
Trees for Group Key Management with Batch Update

Nan Zang

Ph. D. Defense
May 30th, 2008

Outline

- Secure group key management overview
 - Jumping sequence problem
 - Properties of optimal jumping sequences
 - Properties of optimal GKM trees
 - Approximation algorithms to build GKM trees
 - The GLR Algorithm
 - The LR Algorithm ($q \rightarrow 1$)
 - Summary
-

Introduction

■ Security

- Rekeying Process after each member joins or leaves

■ Efficiency

- Batch rekeying process ('01 Li, Yang, Gouda, Lam)
 - Reassign the keys after a period of time (several joins/leaves)
 - Less secure (set the length of the period based on application)
 - More efficient

Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary



Introduction

- **Group Key Management (GKM) Problem**
 - Benefit from IP Multicast (Big Problem: Control the access)
 - Two Tasks
 - Identification and authorization
 - **Maintain the group keys**
 - A popular topic, but less theoretical analysis
- **One specific model('03 Zhu, Chan, Noubir)**
 - Network service has limited resources
 - Only can serve n members
 - Each member in a batch period has probability p of leaving
 - There are always members waiting to join

Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary



Introduction

■ Model definition

- n members in the group
- Each member has the same probability p of leaving.
- The update cost = the expected number of rekeying-messages that have to be sent by the key server.
- Task:

Minimize the update cost in each batch period

Introduction

Jumping Sequence
Problem

GLR Algorithm

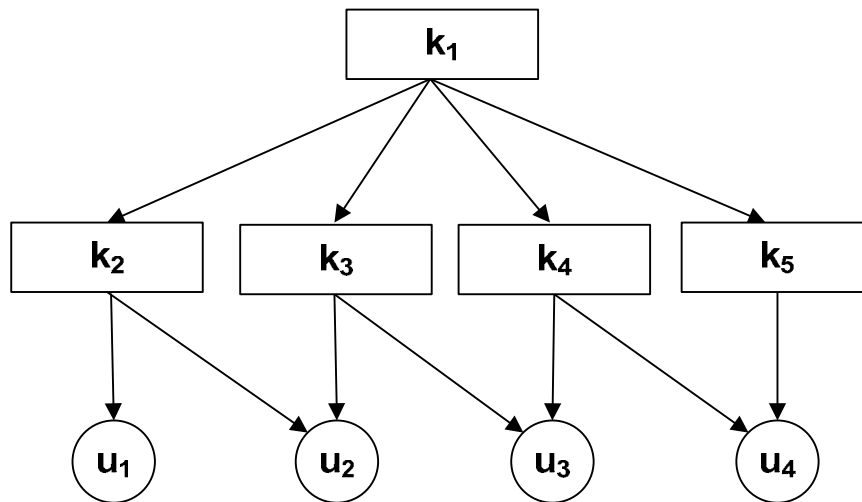
LR Algorithm

Summary



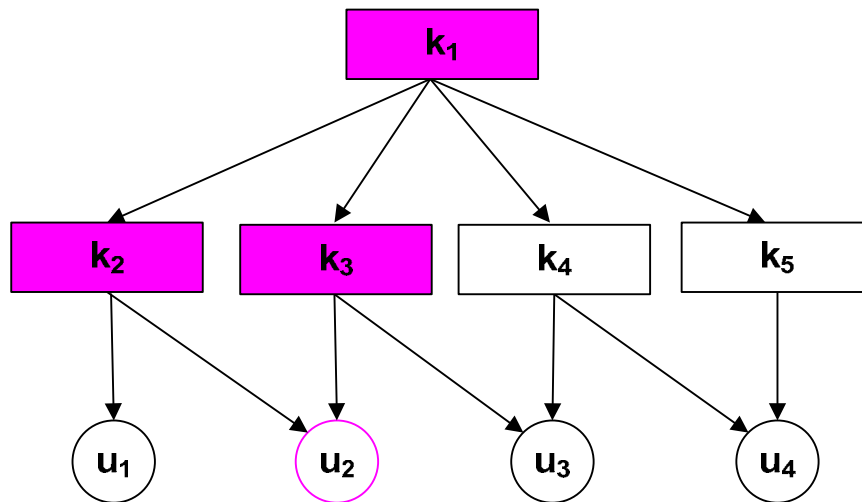
Key graph

- Key graph ('98 Wong, Gouda, Lam)
 - DAG -- Directed Acyclic Graph
 - Key - user relation
 - Users (u-node): All the nodes with only incoming edges.
 - Keys (k-node): All the other nodes.
 - User U_i contains Key $K_j \Leftrightarrow$ There is a directed path from K_j to U_i .



