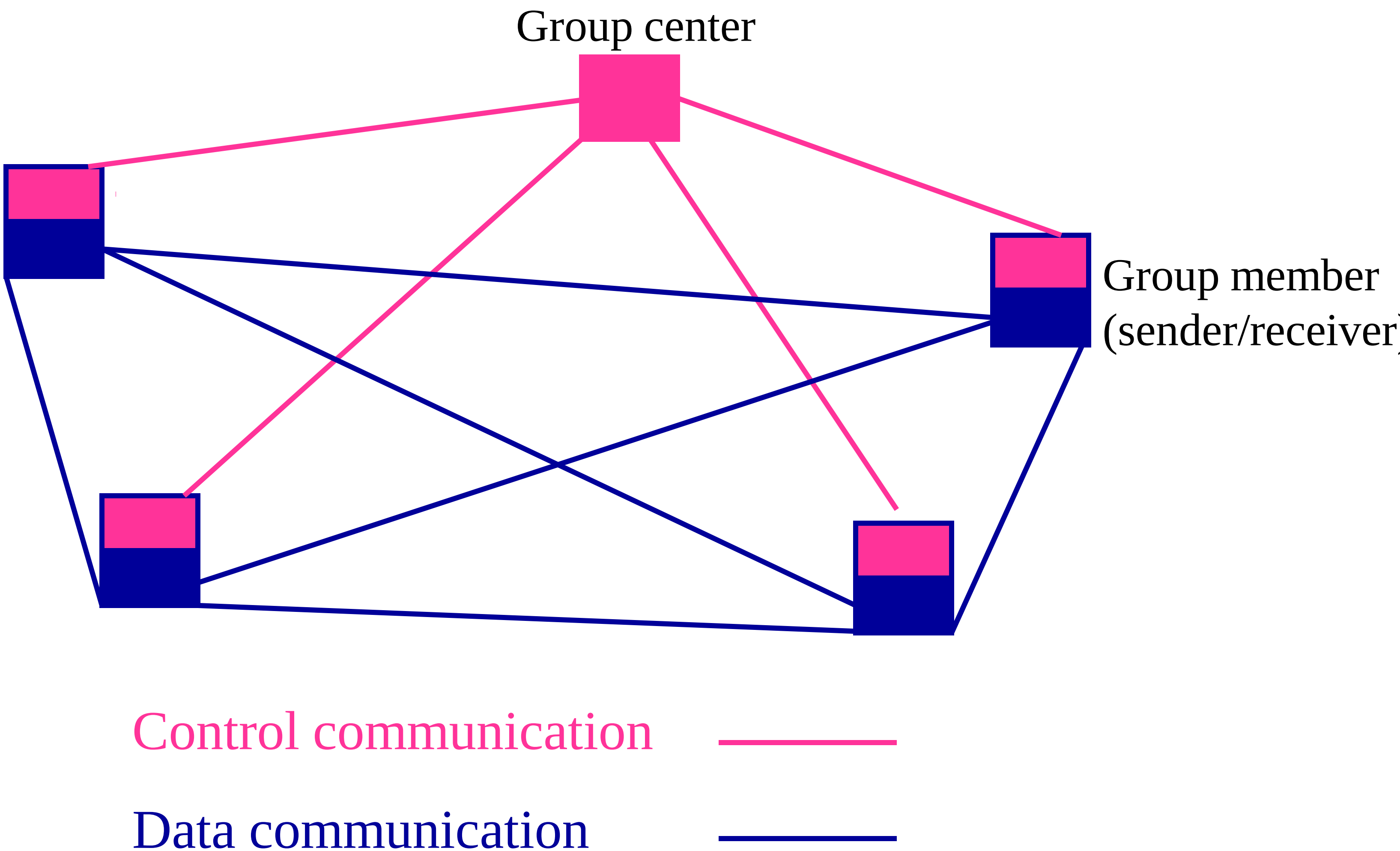
Work done at the Secure Multicast Group (SMuG) of the IRTF:

- Set focus on prominent scenarios and issues
- Develop overall architecture for secure IP multicast and research for appropriate protocols that can be standardized

