Telcordia Technologies

Performance from Experience

Proposed Extensions to Kerberos-Like Shared-Key Methods for Secure Multicast: Secure IGMP and Coalition Support

Internet Draft: **draft-coan-hasm-00.txt**
B. Coan, V. Kaul, S. Narain, and W. Stephens

Brian Coan
coan@research.telcordia.com
Sanjai Narain
Narain@research.telcordia.com
973-829-4489
December 10, 2001

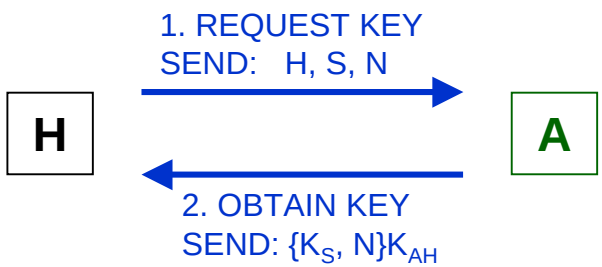
Outline

- Review of current shared-key security for multicast
  - Simple Kerberos-like approach, requiring no router support
  - Basis of GDOI
  - Starting point for Telcordia's security design and implementation
- Secure IGMP
  - Current vulnerabilities (principally denial-of-service attacks)
  - Proposed extensions
- Extensions for coalitions (balance of presentation)
  - Build on existing approach to address the needs specific to dynamic coalitions (limited trust among members, concerns about a single authentication and authorization server, wish for local autonomy in managing authorization policy, scalability, and defense against denial-of-service attacks)
  - Provide similar security functionality to current practice
  - Possible basis for extensions to GDOI or Telcordia implementation

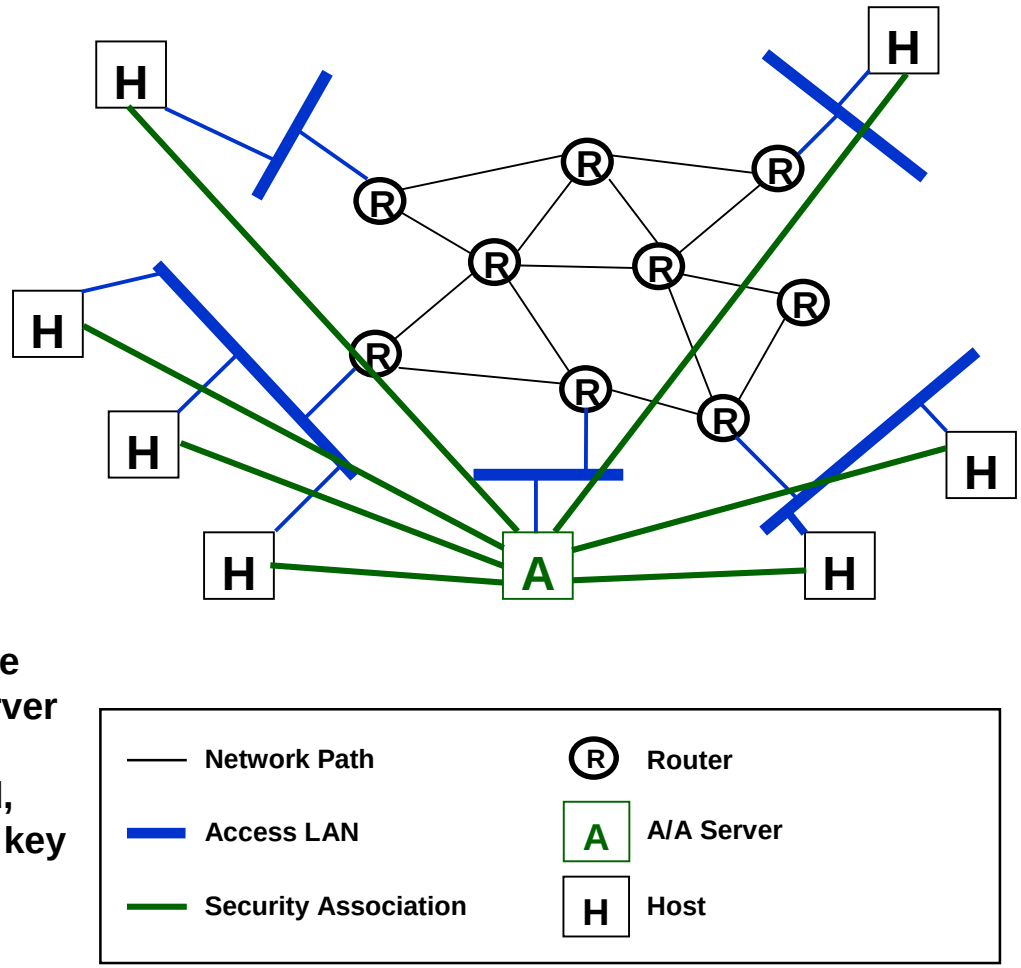
Telcordia Technologies
Performance from Experience

Shared-Key Multicast Security (Current Practice)
Network Picture

Each potential multicast session member must use the existing (green) security association to conduct the following exchange with the authentication and authorization (A/A) server to obtain the multicast session key(s).



