Iolus: A Framework for Scalable Secure Multicasting

Suvo Mittra

Presented by Leland Smith

CS 6204, Spring 2005

Overview

- What is multicast?
  - Characteristics
  - Limitations
  - How to secure?
- Motivation
  - Scalability
- Iolus description
  - Design requirements
  - Protocol summary
- Discussion

Multicast [1]

- Characteristics
  - Any number of principles for a group
  - Membership of groups change
  - Security association must be dynamic
  - Difficult to extend features in a scalable way
    - Including security…
    - Fundamentally different requirements from unicast.
- Limitations
  - More susceptible to attack
  - More opportunities for interception of traffic
  - Larger number of principles affected by attack
  - Easier for attacker to target an attack
  - Easier for attacker to pose as a legitimate principle.

Multicast [2]

- Securing Multicast
  - Security association among authorized principles
  - Blocks of time
  - Principle authorized to participate in multicast during various blocks of time.
  - Only allowed to participate when authorized to do so.
  - Security association must be changed on each join and leave to ensure that:
    - Leaving entity is no longer able to access further transmissions.
    - Joining entity is not able to access previous transmissions.

Motivation [1]

- Key Management
  - Main difference between multicast and unicast
  - When and how to rekey?
- Scalability
  - Two failures
    - **1 affects n**
      - Joins and leaves require all members to process the key change when one new member joins
    - **1 does not equal n**
      - Protocol cannot deal with group as a whole and must consider each member individually
      - Leaves

Motivation [2]

- Updates require at-least-once semantics
  - Potential for transient security breaches.
  - Receiver fails to receive updated key
    - Will not be able to continue decrypting group communications
    - May accept communications from members that have been removed from the group but still have old key.
  - Sender fails to receive updated key
    - Will continue to encrypt using old key
    - Receivers will be unable to decrypt transmissions
    - Former group members will be able to continue decrypting transmissions.
- Large & dynamic membership
  - Increase control traffic and probability of security breach.
  - Need to maintain integrity of group.
  - How?

Iolus Framework

- Enables scalable, secure communication among a group of principles using multicast.
- Can be used to build protocols that:
  - Secure arbitrary multicast transmissions
  - Implement a common key management service that works in conjunction with specialized applications.
  - Provide a group key management service to unicast applications
  - Secure distribution tree

