An Architecture for Secure Internet Multicast

draft-ietf-ipsec-sec-mcast-arch-00.txt

R. Canetti, P-C. Cheng, D. Pendarakis,
J.R. Rao, P. Rohatgi, D. Saha
IBM

A Host Architecture for Secure IP Multicast

- Identifies basic components, their functions and interactions
  - Details of each component to be finalized later based on **SmuG consensus**.
- Structured to meet multicast security requirements
  - (e.g., listed in taxonomy draft [Canetti, Pinkas])
- Consistent with SmuG charter
  - Flexibility in crypto and group key management mechanisms.
  - Independence from routing/reliability mechanisms.
  - Reuse of existing components where possible.
  - Suitable for main multicast usage scenarios, e.g., one-many, few-many and many-many.

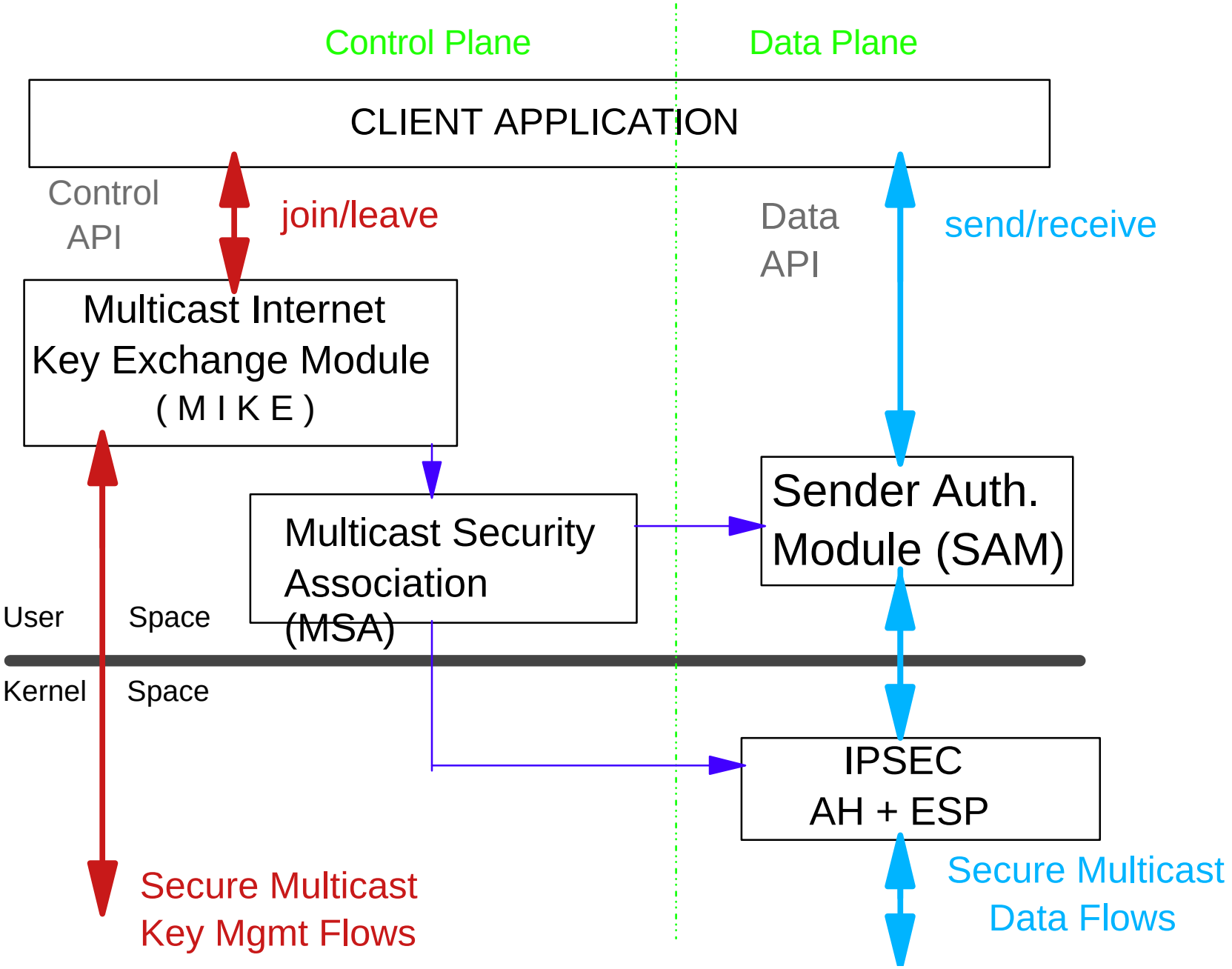
Security Requirements

- Access Control/Group Communications Secrecy
  - Usually done by encrypting multicast data with group secret key.
    - Joining members provided with key(s) upon member authentication.
    - Key(s) may require secure update when a member joins/leaves.
  - Must be able to handle highly dynamic groups.
- Group data authentication
  - Guarantees that data originated from within the group.
- Individual Source authentication
  - Guarantees that data originated from a particular source.
- Non-Repudiation, Anonymity
  - Not addressed, left to specific implementations

Design Goals and Guidelines

- **Goals**: Simplicity, Flexibility, Ease of incorporation in existing systems