Key graph

- Key graph ('98 Wong, Gouda, Lam)
 - DAG -- Directed Acyclic Graph
 - Key - user relation
 - Users (u-node): All the nodes with only incoming edges.
 - Keys (k-node): All the other nodes.
 - User U_i contains Key $K_j \Leftrightarrow$ There is a directed path from K_j to U_i .



Introduction

Jumping Sequence
Problem

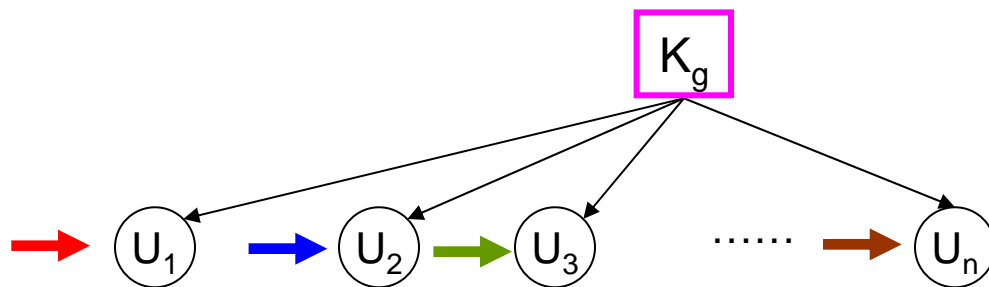
GLR Algorithm

LR Algorithm

Summary



n-star tree example



$$E_{K_i}\{K_g'\}$$

K_g' is the plain message.

K_i is the encryption key.

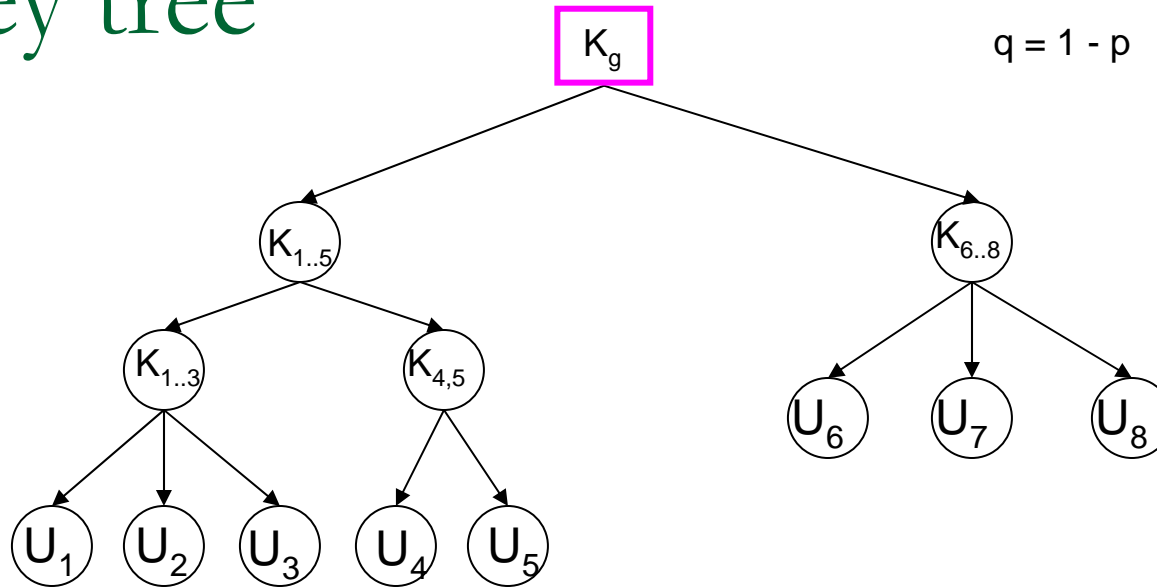
- If the member U_1 changes, the group key K_g has to be changed.
- K_g is not secure. To tell every user the new K_g , n different messages have to be broadcast, each is encrypted new K_g by user's private key K_i .
- Each member has the probability p of leaving.