The A/A server uses the existing security association to authenticate the potential member. The A/A server then determines authorization. If the potential member is authorized, the A/A server gives it the session key



Telcordia Technologies
Performance from Experience

Shared-Key Multicast Security (Current Practice)
Who Does What

- Need to establish a (phase 1) security association between the authentication and authorization server and each host in the universe of potential multicast session members
- The authentication and authorization server needs to maintain a database that for each multicast session indicates which of the known hosts are and are not authorized members
- The authentication and authorization server must also maintain the current shared key for each multicast session
- For each authorized session member, the authentication and authorization server will give it the current shared multicast session key on demand
- Multicast senders use this key to encrypt data before multicasting
- Multicast receivers use this key to decrypt incoming data (which also provides limited authentication of the sender)

Shared-Key Multicast Security (Current Practice) Rekeying

- The three reasons for rekeying a multicast session
  - On Member Join
    - Rekey to stop new member X from decrypting traffic from before X joined
    - Implementation:
      - Multicast new key to current members encrypted with current key
      - Give new key to X on point-to-point secure connection
  - On Expiration of Multicast key
    - Rekey to limit amount of data encrypted with old key
    - Implementation:
      - Multicast new key to current members encrypted with current key
  - On Member Ejection
    - Rekey to stop departing member X from decrypting new traffic
    - Implementation (two choices):
      - Require each member to obtain new key in a one-to-one interaction with the authentication and authorization server
      - Multicast new key encrypted with an existing key, using some fancy method, like tree-structured keys, to hide the new key from the ejected member

Security Goals
Definitions

- **Confidentiality:** Enemy can't read message
- **Integrity:** Enemy can't change message
- **Authentication**: Claimed sender equals actual sender
- **Access control:** Only authorized parties can participate
- **Dynamic authorization:** Allow change in authorized parties
- **Non-repudiation:** Recipient can prove what messages it got
- **No denial of service:** No interference by un-authorized parties

Telcordia Technologies
Performance from Experience

Satisfaction of Security Goals
Shared-Key End-to-End Kerberos as Baseline

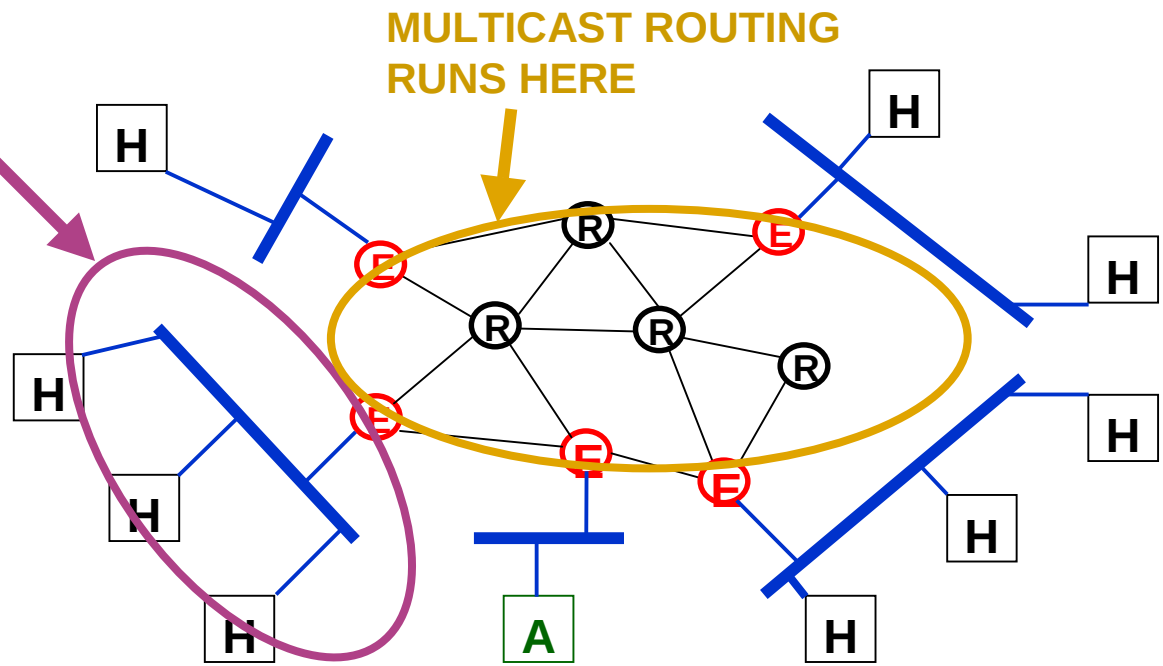
	KERBEROS SHARED KEY	TREE OF SHARED KEYS	MAC CODES OR LOCAL KEY SCOPE	PUBLIC KEYS	SECURE IGMP PROTOCOL
CONFIDENTIALITY	√				
INTEGRITY	√ NEED REPLAY DEFENSE				
AUTHENTICATION	X SEND BY ANY GROUP MEMBER		√	√	√?
ACCESS CONTROL	√				
DYNAMIC AUTHORIZATION	½ JOIN: O(1) LEAVE: O(N)	√ JOIN: O(1) LEAVE: O(log N)			
NON-REPUDIATION	X			√	
NO DENIAL-OF-SERVICE	X				√