- **Guidelines**:
  - Independence from underlying routing/reliability mechanisms
  - Mimic IPSEC architecture and design as much as possible
    - Separate data and key management flows.
    - Keep key management in Application Layer.
  - Reuse existing components where possible
    - AH/ESP protocols in IPSEC can provide group data confidentiality/authentication but not source authentication.
  - Minimize modification to OS Kernels
  - Flexibility in choice of crypto and key management schemes
    - ESP/AH allows flexibility in crypto. Use frameworks for other components.

Architectural Block Diagram



ESP/AH Usage for Multicast

- Reuse current IPSEC implementations.
- Existing ESP protocol in IPSEC to be used for encryption/group-authentication of IP Multicast Data Payload.
- Existing AH protocol in IPSEC to be used for group-authentication of IP Multicast Data Payload + Immutable fields of Header.
- Controlled by **Multicast Security Association.** **MSA** = IPSEC SA + Secure Multicast specific info (e.g., source auth info).

SAM Functionality

- Resides above IP-Multicast and IPSEC.
- Implements the Data API, i.e., send/receive.
- Adds source authentication information on outgoing data.
- Authenticates source of multicast data received from IPSEC/IP-Multicast Layer.
- Authentication information can be self-contained in each Data packet or spread over multiple Data packets.
- Controlled by **Multicast Security Association (MSA).**

