Explicit Design of Sub-Linear Trees with Pre-specified Key Update Communication Bound

Mingyan Li   Radha Poovendran

Department of Electrical Engineering

University of Washington, Seattle, USA

Work Supported by

NSF Faculty Career Award 00-93187

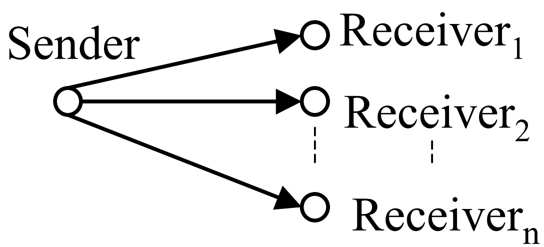
Acknowledgment: Dr. Eric J. Harder

Outline

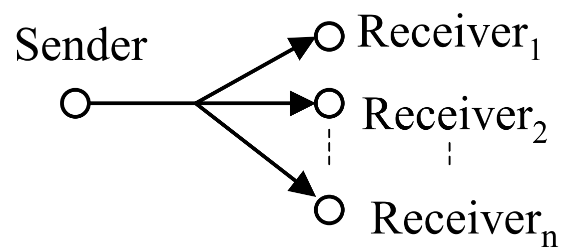
- Motivation
- Key Management Problem
- Logical Key Hierarchy; One-way Function Tree
- Tradeoff between Key Update Communication and Storage
- Hybrid Tree
- Optimization Problem
- Design Solutions and Algorithm
- Numerical Comparison
- Conclusions

Motivation

- Many real-time network applications involve group communications, e.g., video conferencing



Unicast



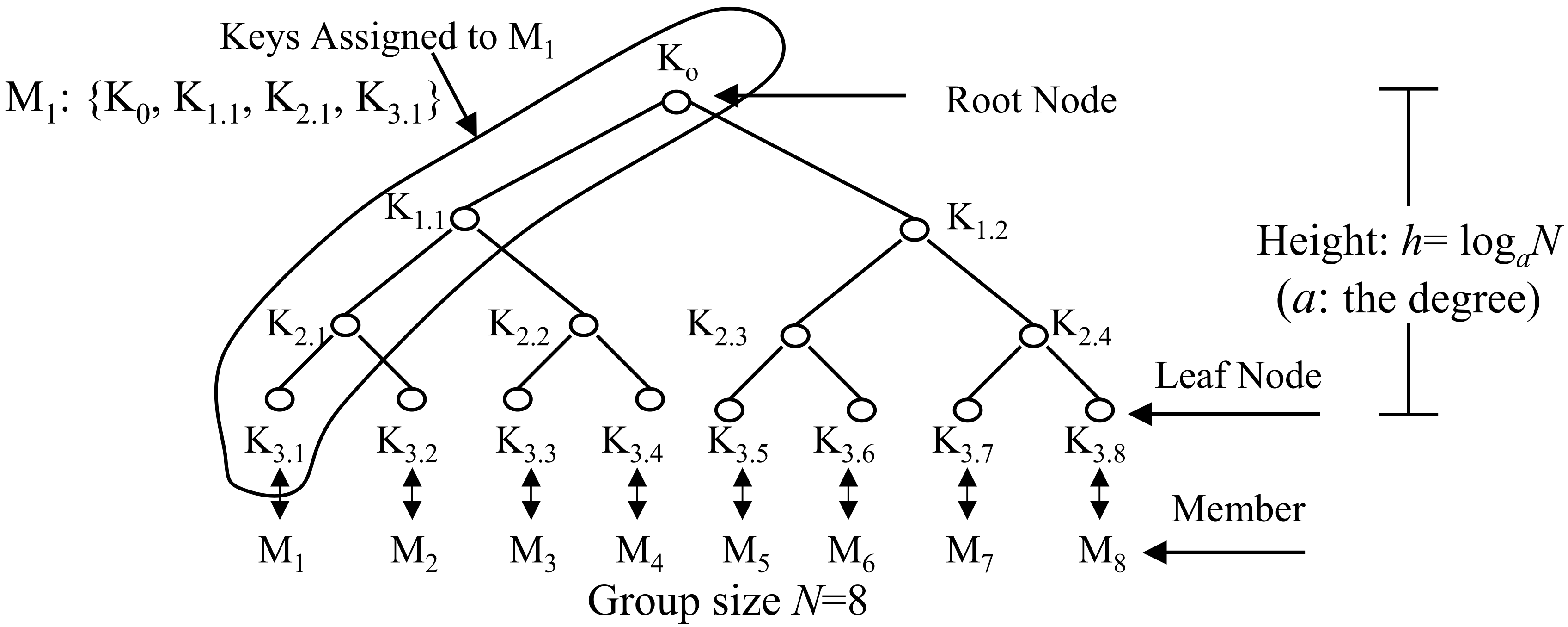
Multicast

- When an identical message is to be sent to multiple receivers, multicast is suitable.
  - Sender resources saved
  - Network resources saved
- Our research focuses on point-to-multiplepoint multicast