DELTA FROM BASIC KERBEROS SHARED KEY

Telcordia Technologies
Performance from Experience

IGMP
Network Picture

Function	IGMP V1	IGMP V2	IGMP V3
QUERY	Yes	Yes	Yes
REPORT	Yes	Yes	Yes
LEAVE		Yes	Yes
Granularity	Group	Group	Group & Source



Telcordia Technologies
Performance from Experience

IGMP
Security Vulnerabilities

- Denial of service
  - Bogus receiver wastes bandwidth by requesting to join a group whose traffic it can't read
  - Bogus receiver disrupts service on LAN by sending IGMP leave messages
  - Bogus sender floods multicast group, wasting bandwidth even if receivers can identify the extra packets as invalid
- Leaking information
  - Some extra exposure to traffic analysis
  - Some extra access to encrypted text for cracking attacks

Telcordia Technologies
Performance from Experience

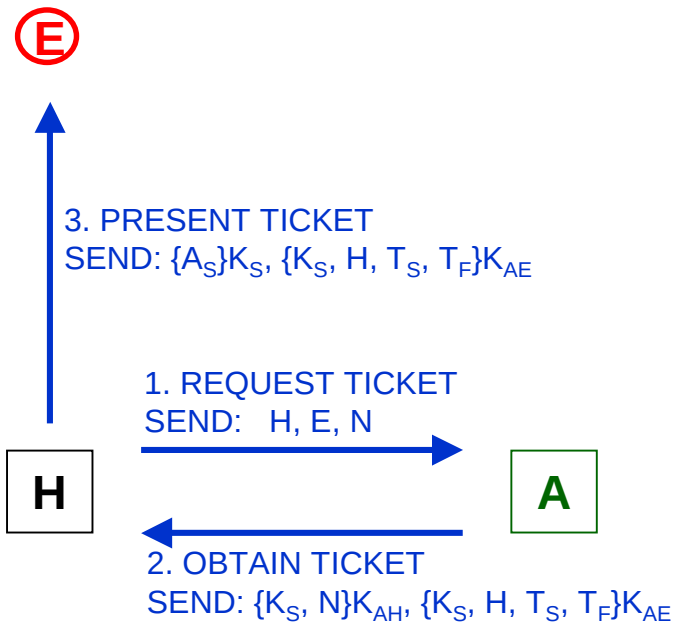
How to Secure IGMP

Telcordia Technologies
Performance from Experience

Secure IGMP

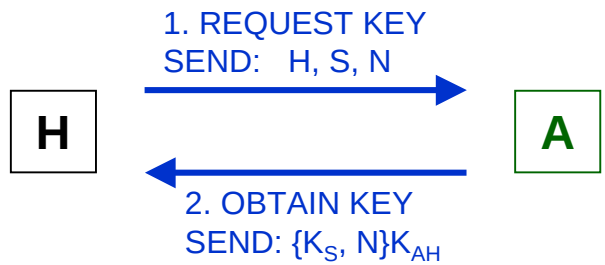
A/A Protocols a la Kerberos (both needed)

Obtaining a Ticket to present to edge router in secure IGMP



Obtaining a Session Key for use with multicast session data

Each session member must do the following:



Key:

H: Hosts
A: Authentication Server
E: Edge Router

Telcordia Technologies
Performance from Experience

Securing IGMP

IGMP currently has no security support…

IGMP Current Functions

- Enables an edge router to know if there exists one or more hosts on its subnet that wish to receive a given multicast group
- Provides edge router with inexact knowledge of which hosts are receiving a given group
- Does not give count of number of recipients
- No security functionality

IGMP Absent Functions

- Enabling receiver authentication to edge router
- Prevent bandwidth use by unauthorized receivers
- Support of authenticated membership requests for senders
- Prevent bandwidth use by unauthorized senders
- Give edge routers exact knowledge of group membership on local subnetwork

Telcordia Technologies

Performance from Experience

Securing IGMP
Four Options

Capabilities	Options for Secure IGMP (details follow)			
	Option 0	Option 1	Option 2	Option 3
End-to-end encryption of multicast session, with key given only to authorized members	√	√	√	√
Receivers demonstrate authentication and authorization to edge routers		√	√	√
Senders demonstrate authentication and authorization to edge routers			√	√
Edge routers have exact knowledge and control of group members on subnet				√

Telcordia Technologies
Performance from Experience

Securing IGMP
Option 0 (Baseline)

Description: No change to IGMP. Rely on multicast session key(s) for security. Each sender and receiver authenticates itself to the A/A server and (if authorized) obtains and uses the session key

- Bogus receivers can obtain traffic to do code breaking and traffic analysis
- Bogus receivers can launch limited DOS attacks by requesting groups for which they are not authorized
- Bogus senders can launch significant DOS attacks by sending garbage traffic
- Hosts on same LAN may share same session key, edge router has inexact knowledge of local group membership
- Simple, no code changes to routers or host IGMP implementations, could be sufficient in some applications

Securing IGMP
Option 1

Description: Receiver must demonstrate to edge router that it is authorized. Edge router accepts report and leave messages only from authorized receivers. Also, Option 0 functionality is provided