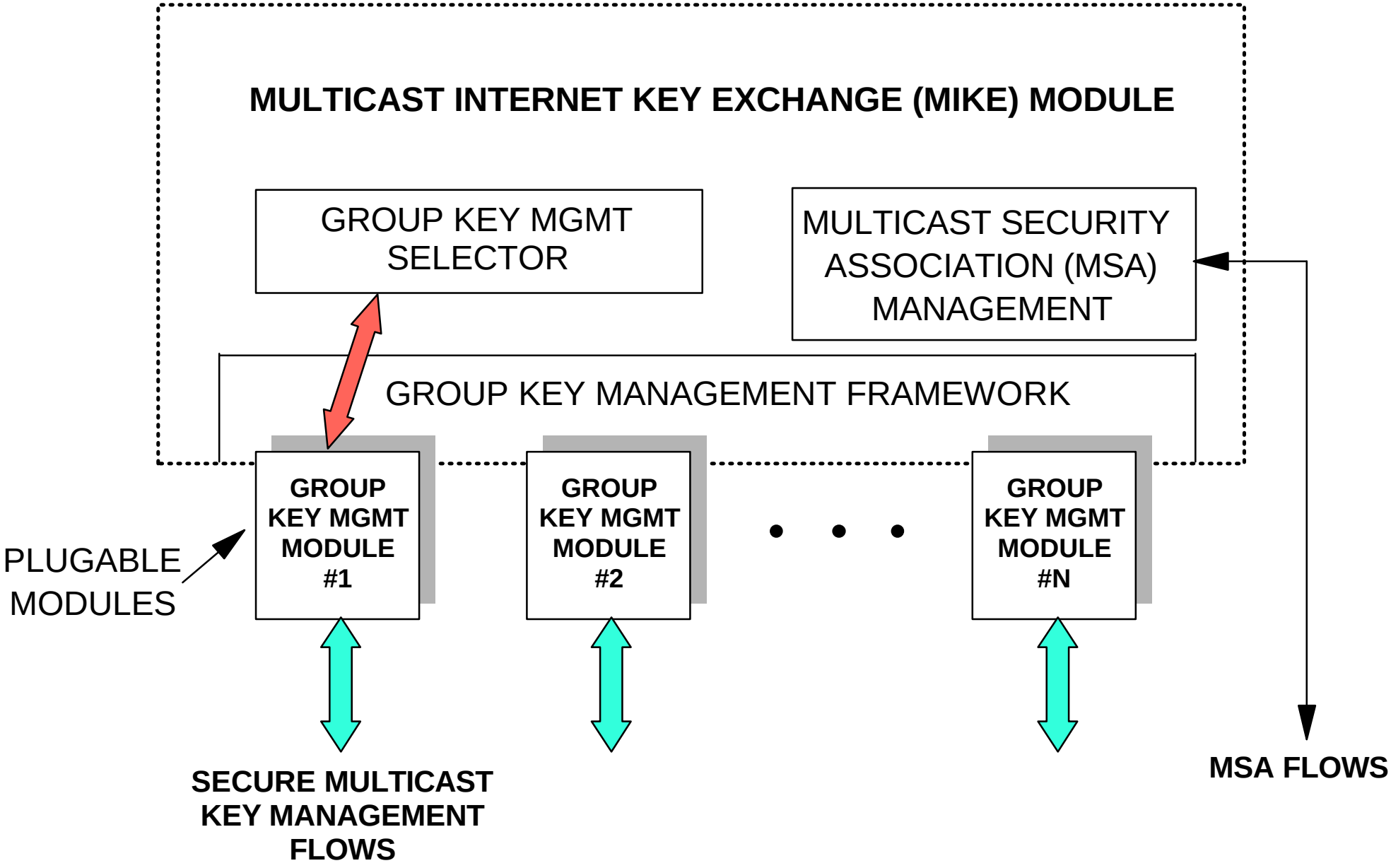
MIKE Functionality

- Handles Multicast Key Management: privacy, group and source authentication.
  - Key Management relies on hosts communicating with group controller(s).
- Implements Control API (secure group join/leave)
- Maintains Multicast Security Association (MSA) which is associated with at least the following:
  - Group keys for encryption/decryption and authentication
  - Source authentication keys
  - Connection specific information (e.g., SPI)

MIKE Requirements

- Must be a **framework,** to accommodate wide range of multicast group key management schemes.
- Ability to operate with single group controller; extensible to multiple controllers
- Should allow distribution of keys among all group members
  - symmetric encryption + authentication
  - (temporary) public keys of controller and senders
- Placement in the application layer