Key Management— Review

- Cryptography is one approach to secure multicast.
- Every group memeber shares the same Session Encryption Key (SEK)
- SEK has to be updated whenever there is a change in membership (addition/deletion of member(s))
- Key Encryption Keys (KEK) are used to encrypt and transmit updated SEK to valid members.
- Key management problem
  – How to ensure that only valid members have an access to SEK at any given time instance
- Key management problem reduces to
  – How to distribute KEKs so that all valid members can be securely reached and updated with the new SEK

Logical Key Hierarchy (LKH) (with 8 members)

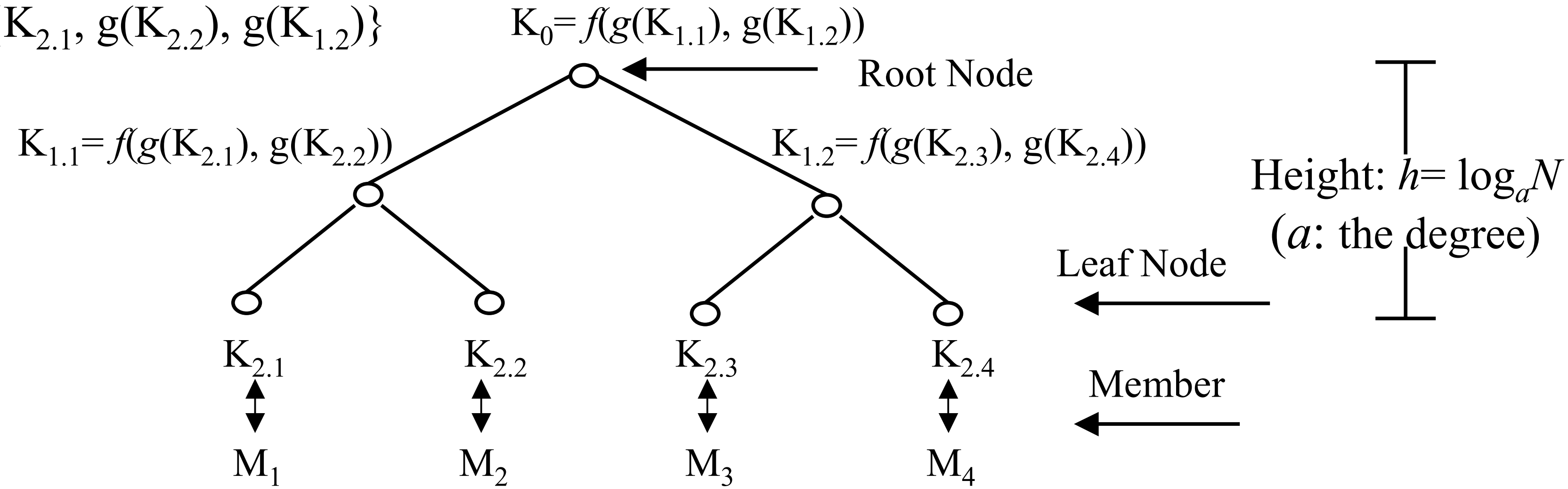


1. Each member is uniquely assigned to a leaf node
2. A KEK is assigned to every node
3. A member is assigned the keys from root to its leaf node

- Storage of KEKs
  - Group controller (GC):
    $$S(LKH) = \frac{aN - 1}{a - 1} \Rightarrow O(N)$$
  - Member: $\log_a N + 1$
- Update Comm. : $a \log_a N$

One-Way Function Tree (OFT) (with 4 members)

