

Hierarchical Key Management Schemes for Multicasting in Large Wireless Networks

H. Zhao, M. Striki, V. Bharadwaj / John S. Baras

Problem Statement

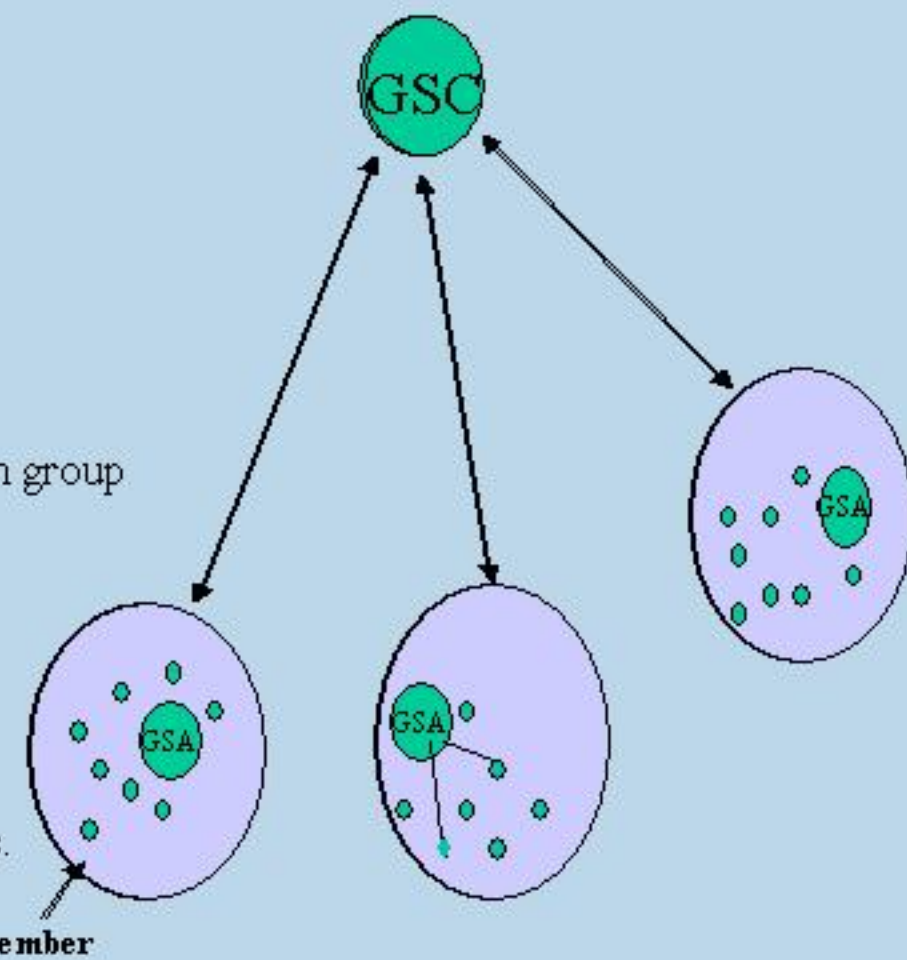
- Objective:**
- Study and design **efficient** and **Secure** Key Management Schemes for Multicasting in Large Wireless Networks.
- Efficient** in that they minimize the total cost deriving from
 - Communication Resources
 - User key storage
 - Computational Complexity
- Principal aim from this trade-off among Communication, Storage and Computation is to achieve a **scalable** system that distributes the key updates reliably to all receivers.
- Secure** in that it can:
 - update the group key securely if receivers join or leave.
 - establish efficiently a shared secret among the legitimate receivers
 - is user collusion free.

Key Management in Large Wireless Multicast Network (design issues)

- Only legitimate group members have access to current group data
- Group key should be changed after every group membership update so that the secure group requirements are fulfilled.
- Group Key Secrecy**
- Forward Secrecy**
- Backward Secrecy**
- Group key should be changed **periodically**.
- For a scalable system overhead involved in key updates, data transmission and encryption must be independent of the size of the multicast group. Addition or removal of a host from the group must not affect all the members of the group ("**1 affects n**" scalability).
- Multicasting in Large Wireless Networks must address the following challenges:
 - Heterogeneous links (variable quality)**
 - Rapidly changing connectivity**
 - Varying mobility levels**
 - Different bandwidth**
 - Physical Layer Hierarchy**
 - Different connectivity modes**
- Design Key Management Schemes that take into account network topology, hierarchy, routing, and predicted and non-predicted member mobility.
- So by using **Hierarchical Key Management Schemes** we can capture all the variations in the network and derive the most efficient scheme in terms of the metrics: **Communication, Storage, Computation**