Probability (K_g changes) = $1-(1-p)^n$ The cost of the n-star = $n(1-(1-p)^n)$

- **Problem: No common keys other than K_g .**
- Is there any better way to assign the keys to the users?

Key tree

p : the prob. of each member changing
 $q = 1 - p$



Hierarchical key structure

users - keys

Manage users in subgroups

One key for each subgroup

Key tree

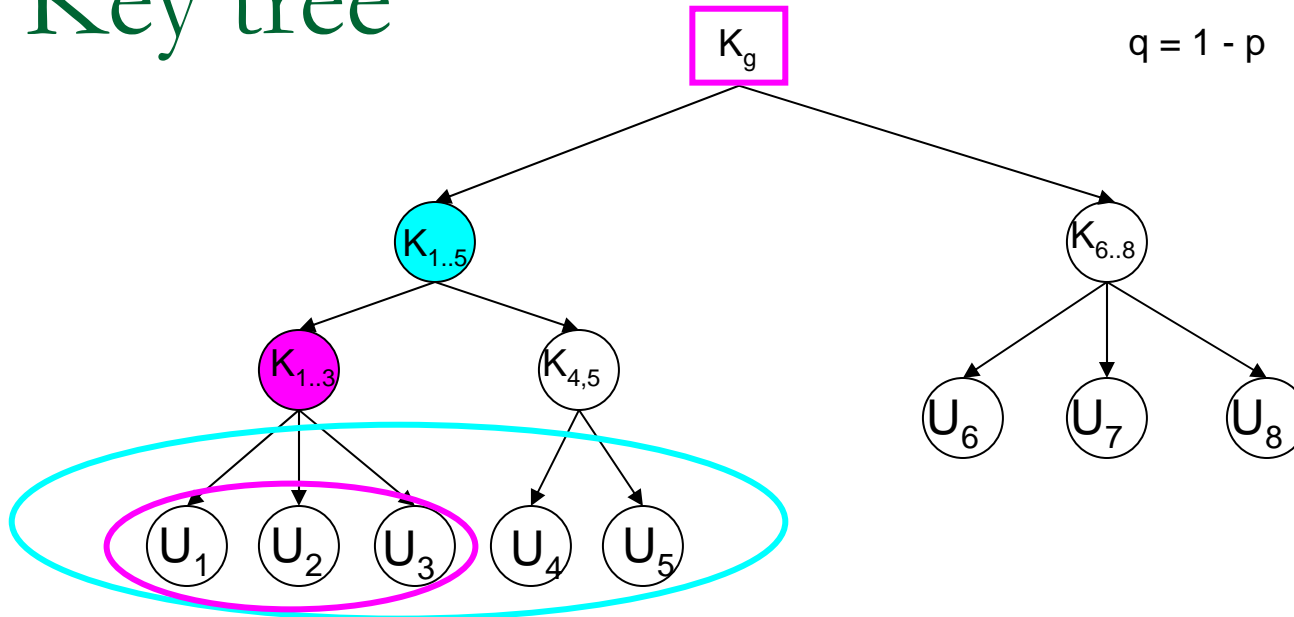
leaves (users); internal nodes (keys)

Subtrees

The root of each subtree

Key tree

p : the prob. of each member changing
 $q = 1 - p$



Subgroup

Subgroup Key

$\{U_1, U_2, U_3\}$

$K_{1..3}$

$\{U_1, U_2, U_3, U_4, U_5\}$

$K_{1..5}$

K_g : Group Key; Traffic Encryption Key (TEK)

$K_{1..3}, K_{4..5}, K_{1..5}, K_{6..8}$: Subgroup Keys; Key Encryption Key (KEK)

Subgroup key changes if and only if any member in this subgroup changes.