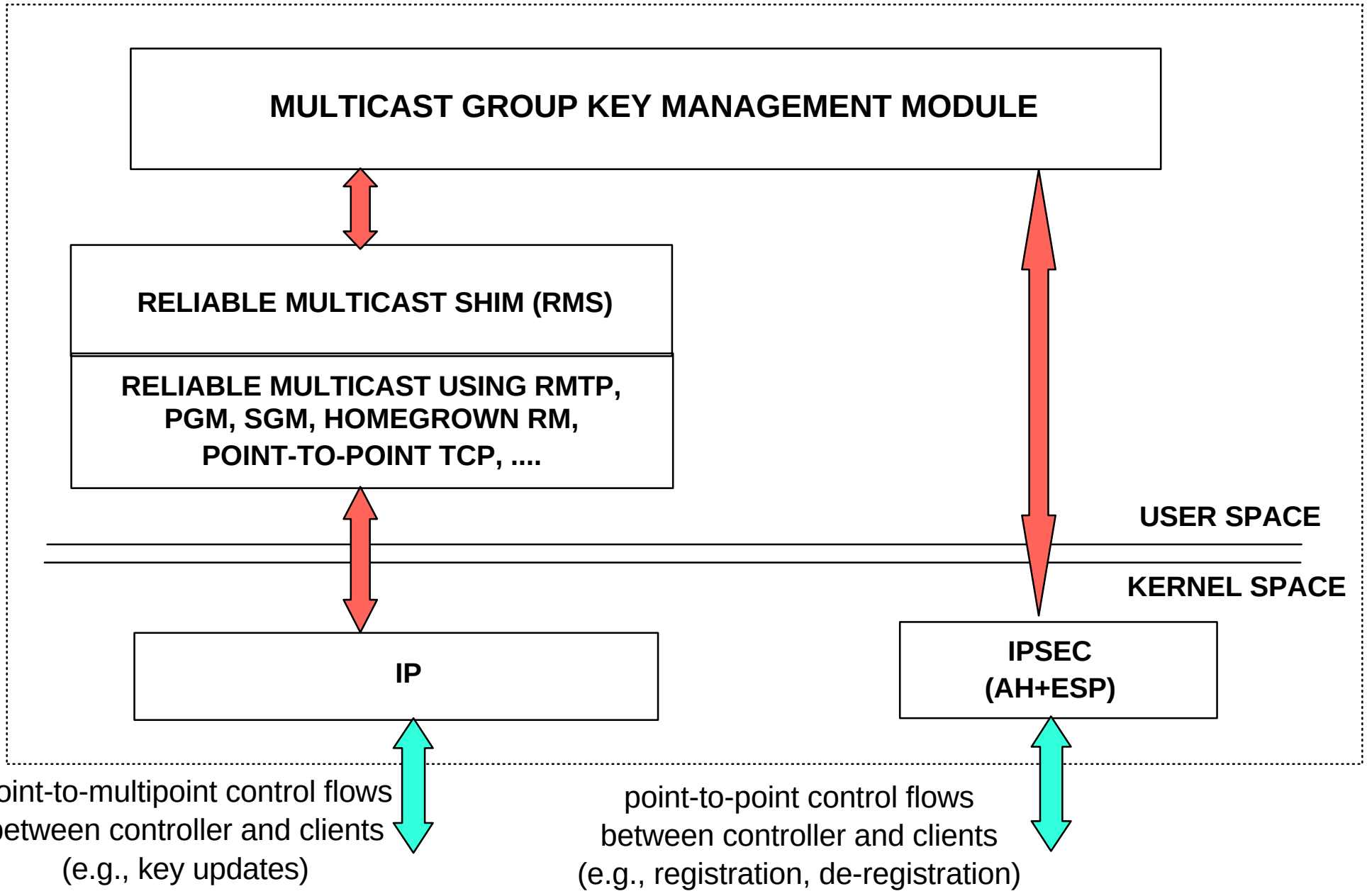
MIKE Block Diagram



Example Multicast Key Management Module

- Objectives: simplicity, reuse of standards
- Point-to-point connections secured via IPSEC/IKE
  - e.g. registration + de-registration between clients and controller
  - keying information transmitted as *data over unicast SA*
    - unicast SA need not be long-lived
- Key updates reliably multicasted (e.g. Wallner)
  - Reliability provided by "Reliable Multicast Shim" Layer
    - Implemented on top of any reliable multicast scheme or via emulation (e.g. using TCP).
  - Source authentication via public key signatures
    - public keys distributed at registration

Example Multicast Key Management Module



ISSUES with REUSE of IPSEC

- Identification of Multicast SAs
- SPI Assignment
- Sequence Number Handling/Replay Prevention

Identification of Multicast SAs

- Definition from the IPSEC Security Architecture document
  - Applies to either unicast or multicast
- Combination of:
  - Destination Address (Class D address for multicast)
  - Security Parameter Index (SPI)
  - Protocol (ESP, AH)

- Implementation note:
  - Current IP multicast implementations *might* discard packets with Class D destination address whose protocol is not UDP
  - Change required to use AH + ESP protocols for mcast

SPI Assignment

- In IPSEC SPI assigned by (single) receiver
- Impractical in the case of multiple receivers
- Alternatives:
  - assignment *by sender*: impractical for multiple senders
  - better idea:
    - SPI for each mcast group selected by group controller
    - communicated to clients during registration

- Two options for assignment:
  - single SA and SPI per group
  - separate SA and SPI *per sender*

Sequence Number Handling/Replay Prevention

- The problem:
  - in AH and ESP, sender MUST increment sequence number counter, starting from 1.
  - BUT, with multiple senders in the same SA, no consistency or monotonicity!
  - -> have to make sure receivers do not perform sequence number processing + verification

- Possible solutions:
  - Use multiple SAs, one per sender
  - Place protection in higher layer (e.g. SAM)