

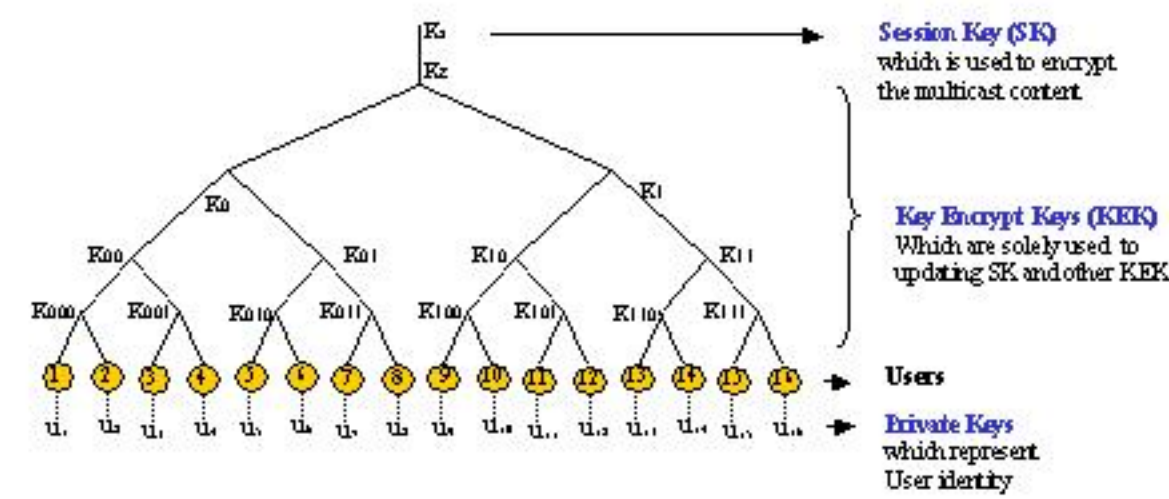
# An Efficient Key Management Scheme for Secure Wireless Multicast

Yan Sun / K. J. Ray Liu

## Introduction and Motivation

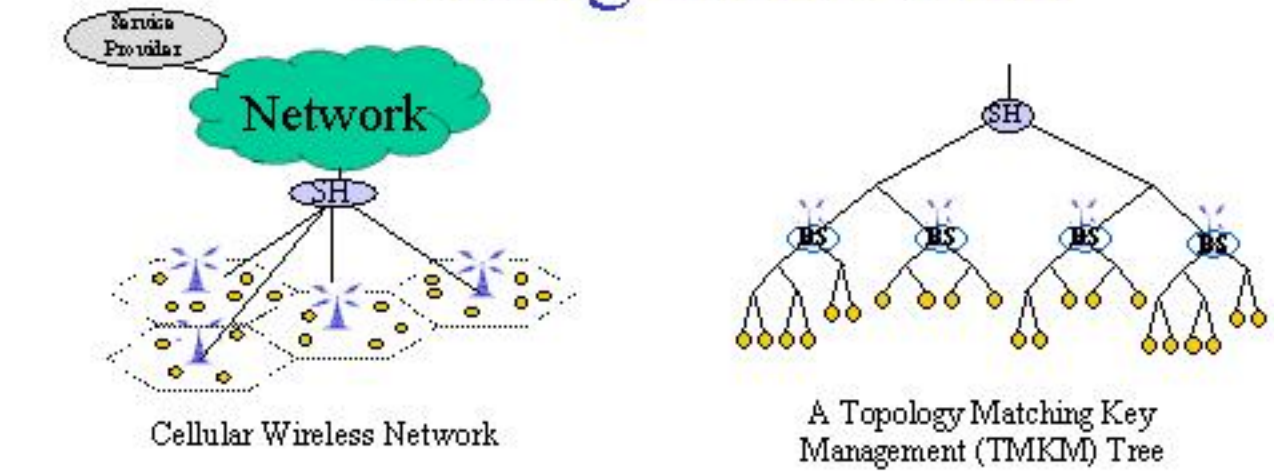
- Many multicast applications require **Access control** such that only authorized users can access the multicast content. Access control is provided by encrypting the content using a key that is shared by all group members.
- Key management** is concerned with generating and updating keys such that only authorized users obtain the valid decryption key.
- For the services with highly dynamic membership and a large number of users, the **reliable** and **timely** delivery of keying information could require a significant portion of network resources.
- Objective : to design a key management scheme that reduces the **communication burden** associated with rekeying.

## Key Management



- When users join or leave the multicast service, keys need to be updated by sending **rekeying messages** in order to prevent the leaving user from accessing the future communication and prevent the joining user from accessing previous communication.
- It is observed that rekeying messages are only useful to a subset of users, who are neighbors on the key management tree.

## Topology Matching Key Management Tree



- We design a key management tree that matches the network topology by
  - Designing a **user-subtree** for the users under each BS.
  - Designing a **BS-subtree** which govern the key hierarchy between the BS's and the SH.

## Topology Matching Key Management Tree (cont.)

- BS's have the knowledge of whether the rekeying messages are useful for their users, and only broadcast the messages when the messages are useful for the users under them.
- Since only a subset of BS's broadcast rekeying messages, we localize the transmission of rekeying messages and therefore reduce the communication burden of rekeying.
- The communication burden of rekeying is described by:

$$C_{wire} = E[\text{the number of messages multicast to the BS's}]$$

$$C_{wireless} = E[\text{the number of messages broadcast by the BS's}]$$

$$C_T = \gamma \cdot C_{wireless} + (1 - \gamma) \cdot C_{wire}$$

## Handoff Schemes for TMKM tree

- Since the physical location of users affects their position on the TMKM tree, users need to be moved from one branch to another branch on the key tree when handoffs occur.
- A simple solution: When a user moves from cell  $i$  to cell  $j$ , he can be treated as if he leaves the service from cell  $i$ , and rejoin the service immediately to cell  $j$ . This scheme is not practical for mobile networks with frequent handoffs.
- An Efficient Scheme: We allow the user having more than one set of valid keys when he stays in the service, and update all of his keys when he leaves the service.