Introduction

Jumping Sequence
Problem

GLR Algorithm

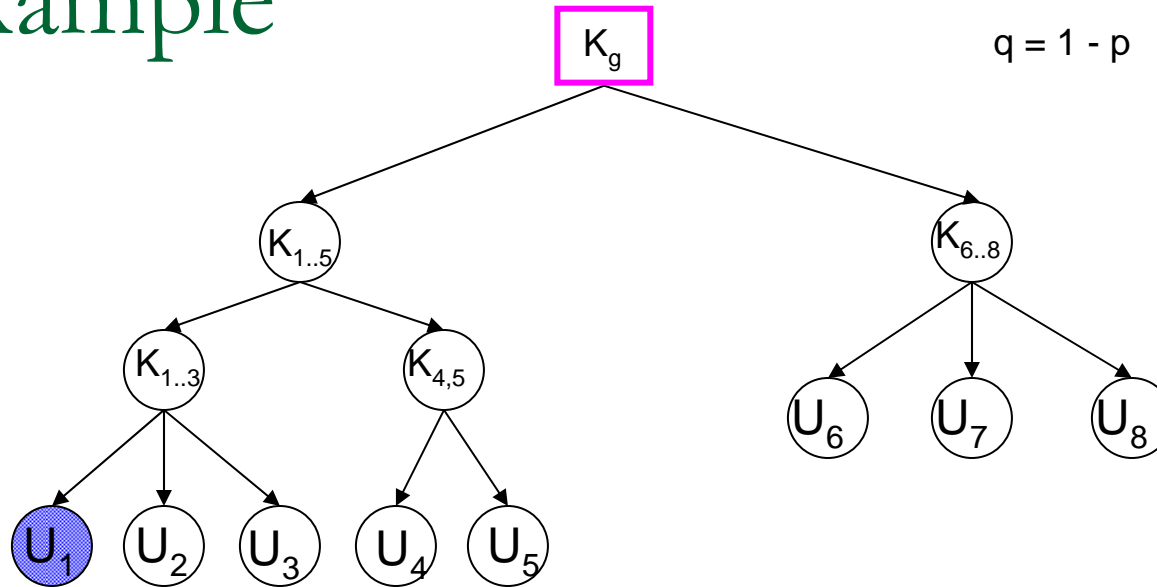
LR Algorithm

Summary



Example

p : the prob. of each member changing
 $q = 1 - p$



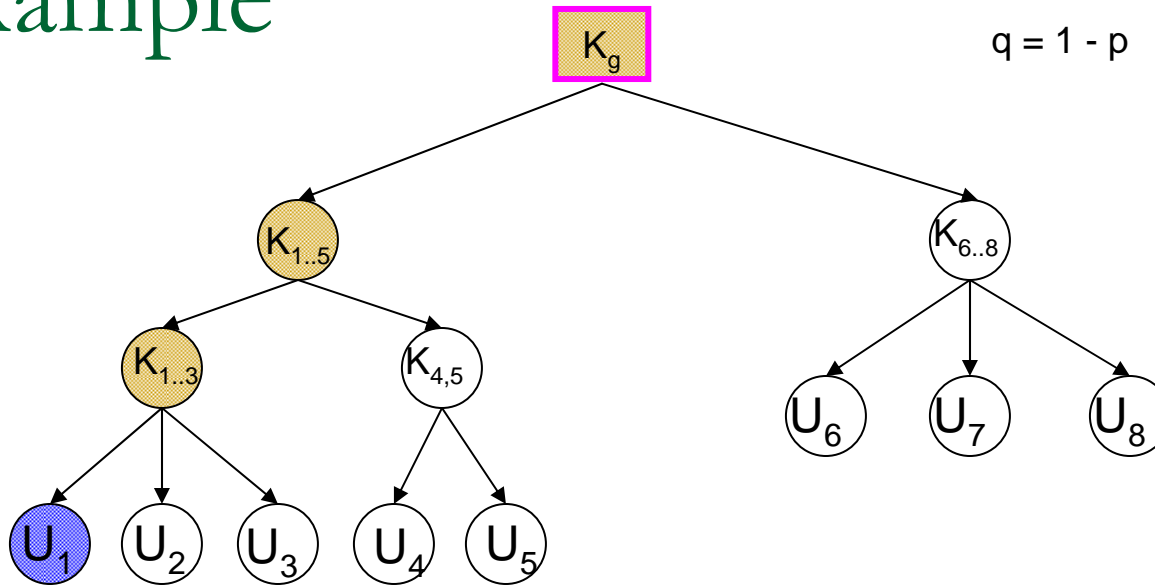
Example:

If member U_1 changes, how to update the keys?

1. Update all the keys U_1 holds.
2. Update the keys from a bottom-to-top order.

Example

p : the prob. of each member changing
 $q = 1 - p$



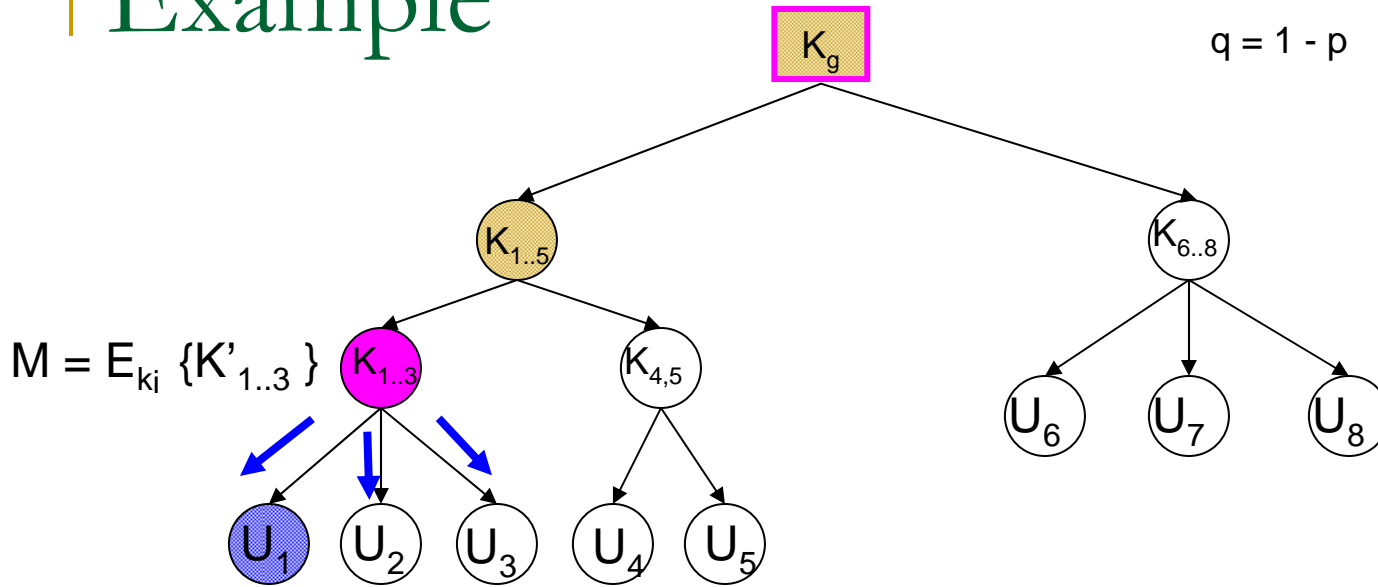
Example:

If member U_1 changes, how to update the keys?