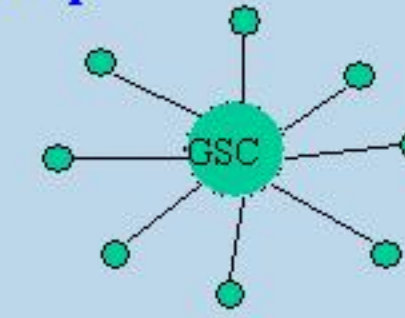
Hierarchical Key Management Model

- GSC:** group security controller
- GSA:** group security agent
- members:** lowest layer
- Independent subgroup key management
- Update limited to subgroup
- One GSC, n_1 GSAs, n_2 members
- Frequency (Probability) of motion from group members/GSAs: p_2 / p_1
- GSC controls GSAs
- GSA communicates keys within its group. When it leaves, new one is selected and group is reconstituted.
- GSAs move in and out of group with different frequencies (lower) than members.
- Members of the same group acquire similar patterns in frequency of moves, and in behavior in general.



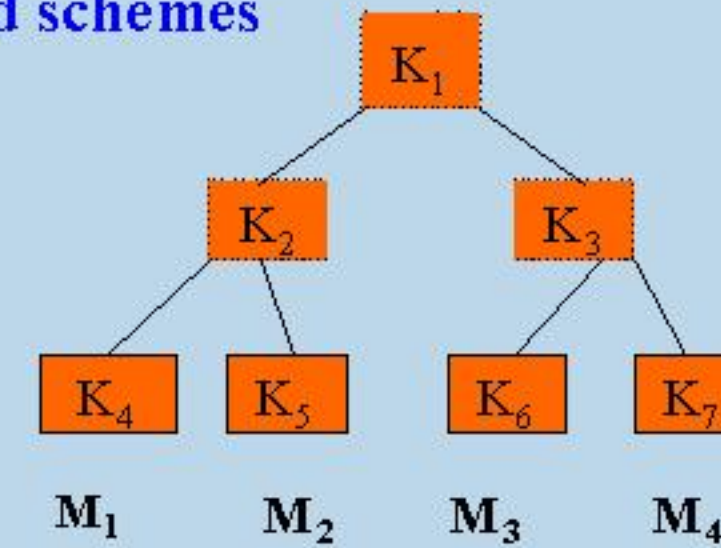
Organization Model Within a Group

- Group Key Management Protocol (GKMP)**
 - Single Group Controller GSA
 - Key pair for member i : $GKMP_i = \{SEK_i, KEK_i\}$
 - Simple, small storage vs. large communication overhead for membership update.
 - Can be used as the organization scheme within each subgroup of GSA in the hierarchical model.
- Tree Based Key Management**
 - Model extends the Logical Tree Key hierarchy
 - Nodes: cryptographic symmetric keys
 - Leaf nodes: group members
 - Each group member knows all keys from its leaf node up to the root node, but no other node in the tree
 - Root Key used as group key, K_G
 - Member joins group: key server authenticates it, assigns it to a leaf node, it receives all keys on the key path to the root, all keys it receives are independent from any previous keys (preserve backward secrecy). Key server replaces all keys on the node member's key path with fresh keys and sends each of these new keys to the group on a "need to know" basis.
 - Member leaves group: all keys leaving member knows will be changed (ensure forward secrecy), keys replaced sequentially from the leaf up to the root key. Leaves are efficient because they only require updating $\log(N)$ keys, N is the number of group members, and assuming balanced tree.