## Performance Analysis

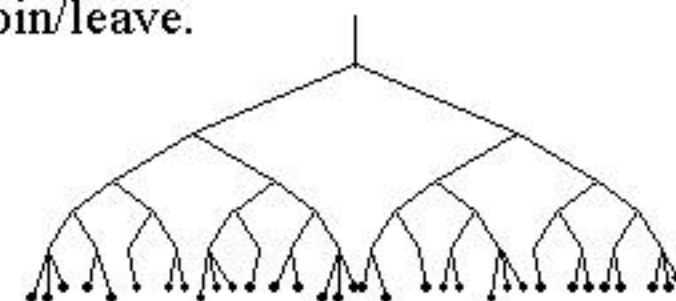
- Matching the key management tree with the network topology has two effects on the communication cost of rekeying:
  - The cost of sending one rekeying message is reduced because only a subset of BS's broadcast the message.
  - The number of rekeying messages is increased because more keys need to be updated when a user leaves due to handoffs.
- The traditional key management trees that are independent of the network topology, shall be called as the **TIKM** trees. The TMKM trees always have smaller wireless cost and larger wire-line cost than the TIKM trees.
- Scalability:  $N$  denote the number of participating BS's, when  $N \rightarrow \infty$

	Wireline Cost	Wireless Cost
TIKM tree	$\sim \log(N)$	$\sim N \log(N)$
TMKM tree	$\sim \log(N)$	$\sim N$

## Subtree Design : ALX tree Structure

- A Challenge of **user-subtree** design : How to maintain the desired properties of the key management tree after users join/leave?
- We design an ALX tree:
  - The tree has totally  $L+1$  levels.
  - The upper  $L$  levels, which compose a symmetric subtree with degree  $a$ , are fixed during the multicast service.
  - Users are attached to the upper nodes randomly in the  $(L+1)^{th}$  level, which changes when users join/leave.

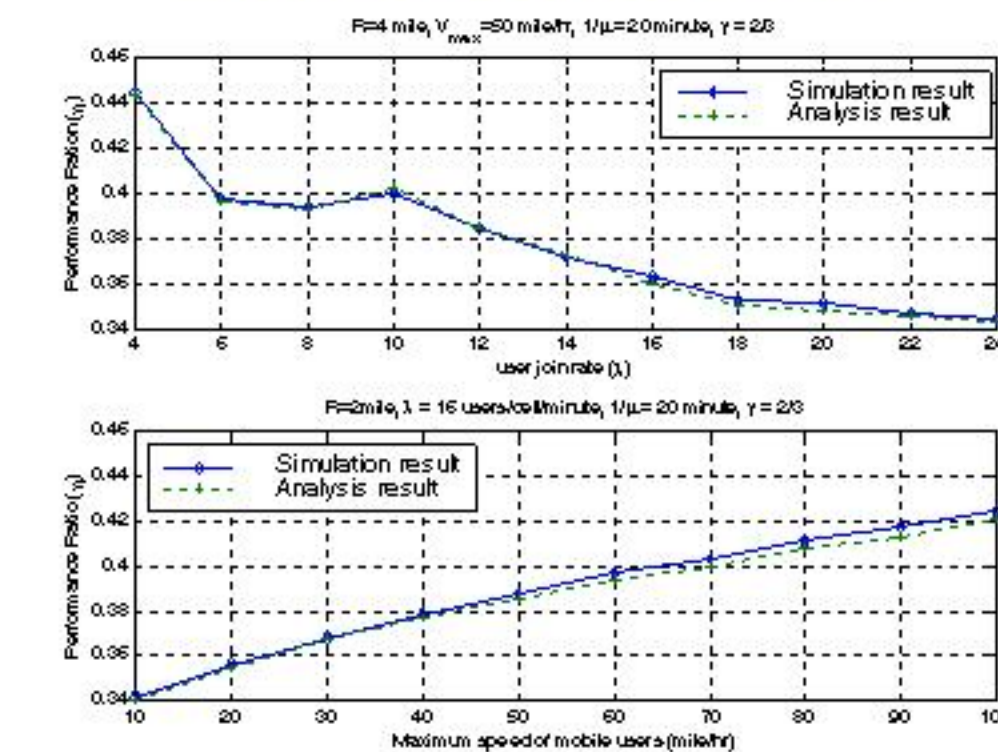
- The performance of the ALX tree is very close to the performance lower bound of the fixed degree trees.



## Optimization

- The TMKM tree consists of
  - user-subtrees, which are designed as ALX trees,
  - and a BS-subtree, which is also designed as an ALX tree.
- We can prove that
  - Optimizing the entire TMKM tree is equivalent to optimizing those subtrees individually.
- Since the optimization problem is separable, the dimension of the searching space for optimal tree parameters is reduced, which would significantly reduce the complexity of tree design.

## Simulation Results



Define the **performance ratio** as :  $\eta = \frac{\text{Total communication cost of TMKM tree}}{\text{Total communication cost of TIKM tree}}$

Compared with the TIKM trees, the TMKM trees reduce the communication cost of rekeying to **33% ~ 45%**.