- Bogus receivers cannot obtain traffic to do code breaking and traffic analysis UNLESS they are on the same subnet as a legitimate receiver (limitation seems inherent in shared LANs)
- Eliminates spurious traffic requests by bogus receivers
- Eliminates spurious leave messages
- Bogus senders can launch significant DOS attacks by sending garbage traffic
- Hosts on same LAN may share same session key, edge router has inexact knowledge of local group membership

Securing IGMP
Option 2

Description: Sender must demonstrate to edge router that it is authorized. Option 0 and option 1 functionality is also provided

- Because the access LAN is shared, the edge router must authenticate the source of each packet that is sent to a multicast group
- Bogus receivers cannot obtain traffic to do code breaking and traffic analysis UNLESS they are on the same subnet as a legitimate receiver (limitation seems inherent in shared LANs)
- Eliminates spurious traffic requests by bogus receivers
- Bogus senders cannot send garbage traffic on the multicast tree
- Hosts on same LAN may share same session key, edge router has inexact knowledge of local group membership

Securing IGMP
Option 3

Description: All interaction between a sender and an edge router is over an IPsec tunnel. All interaction between a receiver and an edge router is over an IPsec tunnel. Sender must prove to edge router that it has authorization. Receiver must prove to edge router that it has authorization. A/A server still useful for authorization. Also, Option 0 functionality is still provided

- If shared session keys are used, prevents a non-member sender from reading the traffic on this group
- Equivalent benefit regarding senders could be obtained by using per sender session keys
- Care needs to be taken to avoid crypto attacks based on having multiple encryptions of the same content
- Seems to conflict with the use of shared access networks

Telcordia Technologies
Performance from Experience

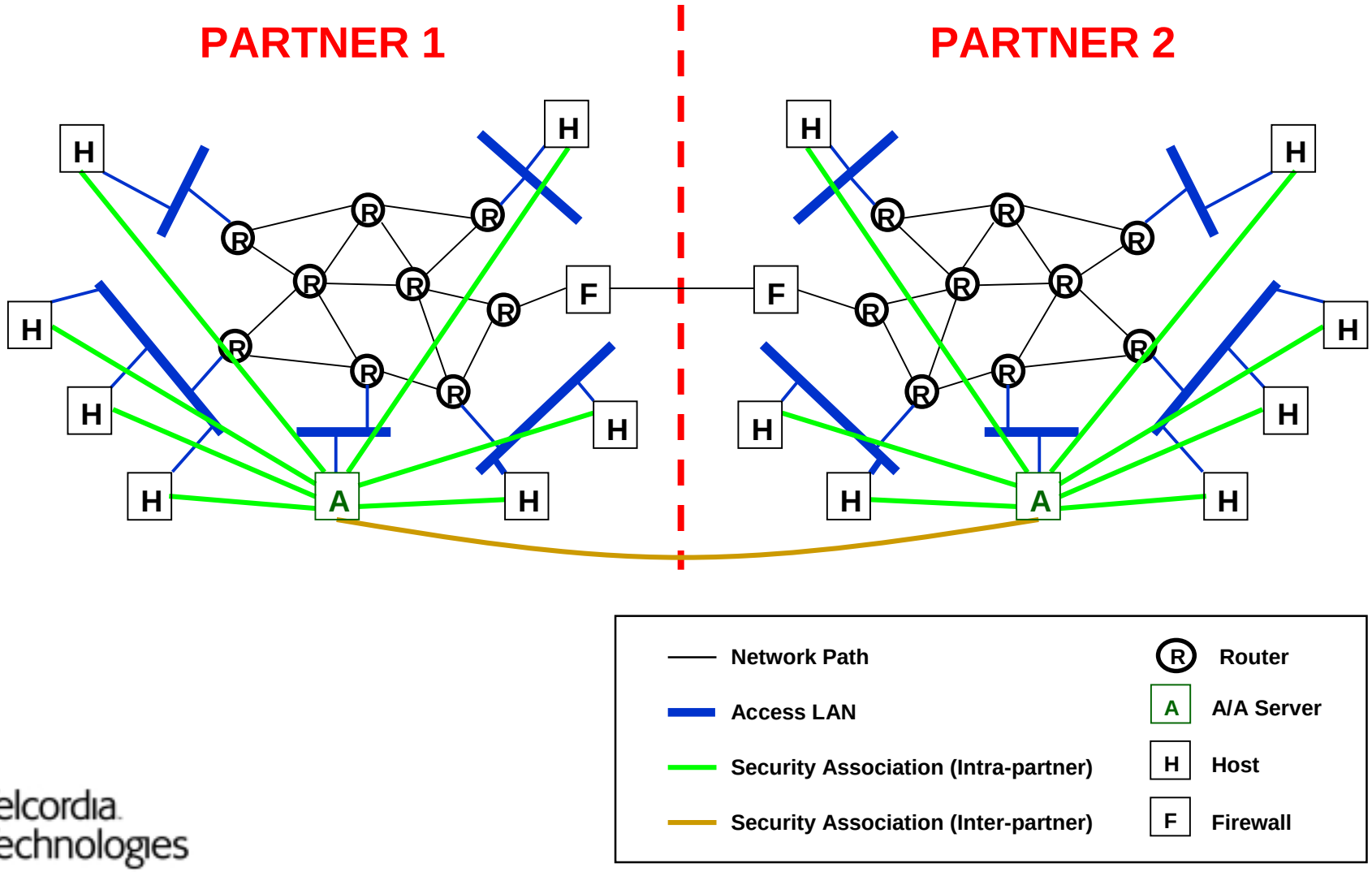
Proposed Extensions to Support Coalitions