1. Update all the keys U_1 holds: $\{K_{1..3}, K_{1..5}, K_g\}$
2. Update the keys from a bottom-to-top order.

Example

p : the prob. of each member changing
 $q = 1 - p$



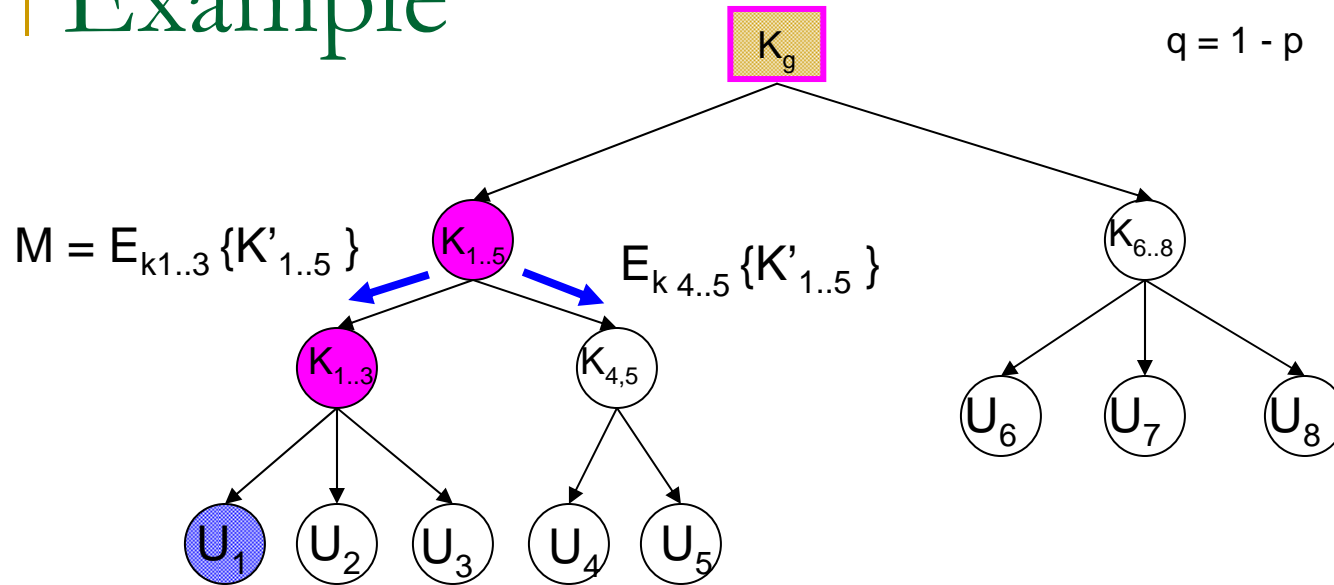
Example:

If member U_1 changes, how to update the keys?