Member M_1 is assigned $\{K_{2.1}, g(K_{2.2}), g(K_{1.2})\}$



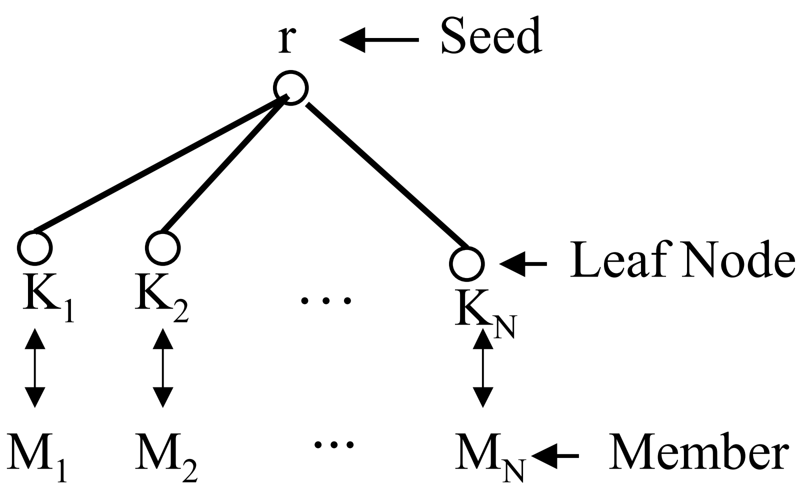
1. Each member is assigned to a leaf node
2. Every node x has two keys, unblinded key K_x and blinded key $K_x' = g(K_x)$
3. Leaf keys: generated by the GC
   Internal keys: $K = f(g(K_{child1}), \ldots, g(K_{childa}))$
4. A member is assigned the unblinded leaf key and the blinded keys of the siblings of nodes in the key path from leaf to root

- Storage of KEKs
  - GC: $S(OFT) = N$
  - Member: $(a-1)\log_a N + 1$
- Update Comm.：
  $(a-1)\log_a N$

LKH and OFT: Similarity and Difference

- Both have a tree structure
  - The height of the tree determines user storage and key update communication related to as $O(\log N)$
- Keys on LKH are independent, while keys on OFT are related by one-way function
  - The GC storage
    - LKH: $\frac{aN-1}{N-1}$, all the keys of a tree are stored
    - OFT: N, only the leaf keys are stored; The storage is independent of the tree degree a
  - Key update communication
    - OFT trades user computation for the reduction in rekey messages by a factor $a/(a-1)$ when compared with LKH

Minimal Key Storage Scheme

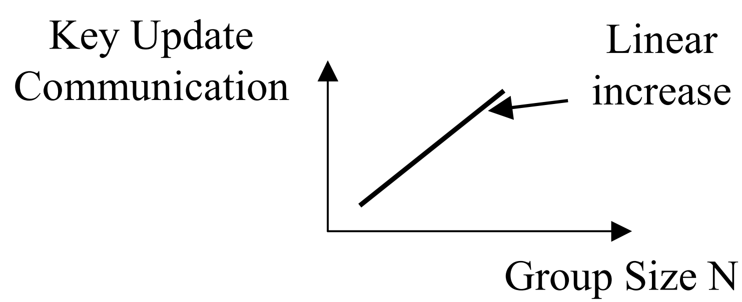


$$K_i = f_r(i)$$

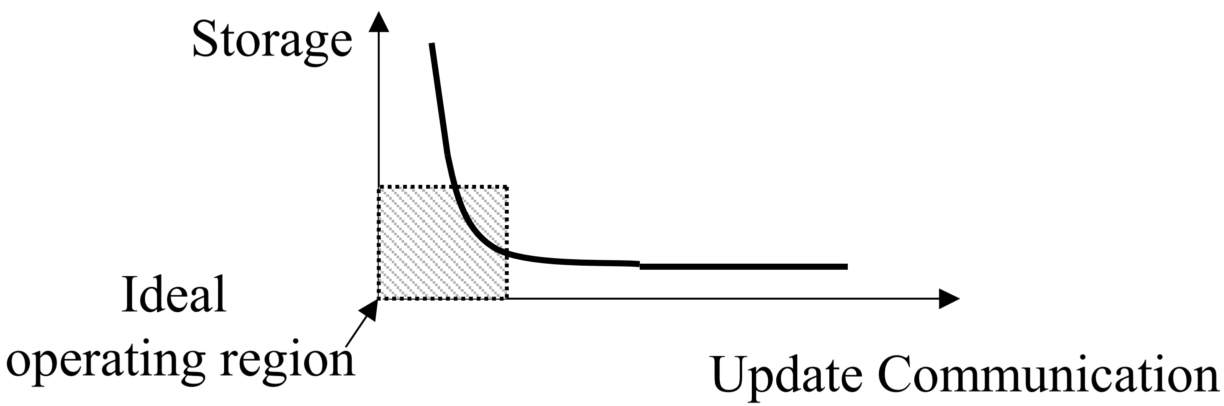
i: member index

N: group size

- Each member shares a KEK with the GC
- Storage
  - GC: $O(2)$ = seed r + SEK
  - Member: 2 keys = KEK + SEK
- Key update communication
  - $(N\text{-}1)$ when a member leave