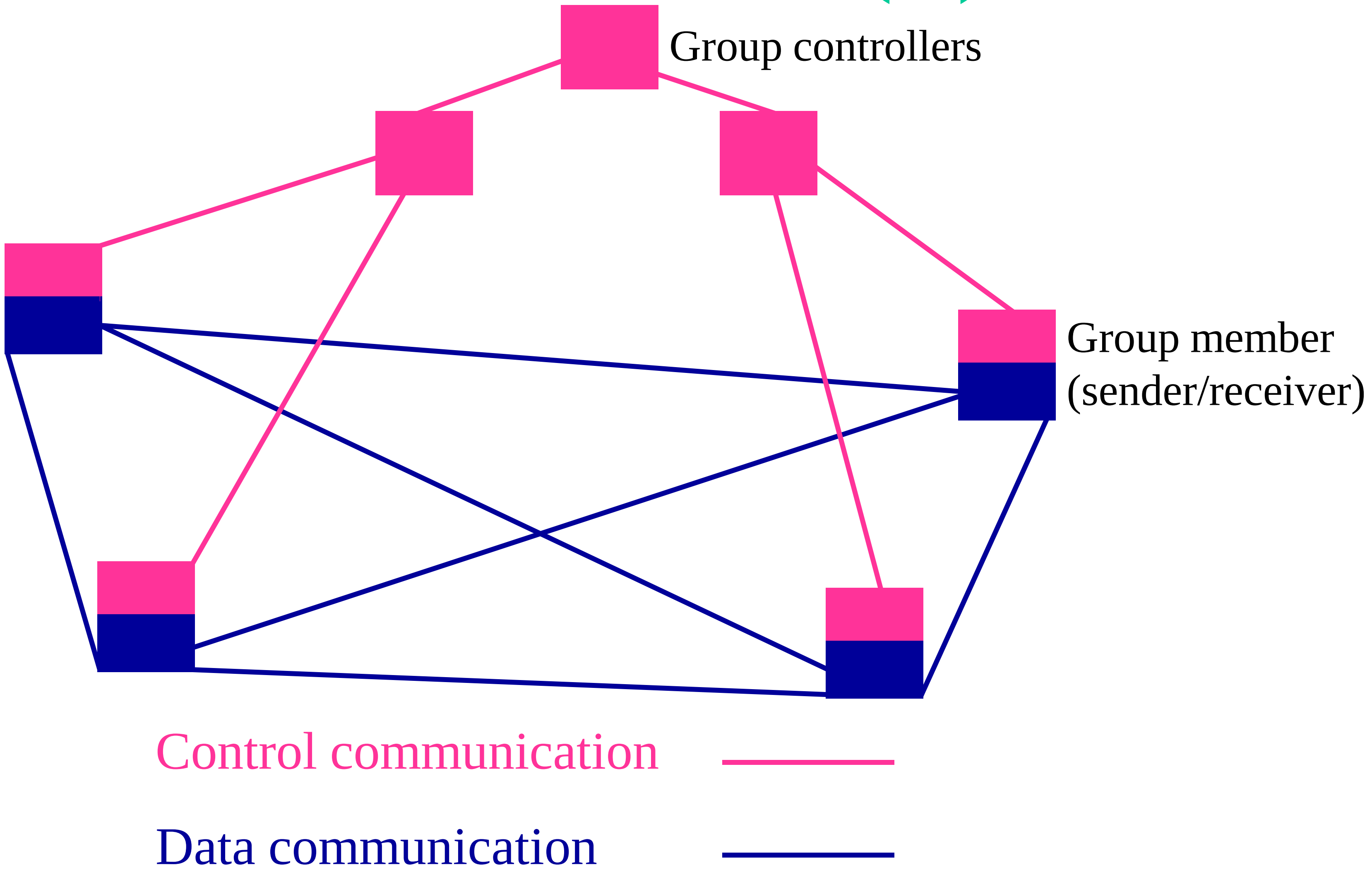
A prominent scenario:

- One-to-many communication
- Medium to large groups (10-100K)
- Centralized group management
- No trust in group members
- Need source authentication, ephemeral encryption
- Dynamic membership

Global architecture for secure multicast (I):