1. Update all the keys U_1 holds: $\{K_{1..3}, K_{1..5}, K_g\}$
2. Update the keys from bottom to top.

Example

p : the prob. of each member changing
 $q = 1 - p$

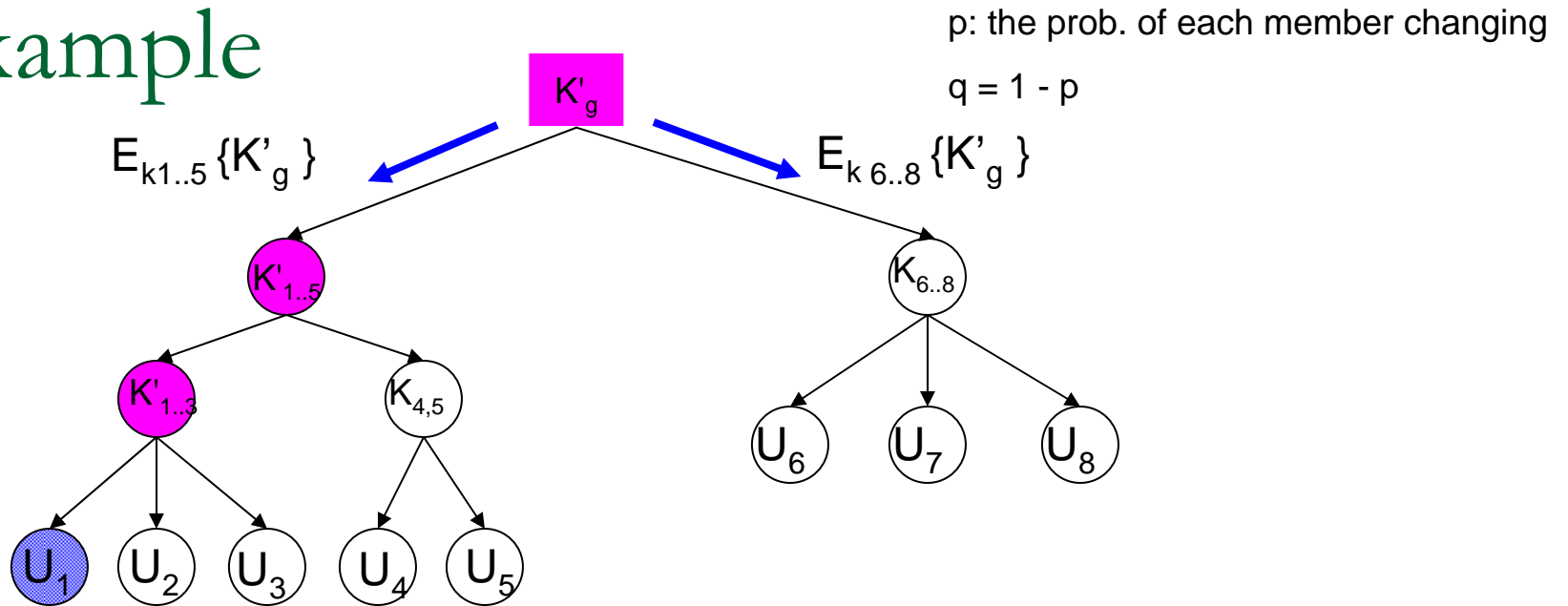


Example:

If member U_1 changes, how to update the keys?