Telcordia Technologies
Performance from Experience

Design Goals – Secure Coalition Networking with More Local Autonomy

- Operate with limited router or firewall support (at network borders)
- Use shared and partially shared session keys distributed by A/A (Authentication/Authorization) servers
- *Each coalition partner has its own A/A server*, with limited trust among A/A servers of the coalition partners
  - Each coalition partner manages its own list of authorized session members for each multicast session
  - Each coalition partner authenticates its own members, even in shared sessions
  - Each coalition partner chooses keys for use within its own network, with some re-keying at network borders
- Some multicast sessions are shared among coalition members and some are not
  - A coalition member controls which of its sessions are shared
  - Firewall support may limit scope of non-shared sessions (configuration issue)
- Potential extension to GDOI or Telcordia Hierarchical Application Secure Multicast (HASM) implementation
- Design requires either a lead coalition partner (to generate shared key) or Diffie-Hellman method for A/A servers to agree on session key

Shared-Key Multicast Security (Extension)
Coalition Network Picture



Telcordia Technologies
Performance from Experience

Shared-Key Multicast Security (Extensions)
Initial Security Associations Needed

- Within a coalition partner
  - There must be a security association (shared key) between the authentication and authorization server (for that partner) and each potential multicast participant located within that partner's network
  - These security associations are shown in green in the network picture
- Among coalition partners
  - Each pair of coalition partners must establish a security association (shared key) between their authentication and authorization servers
  - These security associations are shown in mustard in the network picture

Telcordia Technologies
Performance from Experience

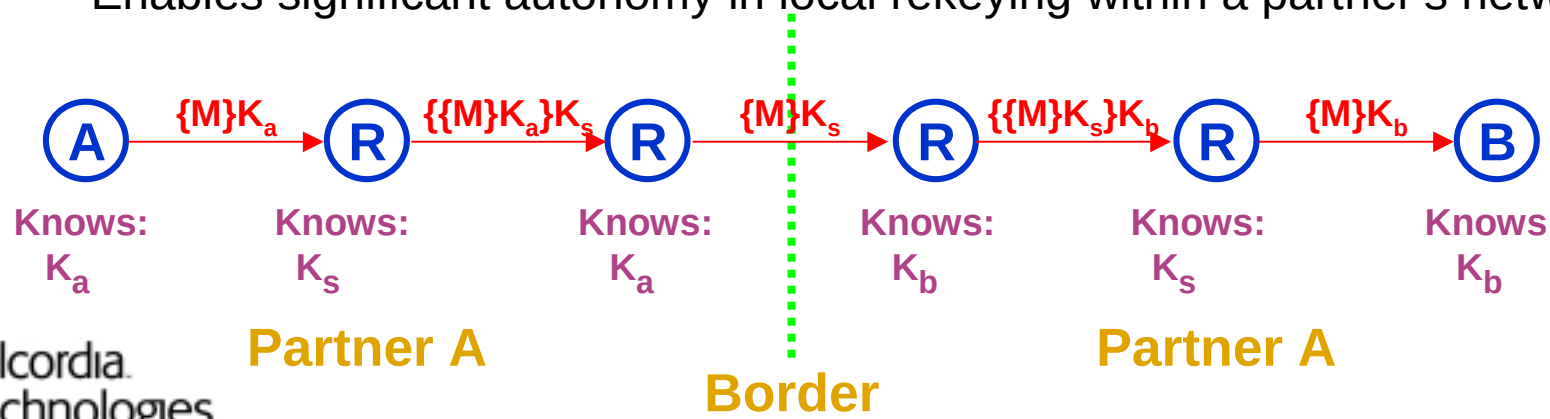
Shared-Key Multicast Security (Extensions)
Behavior of Multicast Session Members

- Go to local authentication and authorization server to obtain the shared key for the multicast session, plus the authentication key
- Use these keys to encrypt and authenticate session messages, authentication is only at granularity of coalition partner
- Accept new multicast session keys that are broadcast with the currently valid multicast session key
- In short, act almost the same as in the un-extended version, except for the added authentication operations