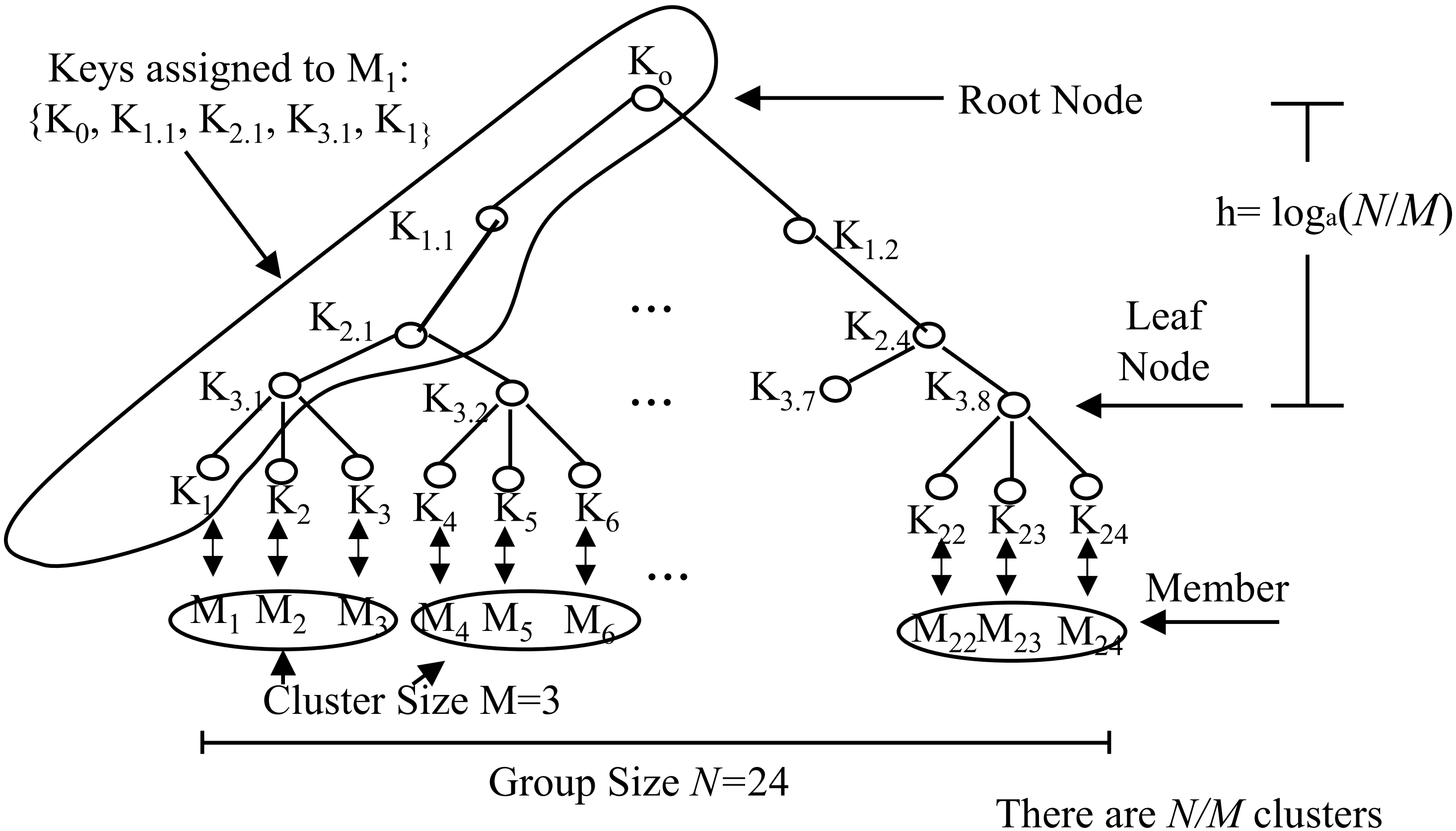
Tradeoff between communication and storage



- We want optimal tradeoff between storage and update communication, under suitable conditions

Hybrid Tree of Canetti

(with 24 members, cluster size of 3)



- Group is divided into clusters of size M
- Each cluster is uniquely assigned to a leaf node
- Within a cluster, a minimal storage scheme is used

Canetti's Examples of Hybrid Schemes

	General M,a	Example 1 $a=2,\ \ M=O(\log N)$	Example 2 $a=M=N^{\frac{1}{2}}$
User Storage	$\log_a\left(\frac{N}{M}\right)+1$	$O(\log N)$	2
Center Storage	$\frac{N}{M}\frac{a}{a-1}$	$O\left(\frac{N}{\log N}\right)$	$N^{1/2}+1$
Communication	$M-1+(a-1)\log_a\left(\frac{N}{M}\right)$	$O(\log N)$	$2N^{1/2}-2$

- To Design a Tree, choose parameters M and a.