Global architecture for secure multicast (II):



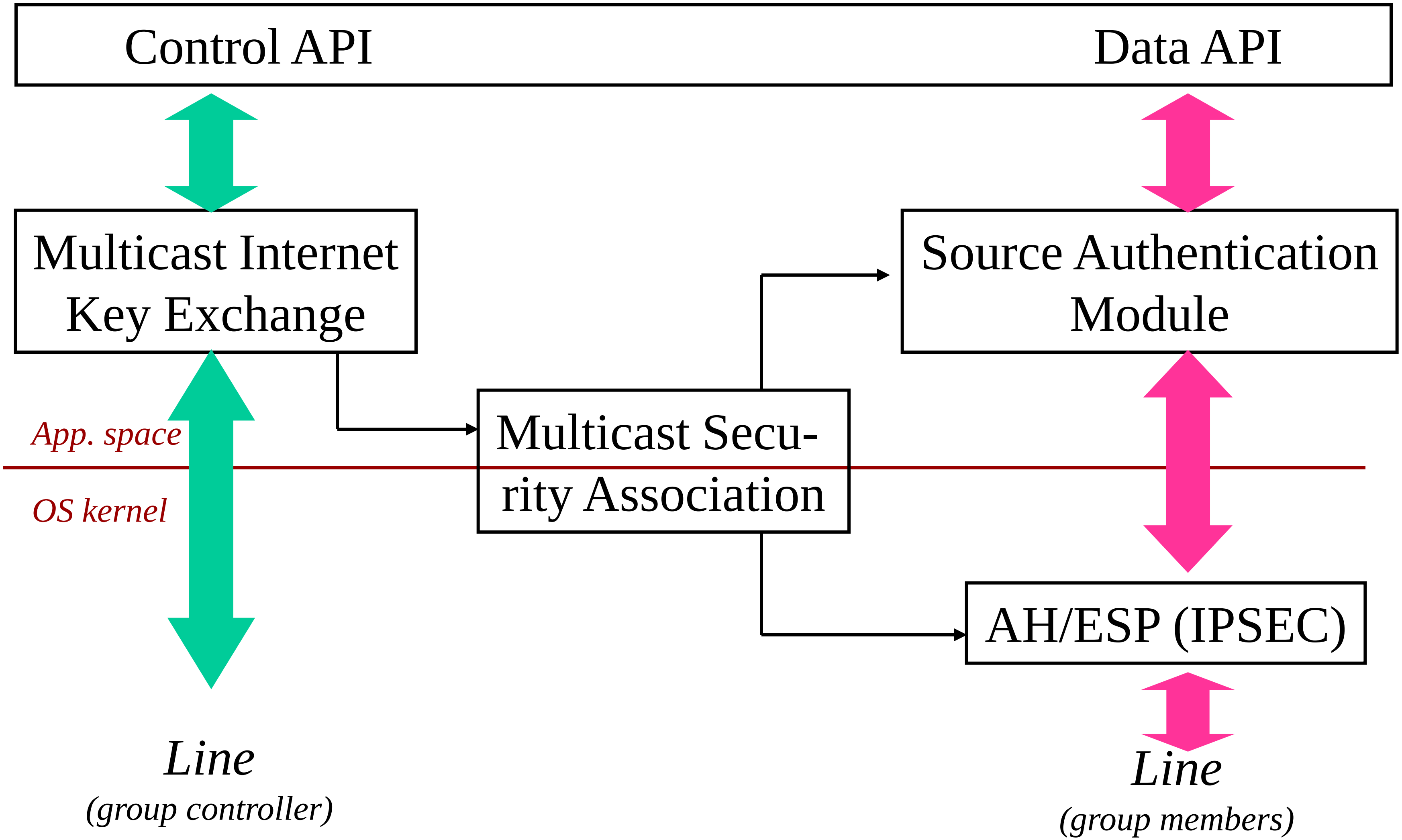
Host architecture: Design tenets

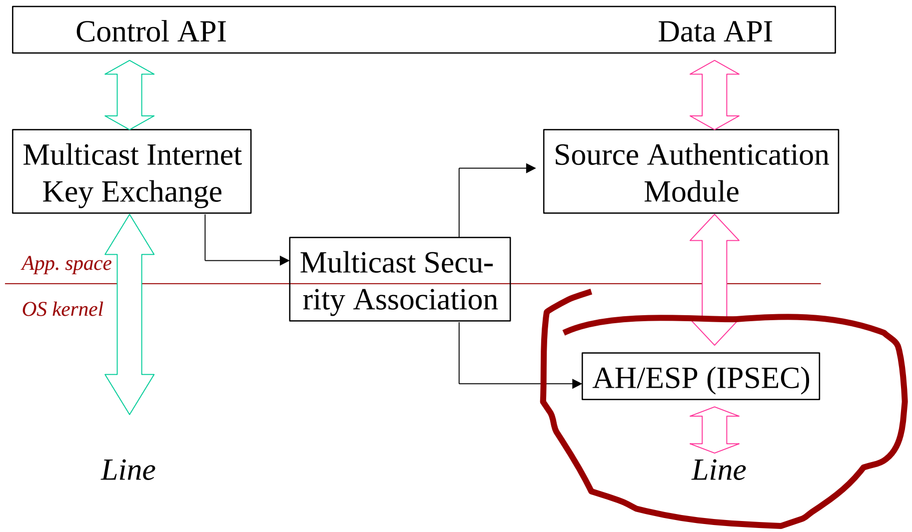
- The security mechanism should be independent of the routing method.
- Separate key management from data handling
- Use existing components when possible (In particular, IPSec)
- Minimize changes to OS kernel
- Maintain ability to plug-in different crypto algorithms

An IPSec-based design

- Motivation:
  - Build on solid and (soon to be) ubiquitous protocol.
  - Provides security in kernel, minimal load on applications.
- Drawbacks:
  - Tie the design to existing protocols
  - Have to deal with compatibility

The architecture at a glance



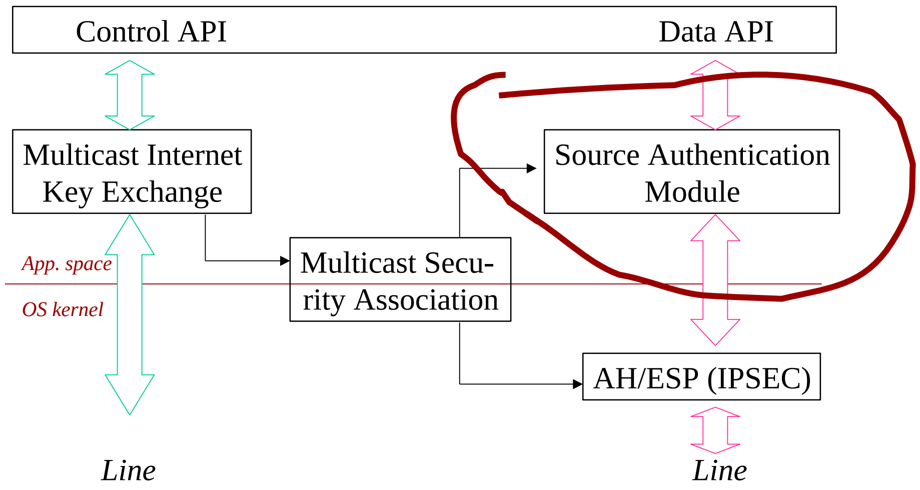


IPSEC transforms (AH/ESP):

-Data encryption with group key

-Group authentication with group key

-Operates on individual packets (No state across packets)



SAM

Signing data efficiently requires:

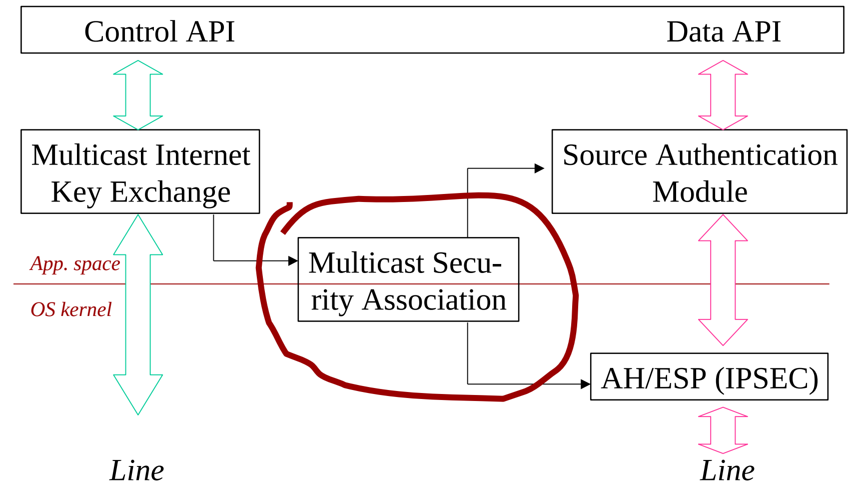
-Signing data in large chunks

-Keeping state across packets

Therefore, SAM is in transport layer (UDP), operates on UDP frames.

Possible realizations:

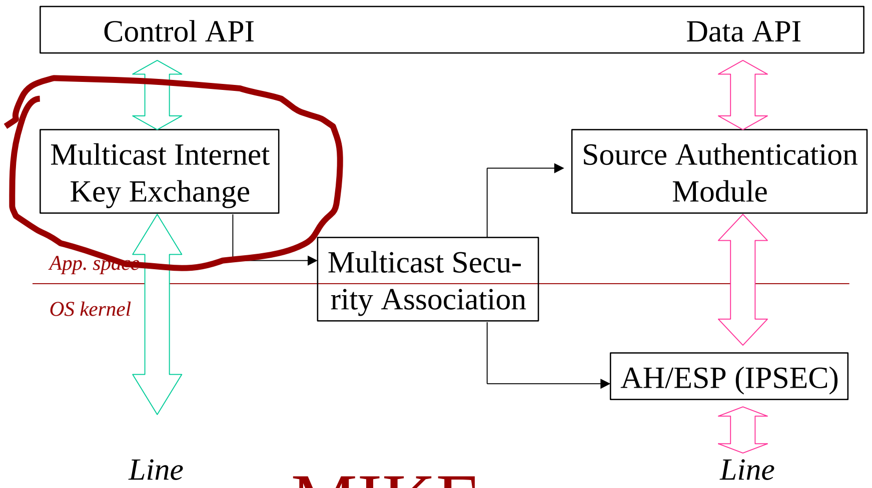
[Wong-Lam 98], [Rohatgi 99], [C+ 99], [Perrig et.al. 00],...



MSA is a database that holds:

- IPSec SA for AH/ESP (group key, algorithms, group address, etc.)
- Information for SAM (Signing/verification keys, algorithms, etc.)
- Re-keying information for MIKE (e.g. path in “LKH tree”)
- Point-to-point SA with the center

Note: MSA is periodically updated by MIKE.



MIKE:

- Invoked by API to join/leave multicast group.

  Join/leave interaction done via standard point-to-point secure connection (such as IPSec, SSL) with the center.