Telcordia Technologies
Performance from Experience

Switching Keys at Administrative Borders in Coalition Networks – Solution 1

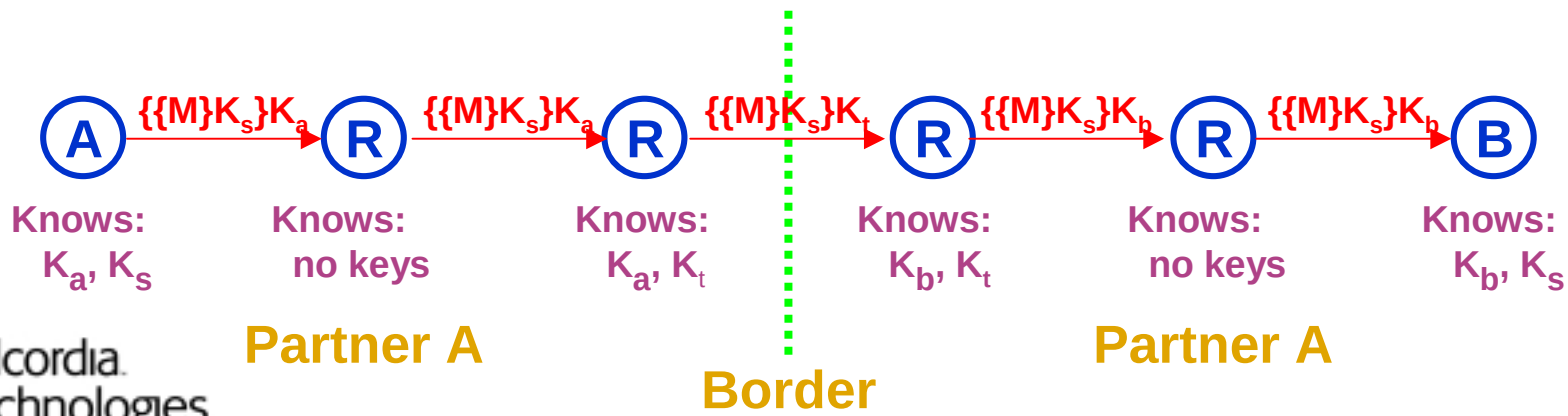
- Requires Commutative Encryption Using Shared Keys
- Enables switch of key within the network, without producing cleartext in the network or having any single point of attack
- Requires a pipeline of two network elements at border to perform the key change
- Coalition partner A uses K_a within its network, coalition partner B uses key K_b within its network, etc. (keys assigned by local A/A server only)
- All coalition partners use K_s for inter-partner traffic (key distribution for K_s is coordinated among all of the A/A servers of all partners)
- Enables significant autonomy in local rekeying within a partner's network



Telcordia Technologies
Performance from Experience

Switching Keys at Administrative Borders in Coalition Networks – Solution 2

- Requires double encryption, but works with any cryptosystem
- Enables switch of key within the network, without producing cleartext in the network or having any single point of attack
- Requires only one network element at border to perform the key change
- Coalition partner A uses K_a within its network, coalition partner B uses key K_b within its network, etc. (keys assigned by local A/A server only)
- Edge routers use K_t for traffic among partners
- All coalition partners use K_s for inter-partner traffic (key distribution for K_s is coordinated among all of the A/A servers of all partners)
- Enables significant autonomy in local rekeying within a partner's network



Telcordia Technologies
Performance from Experience

Shared-Key Multicast Security (Extensions)
Behavior of Multicast Session Members

- Go to local authentication and authorization server to obtain the global (solution 2 only) and local shared keys for the multicast session
- Use these keys to encrypt (only the local key is needed in solution 1, both are needed in solution 2)
- Accept new local multicast session keys that are broadcast with the currently valid local multicast session key
- Accept new global multicast session keys that are broadcast with the currently valid global multicast session key
- In short, act almost the same as in the un-extended version, except for the use of multiple keys

Telcordia Technologies
Performance from Experience

Current Status

- Support for coalitions
  - Implementation available, no Diffie-Hellman
- Secure IGMP
  - Design only

Telcordia Technologies
Performance from Experience

Extra Slides

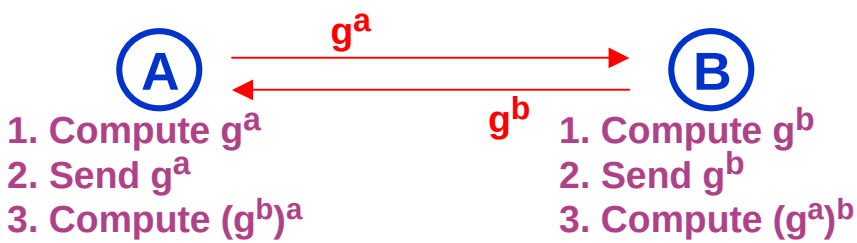
Shared-Key Multicast Security (Current Practice)
Authentication of Multicast Data

- How multicast session members use the session key for encryption and authentication
  - Given message M and session key K_S:
    - Compute $\{M\}K_S$
    - Send $\{M\}K_S$
  - Message can be generated or decrypted by any party with the session key
- Authentication functionality provided:
  - Any session member can authenticate that a message came from an authorized session member, but not which one
- Extend authentication functionality using multiple keys for generating MAC codes (with each key shared by a different subset of members)
  - MAC codes are key-specific secure hashes of the message
  - Given message M and secret authentication key K_A
    - Compute MAC(M, K_A)
    - Send M, MAC(M, K_A)
  - Code can be checked by anyone with the authentication key, furnishing a proof that the message authentication code was generated by someone who knows the particular authentication key

Existing Cryptographic Mechanisms (Primitives Used in Proposed Extensions)

Dynamic Generation of a Shared Key
2-Party Protocol (due to Diffie and Hellman)