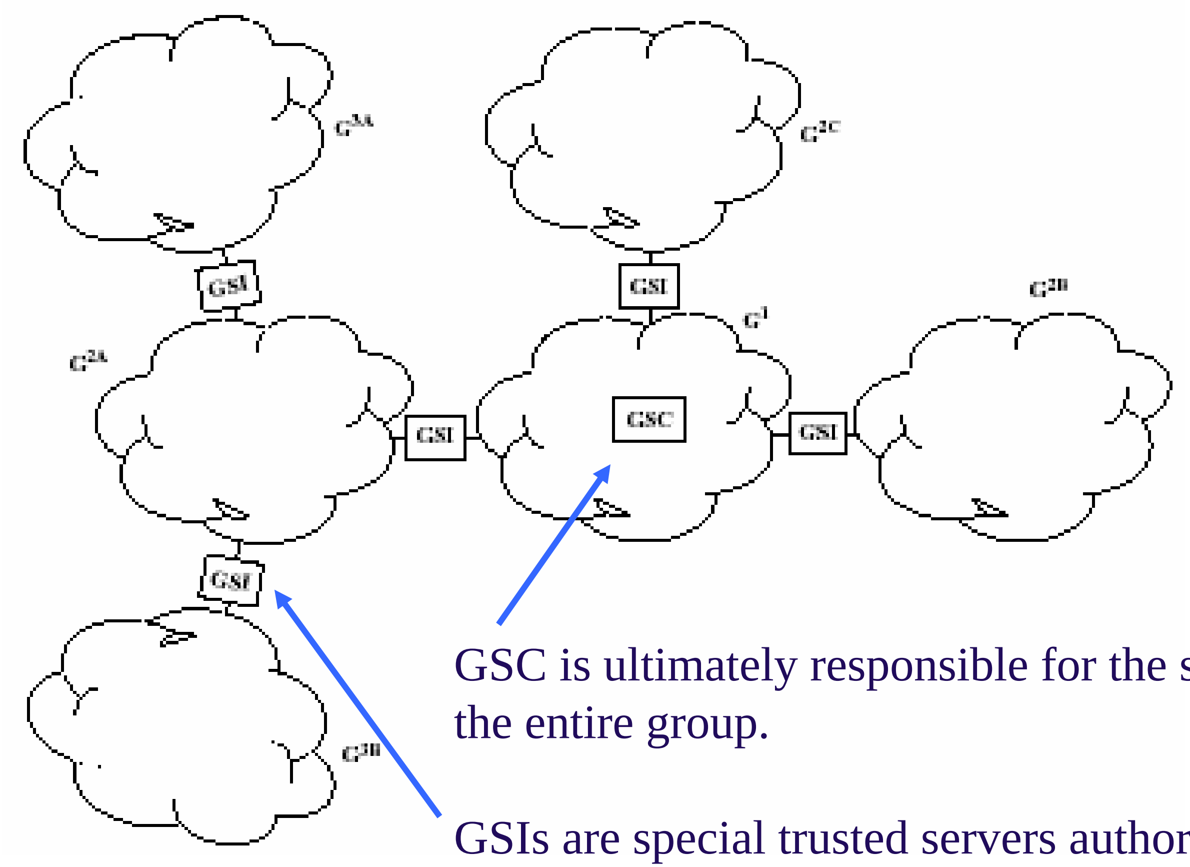
Iolus Design Requirements

- Scalability
  - No arbitrary limits
- Robustness
  - Adapt gracefully to network disruption
  - Minimize and localize effects of disruption
- Security Objective Independence
- Security Technology Independence
- Communication Protocol Independence
- Protocol Layer Independence

Iolus Framework

- Secure distribution tree
  - No single flat secure multicast group
  - Composed of a number of smaller secure multicast “subgroups” arranged in a hierarchy
  - Forms a single “virtual” secure multicast group.
  - Subgroups are relatively independent
    - Each subgroup has its own keying material
    - No global key
    - Only the local subgroup key needs to be changed on joins and leaves
- Management Entities: more three letter names…
  - Group security controller (GSC) manages the top-level subgroup.
  - Group security intermediaries (GSIs) manage each of the other subgroups.
  - Collectively, the GSC and GSIs are known as group security agents (GSAs)
  - GSAs connect the subgroups and work together to deliver locally multicast data to all the subgroups in the virtual group.

Example



GSC is ultimately responsible for the security of the entire group.

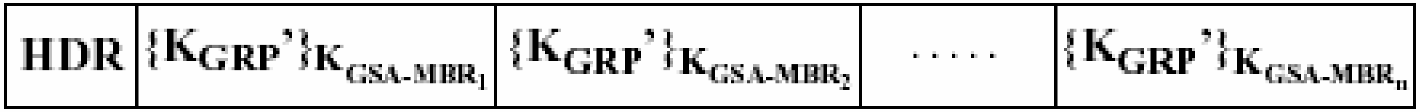
GSIs are special trusted servers authorized to act as proxies of the GSC or their parent GSI and control their local subgroup.

Joining a Group in Iolus

- To join a group, sender or receiver:
  - Locates its designated GSA
  - Issues a JOIN request on a *secure unicast channel*.
- Upon receiving a JOIN request
  - GSA queries database of allowed users
  - If request is approved:
    1. Generate a secret key $K_{GSA\text{-}MBR}$, shared only with joining member
    2. Store this secret key in a local database
    3. Communicate $K_{GSA\text{-}MBR}$ to the new member using the secure channel
    4. Change group key and multicasts a GRP_KEY_UPDATE message containing the new group key encrypted with the old group key to the current multicast subgroup
    5. Communicate new group key to the joining member via the secure unicast channel.

Leaving a Group in Iolus

- Potential for voluntary or involuntary leaves
- Subgroup key needs to be changed
- Problem: no keying material that can be used to securely communicate new subgroup key only to the remaining members.
- Solution: send a copy of the new subgroup key to each member encrypted with that member's $K_{GSA\text{-}MBR}$.
  - Not unicast!
  - Multicast one message containing n copies of the new subgroup key each encrypted with a different member's $K_{GSA\text{-}MBR}$.
  - Have each member pick out the one they can decrypt.
  - Example:

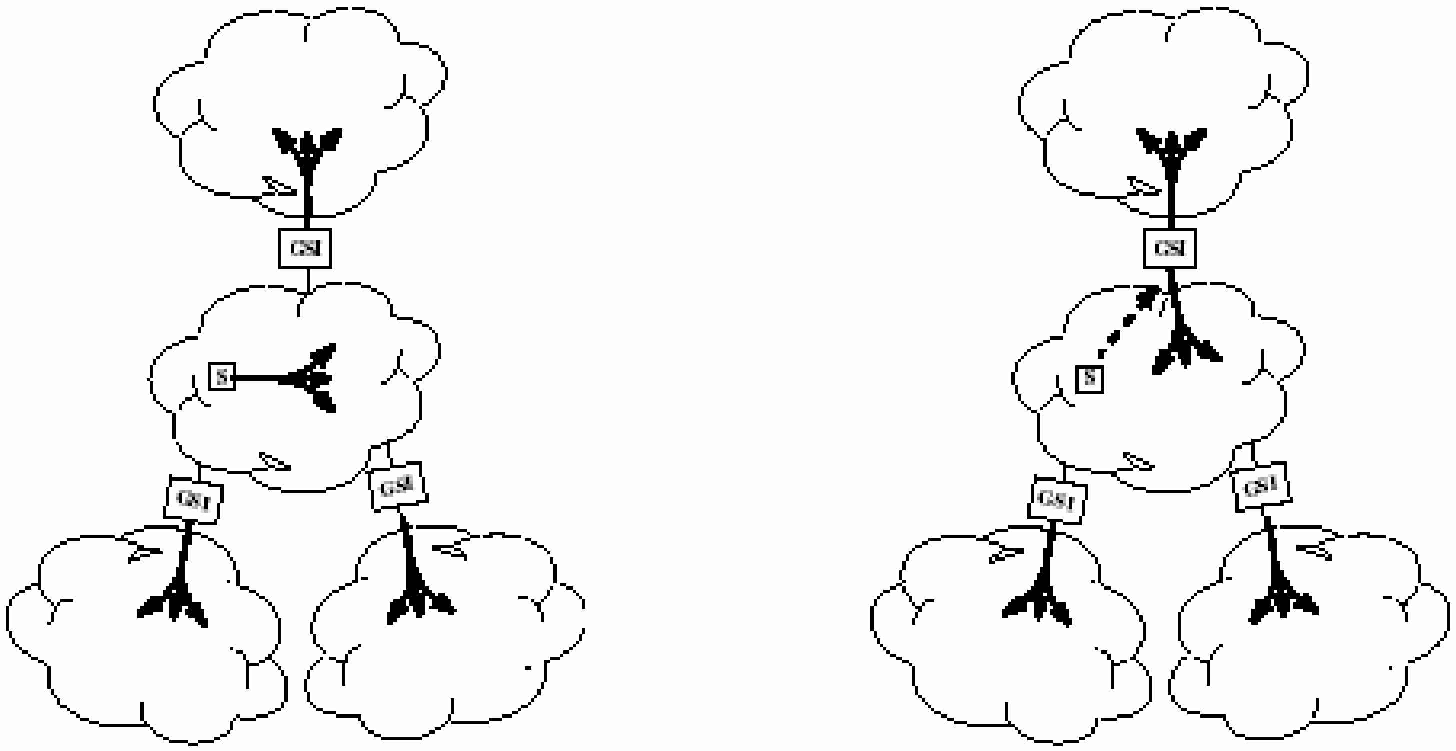


HDR	$\{K_{GRP}'\}K_{GSA\text{-}MBR_1}$	$\{K_{GRP}'\}K_{GSA\text{-}MBR_2}$	. . . . .	$\{K_{GRP}'\}K_{GSA\text{-}MBR_n}$

Data Transmission in Iolus

- Sender could multicast data directly to the subgroup encrypted with the subgroup key.
  - Parent GSI could listen for multicasts, decrypt them then remulticast them to its parent subgroup encrypted with that subgroup’s key.
- Sender unicasts data to the GSA encrypted with their unique $K_{GSA\text{-}MBR}$.
  - GSA decrypts data, re-encrypts it with the subgroup key, signs it, then multicasts it to the group as well as to its parent subgroup.
  - More secure, removes possibility of a sender sending a message with an outdated subgroup key.
  - Receivers have assurance, through the GSA’s signature, that the message is from a valid source.

Transmission Example



(a) Direct Multicasting

(b) GSA-Assisted Multicasting

Discussion

- GSA forwarding penalty ~450µs
  - – “not significant for most applications”
- Enhanced security
  - – Localize group key compromise
- Flexible management
  - – Delegation of responsibility
  - – Pricing schemes