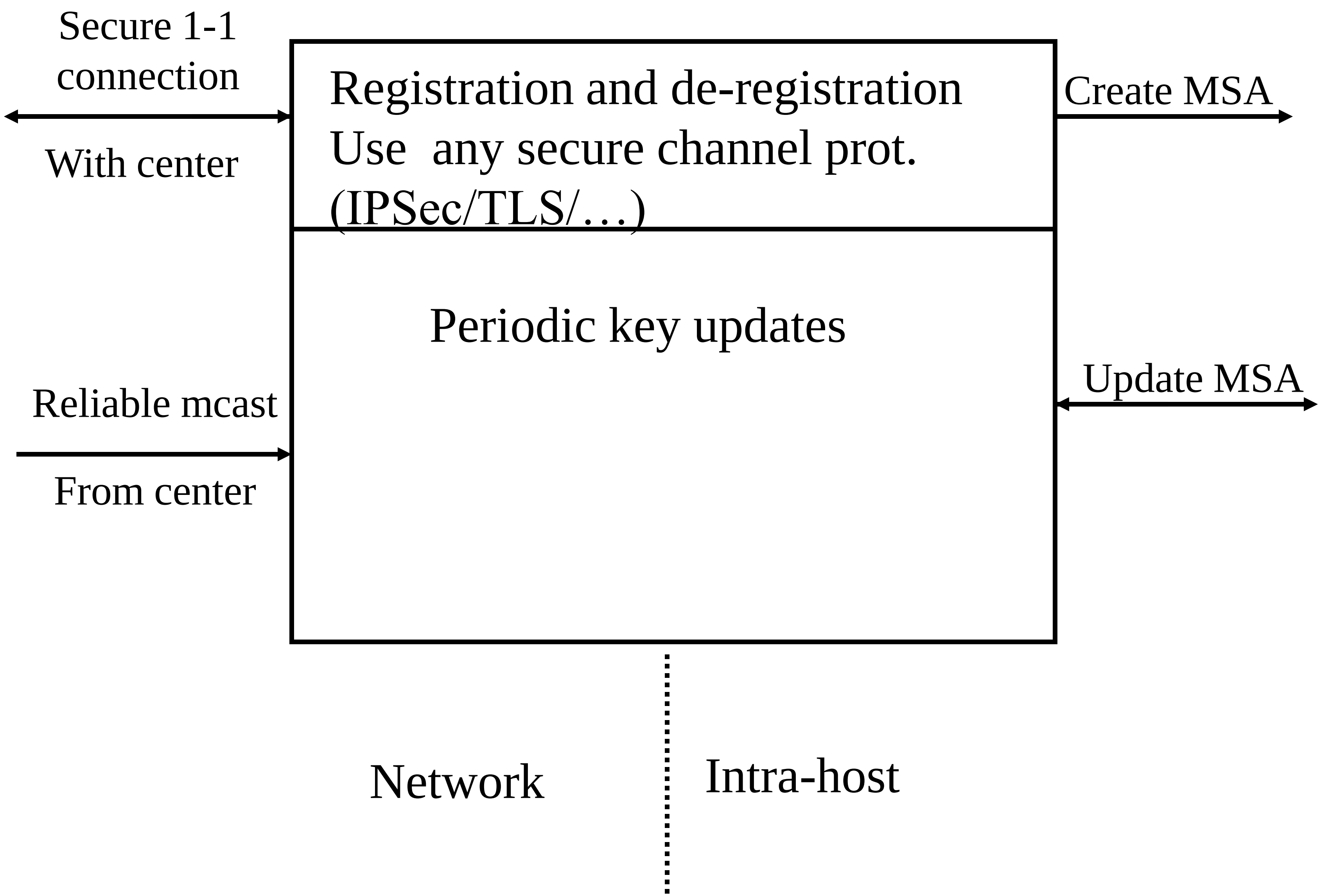
- Receives key updates from controller and updates MSA

-Key updates assume a "reliable multicast shim".

(Can be implemented by any general RM protocol

or by a special purpose protocol.)

Design of MIKE



Outstanding issues

- Handling multi-user hosts:
  Need to provide intra-host access control.
  - MSA must list member applications/users
  - Allow only members to listen to group traffic.
    Can do either:
    - In kernel. (More efficient, needs kernel modification)
    - Using daemon process (Less efficient, no kernel modification).

Outstanding issues

- MSA identification and choice of SPI:
  - An IPSec SA is identified by receiver address, SPI, protocol. SPI is chosen by the receiver.
  - Here SPI cannot be chosen by receiver.
  - Instead it is chosen by the group center.
- Replay protection field:
  - In IPSec, increasing counter set by sender, receiver free to ignore.
  - Unchanged for single sender multicast. With multiple senders receiver must ignore.

Validation of architecture

- Implemented the architecture on Red Hat Linux 5.1, using Freeswan version 0.91 implementation of IPSec.
- Needed a “patch” to make Fswan work with IP-multicast (class D) packets. (Seems to be a pecularity of Fswan implementation.)
- Architecture works smoothly, with good performance.

Conclusion

- Described an IPSec-based host architecture for secure multicast.
- Architecture is compliant with global architecture as developed in the IRTF.
- Can be installed with little or no modification to OS k ernel, with good performance.