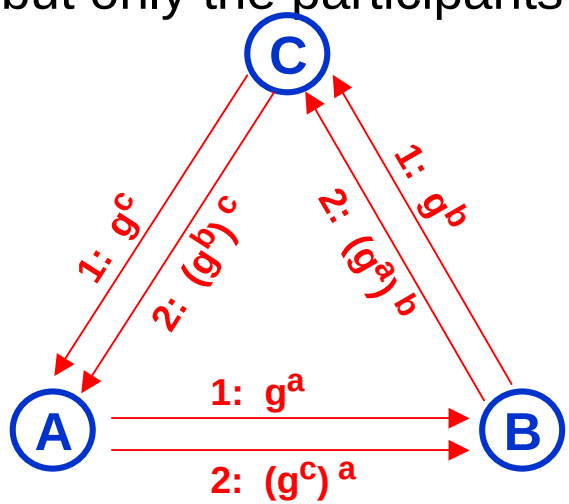
- Required initial information (shared by both parties):
  - *p* is a prime, all computations are done in GF(p), meaning mod *p*
  - *g* is a generator mod *p,* meaning $\forall x \exists y$ such that $x = g^y \bmod p$
- Each participant contributes to the shared key
  - Participant A, picks value *a* and participant B picks value *b*
  - Shared key is $g^{ab} \bmod p$
- Diffie-Hellman messages sent in the clear, but only the two participants learn the shared key
  - Knowledge of *p*, *g*, g^a, and g^b does not enable efficient computation of g^{ab}



Telcordia Technologies
Performance from Experience

Dynamic Generation of a Shared Key
N-Party Protocol (due to Diffie and Hellman)

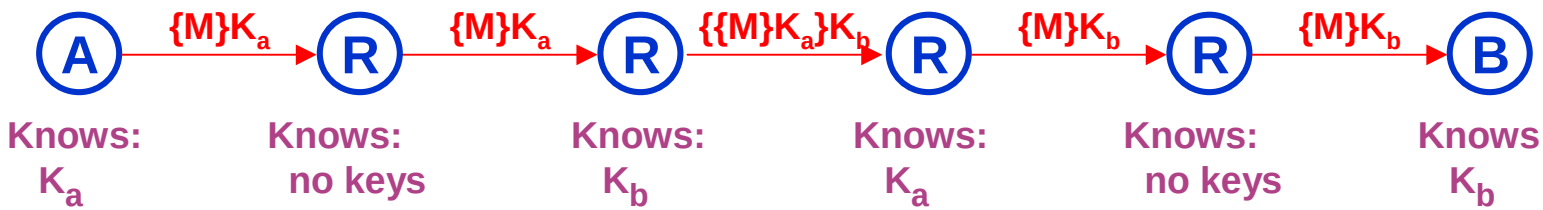
- Required initial information (shared by all parties):
  - *p* is a prime, all computations are done in GF(p), meaning mod *p*
  - *g* is a generator mod *p,* meaning $\forall x \exists y$ such that $x = g^y \bmod p$
- Each participant contributes to the shared key
  - Participant A, picks value *a,* participant B picks value *b,* participant C picks value *c, . . .*
  - Shared key is $g^{abc} \bmod p$ (when there are three participants)
- Diffie-Hellman messages sent in the clear in N rounds of communication, but only the participants learn the shared key



Telcordia Technologies
Performance from Experience

Commutative Encryption Using Shared Keys

- Desired property is $\{\{M\}K_a\}K_b = \{\{M\}K_b\}K_a$
- Encryption algorithm with this property:
  - Use cryptographically secure pseudo-random number generator to produce the pad for one-time pad encryption
  - Encryption commutes because exclusive or (the operation used in one-time pads) commutes
- Use to enable switch of key within the network, without producing cleartext in the network or having a single point of attack



Telcordia Technologies
Performance from Experience

Shared-Key Multicast Security (Extensions)
Changed Role for A/A Servers 1 of 2

- Each server maintains the following information in its Authorization database for each multicast group
  - Which local hosts (i.e. hosts within this coalition member) belong to the multicast group
  - Which other coalition members may have hosts in the multicast group (just the coalition member (e.g. USA) is maintained, not the identity of individual hosts)
  - The identity of the coalition partner who is the lead for the multicast group
    - The lead for a multicast group must be unique
    - The lead for a multicast group must be a authorized to be in the group
  - The current shared key for the multicast group
  - A partner-specific key for authentication (e.g., generating MAC codes)
    - e.g., 5 partners: O(N^2): 25 keys

Telcordia Technologies
Performance from Experience

Shared-Key Multicast Security (Extensions)
Changed Role for A/A Servers 2 of 2

- A valid request for the session key must come either from a host within this coalition partner or from a peer A/A server (in a different coalition partner)
- If the request is from a host in the local network, do the following
  - Authenticate host
  - Determine whether host is authorized
  - If this is the lead partner for the group, return session key and MAC key, encrypted with shared secret
  - If this is not the lead partner for the group, request session key from A/A server from lead partner, return that session key plus local MAC key, encrypted with shared secret
- If request is from a peer A/A server, do the following
  - Authenticate A/A server
  - Determine whether A/A server is authorized and this is the lead partner
  - Return the session key, encrypted with the shared secret