Our Formulation

- Given Prespecified update communication value, design an optimal hybrid tree
  —compute the value of cluster size M

Use of Hybrid Tree

- Storage of GC: Storage(tree)+Storage(cluster)
  - LKH-GC: $S(LKH) = \frac{aN-M}{(a-1)M} + \frac{N}{M} = \frac{(2a-1)N}{(a-1)M} - \frac{1}{a-1}$
  - OFT-GC: $S(OFT) = \frac{N}{M} + \frac{N}{M} = \frac{2N}{M}$
- Update communication under member deletion: Comm(tree) + Comm(cluster)
  - LKH-Comm: $C(LKH) = M - 1 + a\log_a \frac{N}{M}$
  - OFT-Comm: $C(OFT) = M - 1 + (a-1)\log_a \frac{N}{M}$
  - Binary OFTs, i.e., a=2, lead to least key update communication for a member addition/deletion
- How to find optimal cluster size *M* to minimize the storage of GC while the update communication is upper *bounded by* $\beta \log N$ with β being a scalar factor?

Optimization of Hybrid Tree

- Optimization problem (structurally LKH and OFT have same form)
- LKH: $$\min_M \frac{(2a-1)N}{(a-1)M}$$ $$s.t. \quad M - 1 + a\log_a \frac{N}{M} \le \beta \log_a N$$
- OFT: $$\min_M \frac{2N}{M}$$ $$s.t. \quad M - 1 + (a-1)\log_a \frac{N}{M} \le \beta \log_a N$$
  where the communication scale factor $\beta \ge 1$
- Storage is a monotonically decreasing function w.r.t. *M*, the largest value of *M* satisfying the communication constraint will be the solution.
- Solve for *M* by $M - 1 + \lambda \ln\left(\frac{N}{M}\right) = \beta \log_a N$

$$\lambda(LKH) = a / \ln a \qquad \lambda(OFT) = (a-1)/\ln a$$

Design Solution of Cluster Size M

- The 1st order approximation of the optimal M

$$M = (\beta - \lambda \ln a) \log_a N + \lambda \log_e((\beta - \lambda \ln a) \log_a N) \qquad (*)$$

$$where \qquad \lambda(LKH) = a / \ln a \qquad \lambda(OFT) = (a-1) / \ln a$$

- Design Procedure
  - Given a, N, and β
  - Compute M using (*)
  - Build hybrid tree of degree a
- The asymptotic storage when M is optimal

$$S(LKH) = \frac{(2a-1)}{(a-1)(\beta - a)} \left(\frac{N}{\log_a N} \right)$$

$$S(OFT) = \frac{2}{(\beta - a + 1)} \left(\frac{N}{\log_a N} \right)$$

Numerical Comparison for LKH

for the case $\beta = \ln a + a$

(Degree a, Group Size N)	# of Keys in GC by LKH	# of Keys in GC by Hybrid LKH	Storage reduction	Comm. increase
$(2, 2^{10})$	2047	444	78.3%	1.7%
$(2, 2^{20})$	2097151	226920	89.2%	13.2%
$(3, 2^{10})$	1535	370	75.9%	3.4%
$(4, 2^{10})$	1365	345	74.7%	1.7%
$(4, 2^{20})$	1398101	176490	87.4%	13.2%

Numerical Comparison for OFT

for the case $\beta = \ln a + a - 1$

(Degree a, Group Size N)	# of Keys in GC by OFT	# of Keys in GC by Hybrid OFT	Storage Reduction	Comm. increase
$(2, 2^{10})$	1024	205	80.0%	31.4%
$(2, 2^{20})$	1048576	104858	90.0%	45.4%
$(3, 2^{10})$	1024	325	68.3%	19.1%
$(4, 2^{10})$	1024	410	60.0%	11.6%
$(4, 2^{20})$	1048576	209715	80.0%	23.9%

Conclusions

- An explicit design procedure for a given communication budget is presented for hybrid tree schemes (hybrid LKH, hybrid OFT).
- The reduction in storage of GC from $O(N)$ to $O(N/\log N)$ is proved.
- More details of our work in CISS'01 handout