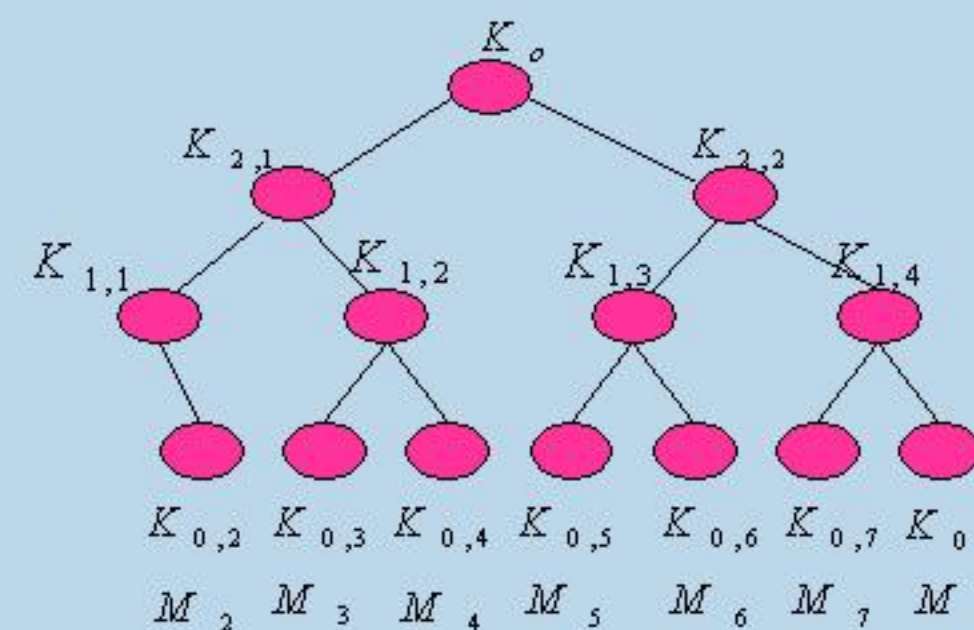
Tree based schemes

- The key path of member M_2 is nodes associated with the keys $\{K_5, K_2, K_1\}$
- New member M_4 joins in empty leaf. Server chooses new keys on the path of M_4 and sends K'_7, K'_3, K'_1 to M_4 over secure link. To update the path server broadcasts the following messages to the group: $\{K'_3\}_{K_6}, \{K'_1\}_{K_3}, \{K'_1\}_{K_2}$
- Member K_3 leaves the group. Key server updates K_1, K_3 and generates new keys: K'_1, K'_3 . It then broadcasts: $\{K'_3\}_{K_7}, \{K'_1\}_{K_3}, \{K'_1\}_{K_2}$



Key update - member 1 leaves:

$$\{\{K_{1,1}, K_{2,1}, K_0\}_{K_{0,2}}, \{K_{2,1}, K_0\}_{K_{1,2}}, \{K_0\}_{K_{2,2}}\}$$



Efficient in communication vs. more keys to store for each member

	SKDC	OFT	ELK
Single member join	Nn	dn	0
Multiple Member join j	jNn	$a_j n$	0
Single member leave	(N-1)n	dn	n_2
Multiple members leave j	(N-j)n	$(a_j - 1) n$	$b_j n_2$

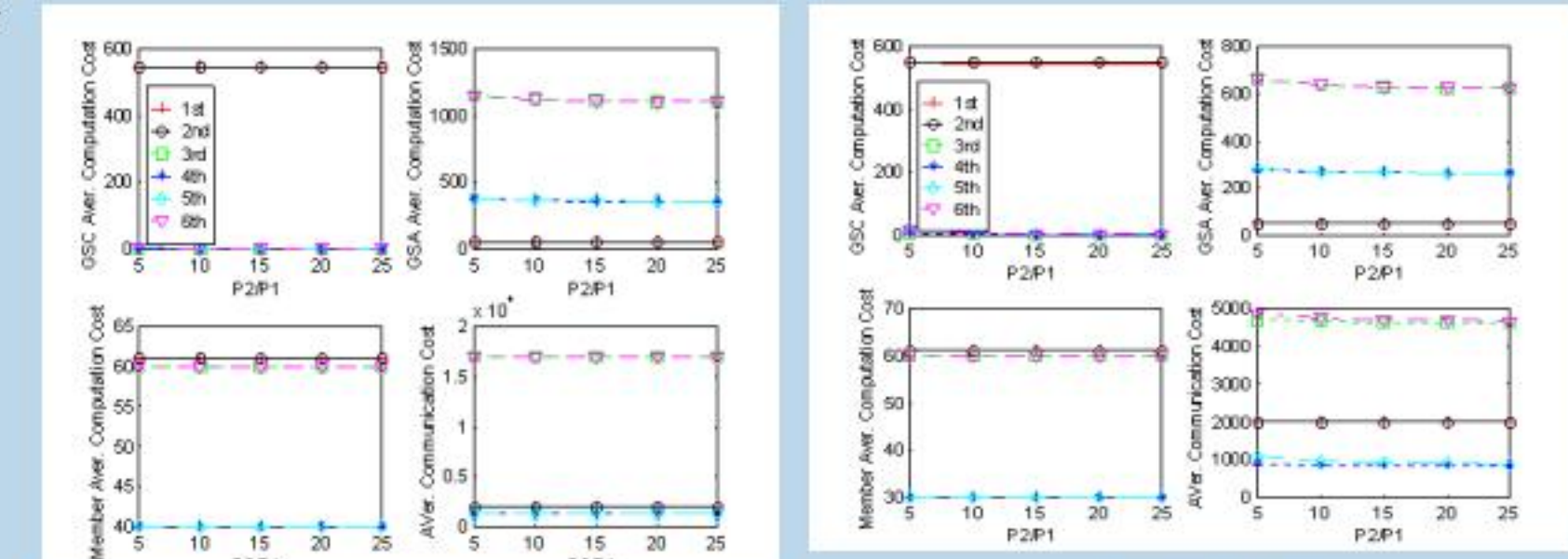
- In the table broadcast size is calculated for each case of some popular protocols.
- j is the number of members joining/leaving the group and a, b, d are some constant numbers, N the number of nodes, n is the length of the key and n_1, n_2 substrings of the key. Observations: in the case of ELK members join does not require any multicast at all, so the protocol is very scalable.