Telcordia Technologies
Performance from Experience

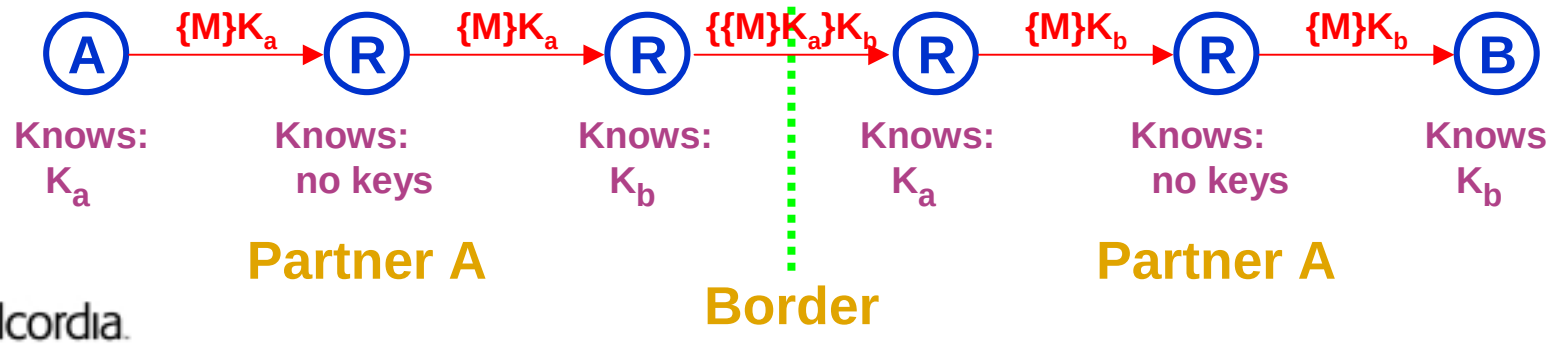
Shared-Key Multicast Security (Extensions)
Network Support

- All of this can operate with no network support in routers and firewalls
- One vulnerability is that it opens multicast groups that should be private to a single coalition member to tampering by other coalition members, including
  - Generating garbage traffic
  - Receiving encrypted traffic
- Improved security can be obtained by *configuring firewalls to pass only that multicast traffic from groups that involve at least two coalition members*
  - e. g., Set policy via service grammar on firewalls
- Following are additional security measures that can be taken with support in some network elements, possibly border routers or firewalls

Telcordia Technologies
Performance from Experience

Switching Keys at Administrative Borders in Coalition Networks – Solution 0

- Requires Commutative Encryption Using Shared Keys
- Enables switch of key within the network, without producing cleartext in the network or having any single point of attack
- Coalition partner A uses K_a within its network, coalition partner B uses key K_b within its network, etc.
- Main deficiency is still that each coalition partner must have knowledge of all keys – deficiency is addressed in the next two solutions
- Note: only portions of the network are shown, R is a router, A and B are hosts



Telcordia Technologies
Performance from Experience

Multicast Security (Solutions 1 &2)
Required Behavior of A/A Servers 1 of 3

- Changed requirements are highlighted in blue
- Within a coalition partner
  - need a security association (shared key) between the local A/A server and each potential multicast participant (host)
  - Need a security association (shared key) between the local A/A server and each network element that must perform either encryption of decryption of multicast traffic
- Each pair of coalition partners must have a security association (shared key) between their A/A servers

Telcordia Technologies
Performance from Experience

Multicast Security (Solutions 1 &2)
Required Behavior of A/A Servers 2 of 3

- Each A/A server maintains the following information in its Authorization database for each multicast group
  - Which local hosts (i.e. hosts within this coalition member) belong to the multicast group
  - Which network elements are authorized to perform encryption and decryption (and for solution 1, which key the element is authorized to use)
  - Which other coalition members may have hosts in the multicast group (just the coalition member (e.g. USA) is maintained, not the identity of individual hosts)
  - The identity of the coalition partner who is the lead for the multicast group
    - The lead for a multicast group must be unique
    - The lead for a multicast group must be a authorized to be in the group
  - The current shared key for the multicast group and the local key for the group (key specific to this partner)

Telcordia Technologies
Performance from Experience

Multicast Security (Solutions 1 &2)
Required Behavior of A/A Servers 3 of 3

- A valid request for the session key must come from
  - A host within this coalition partner
  - A peer A/A server (in a different coalition partner)
  - A network element within this coalition partner
- If the request is from a host in the local network, do the following
  - Authenticate host and verify that host is authorized
  - If this is the lead partner for the group, return global (solution 2 only) and local session keys, encrypted with shared secret
  - If this is not the lead partner for the group, request global session key from A/A server at lead partner, return that global session key plus local session key, encrypted with shared secret (global key needed only in solution 2)
- If the request is from a local network element, do the following
  - Authenticate network element and verify that it is authorized
  - Determine what key the network element is authorized to have, obtain key as above (either locally or from peer A/A server) and return encrypted key
- If request is from a peer A/A server to the lead partner, do the following
  - Authenticate A/A server and verify authorization
  - Return the global session key, encrypted with the shared secret

Telcordia Technologies
Performance from Experience