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

  Sender → Receiver₁
  ↓      ↓
  |      |
  Receiver₂
  ↓      ↓
  |      |
  Receiverₙ

Unicast

Sender → Receiver₁
↓      ↓
|      |
Receiver₂
↓      ↓
|      |
Receiverₙ

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 member 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 
$\{K_{2.1}, g(K_{2.2}), g(K_{1.2})\}$

$K_1.1 = f(g(K_{2.1}), g(K_{2.2}))$

$K_0 = f(g(K_{1.1}), g(K_{1.2}))$

$K_1.2 = f(g(K_{2.3}), g(K_{2.4}))$

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

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

$$K_i = f_i(i)$$

$i$: member index
$N$: group size

Key Update Communication

Linear increase

Group Size N
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 Size $N=24$
Cluster Size $M=3$

- 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

There are $N/M$ clusters
Canetti’s Examples of Hybrid Schemes

<table>
<thead>
<tr>
<th></th>
<th>General $M,a$</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$a = 2, \quad M = O(\log N)$</td>
<td>$a = M = N^{\frac{1}{2}}$</td>
</tr>
<tr>
<td>User Storage</td>
<td>$\log_a \left( \frac{N}{M} \right) + 1$</td>
<td>$O(\log N)$</td>
<td>2</td>
</tr>
<tr>
<td>Center Storage</td>
<td>$\frac{N}{M} \frac{a}{a-1}$</td>
<td>$O \left( \frac{N}{\log N} \right)$</td>
<td>$N^{1/2} + 1$</td>
</tr>
<tr>
<td>Communication</td>
<td>$M - 1 + (a - 1) \log_a \left( \frac{N}{M} \right)$</td>
<td>$O(\log N)$</td>
<td>$2N^{1/2} - 2$</td>
</tr>
</tbody>
</table>

• 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:** \( \text{Storage(tree)} + \text{Storage(cluster)} \)
  - 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:** \( \text{Comm(tree)} + \text{Comm(cluster)} \)
  - 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 \( \beta \) being a scalar factor?**
Optimization of Hybrid Tree

- Optimization problem (structurally LKH and OFT have same form)

- LKH: \[ \min_M \frac{(2a - 1)N}{(a - 1)M} \]
  \[ s.t. \quad M - 1 + a \log_a \frac{N}{M} \leq \beta \log_a N \]

- OFT: \[ \min_M \frac{2N}{M} \]
  \[ s.t. \quad M - 1 + (a - 1) \log_a \frac{N}{M} \leq \beta \log_a N \]

  where the communication scale factor \( \beta \geq 1 \)

- 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
  \[ M - 1 + \lambda \ln \left( \frac{N}{M} \right) = \beta \log_a N \]

  \[ \lambda (LKH) = a / \ln a \quad \lambda (OFT) = (a - 1) / \ln a \]
Design Solution of Cluster Size $M$

- The 1st order approximation of the optimal $M$

\[
M = (\beta - \lambda \ln a) \log_a N + \lambda \log_e ((\beta - \lambda \ln a) \log_a N) \quad (*)
\]

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

- Design Procedure
  - Given $a$, $N$, and $\beta$
  - 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 $\beta = \ln n + a$

<table>
<thead>
<tr>
<th>(Degree $a$, Group Size $N$)</th>
<th># of Keys in GC by LKH</th>
<th># of Keys in GC by Hybrid LKH</th>
<th>Storage reduction</th>
<th>Comm. increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(2, 2^{10})$</td>
<td>2047</td>
<td>444</td>
<td>78.3%</td>
<td>1.7%</td>
</tr>
<tr>
<td>$(2, 2^{20})$</td>
<td>2097151</td>
<td>226920</td>
<td>89.2%</td>
<td>13.2%</td>
</tr>
<tr>
<td>$(3, 2^{10})$</td>
<td>1535</td>
<td>370</td>
<td>75.9%</td>
<td>3.4%</td>
</tr>
<tr>
<td>$(4, 2^{10})$</td>
<td>1365</td>
<td>345</td>
<td>74.7%</td>
<td>1.7%</td>
</tr>
<tr>
<td>$(4, 2^{20})$</td>
<td>1398101</td>
<td>176490</td>
<td>87.4%</td>
<td>13.2%</td>
</tr>
</tbody>
</table>
Numerical Comparison for OFT
for the case $\beta = \ln a + a - 1$

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