Hierarchical Key Management Schemes: Design

- Single Key Tree, for the two groups of members (different mobilities)
- GSC to GSAs Key Tree, each cluster GKMP
- GSC to GSAs Key Tree, each cluster Key Tree
- Single Key Tree, Single group of members (same mobility)
- Our Goal:**
 - Analyze the metrics: Communication, Storage and computation costs and all their components (parameters).
 - Derive Detailed Objective functions of the metrics which can apply for any of the Hierarchical Schemes in use, and determine via trade-off analysis which scheme is the most efficient after the optimization of all parameters. **Match schemes to resources available.**
- Parameters:**
 - Key Tree parameters (d, h), frequencies of member/GSA motion (p_2 / p_1), length of keys (K), (computation cost for generating a key (C_r), computation cost per PKI encryption/decryption (C_{PE} / C_{PD}), computation cost per symmetric key encryption/decryption (C_{SE} / C_{SD}), communication key updates.
- Current Work**
 - Protocols like ELK (Efficient Large-group Key Distribution), OFT, key distribution over elliptic curves, analyze the computation costs per PKI or per symmetric key encryption/decryption and comm. Key updates. We are working on parametrizations of these costs, and on efficient comparison methods for combinations of protocols.

Parameters:	GKMP (K)	Tree (K')
GSC Storage	$2K$	$(dn-1)K'(d-1)$
Member Storage	$2K$	$(h+1)K'$
Init. GSC Comp.	$(n+1)C_r + nC_{PE} + nC_{SE}$	$(dn-1)C'_r(d-1) + nC'_{PE} + d(n-1)C'_{SE}(d-1)$
Init. Mem. Comp.	$C_{PD} + C_{SD}$	$C'_{PD} + hC'_{SD}$
Init. Comm.	$2nK$	$[n + d(n-1)(d-1)]K'$
Add GSC Comp.	$2C_r + C_{PE} + 2C_{SE}$	$(h+1)C'_r + C'_{PE} + 2hC'_{SE}$
Add Mem. Comp.	$C_{PD} + C_{SD}$	$C'_{PD} + hC'_{SD}$
Add. Comm.	$3K$	$(2h+1)K'$
Del. GSC Comp.	$C_r + (n-1)C_{SE}$	$hC'_r + dhC'_{SE}$
Del. Mem. Comp.	C_{SD}	hC'_{SD}
Del. Comm.	$(n-1)K$	dhK'

Hierarchical Key Management Schemes: Comparisons



- Average Costs when varying r , keeping parameters $C_{PE}, C_{SE}, C_{PD}, C_{SD}$ fixed.

- Analyze effects of these parameters on cost components. No longer fixed. Will depend on:
 - the frequency of members moves
 - on the techniques used for broadcast,
 - on the encryption and decryption mechanisms being used
 - on the bandwidth available for communication,
 - on the power consumption for computation and communication infrastructure
 - on techniques that compress efficiently the information sent during multicast
 - on the frequency of new group sessions taking place,