1. Update all the keys U_1 holds: $\{K_{1..3}, K_{1..5}, K_g\}$
2. Update the keys from bottom to top.

Example

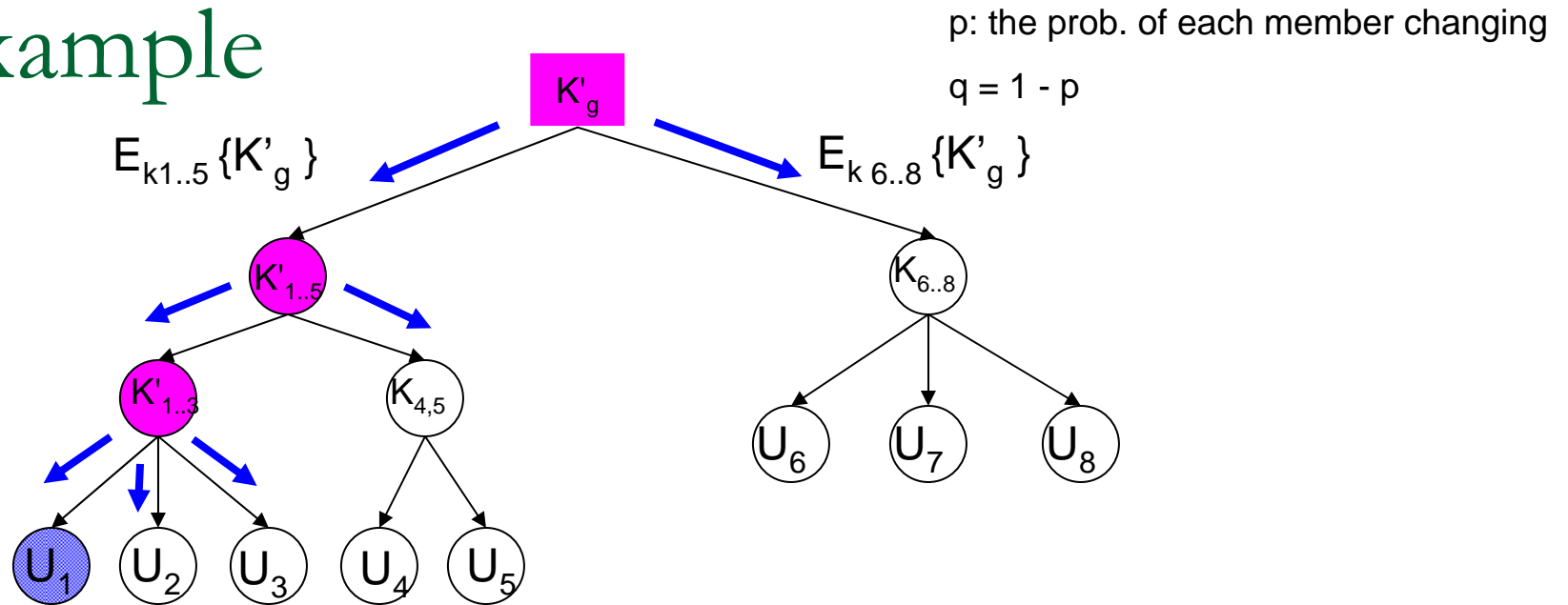


Example:

If member U_1 changes, how to update the keys?

1. Update all the keys U_1 holds: $\{K_{1..3}, K_{1..5}, K_g\}$
2. Update the keys from bottom to top.

Example



Example:

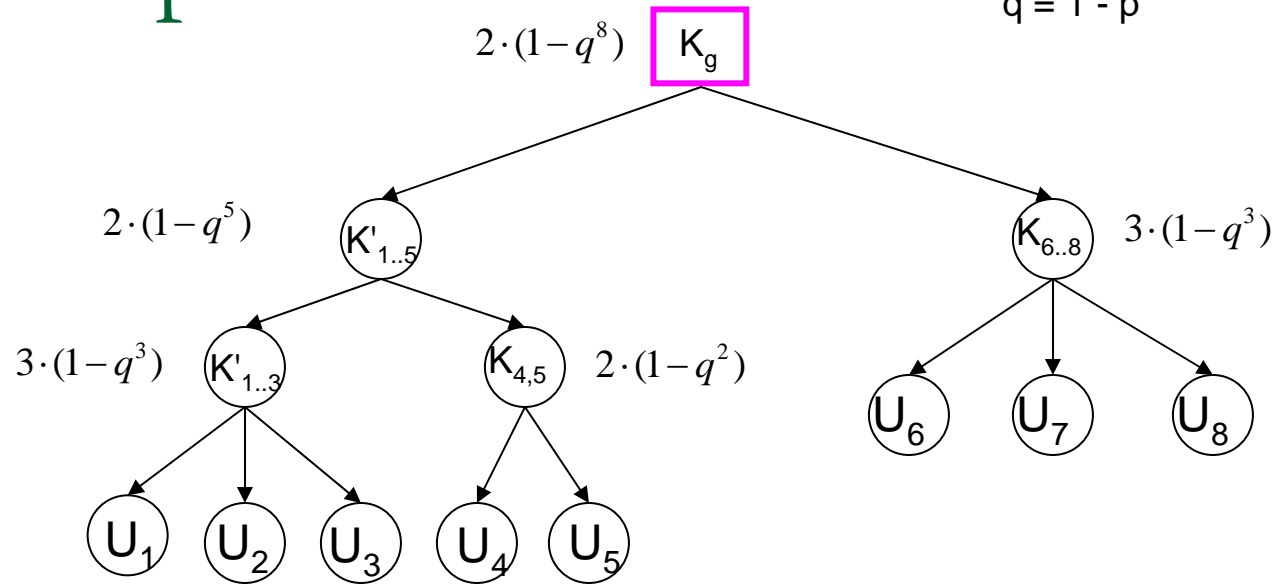
7 Messages have to be sent!

If member U_1 changes, how to update the keys?

1. Update all the keys U_1 holds: $\{K_{1..3}, K_{1..5}, K_g\}$
2. Update the keys from bottom to top.

Example

p: the prob. of each member changing
 $q = 1 - p$



Subgroup Key	Probability of Changing	No. of Messages
$K_{1..3}$	$1 - (1 - p)^3 = 1 - q^3$	3
K_v	$1 - (1 - p)^{N(v)} = 1 - q^{N(v)}$	$d(v)$

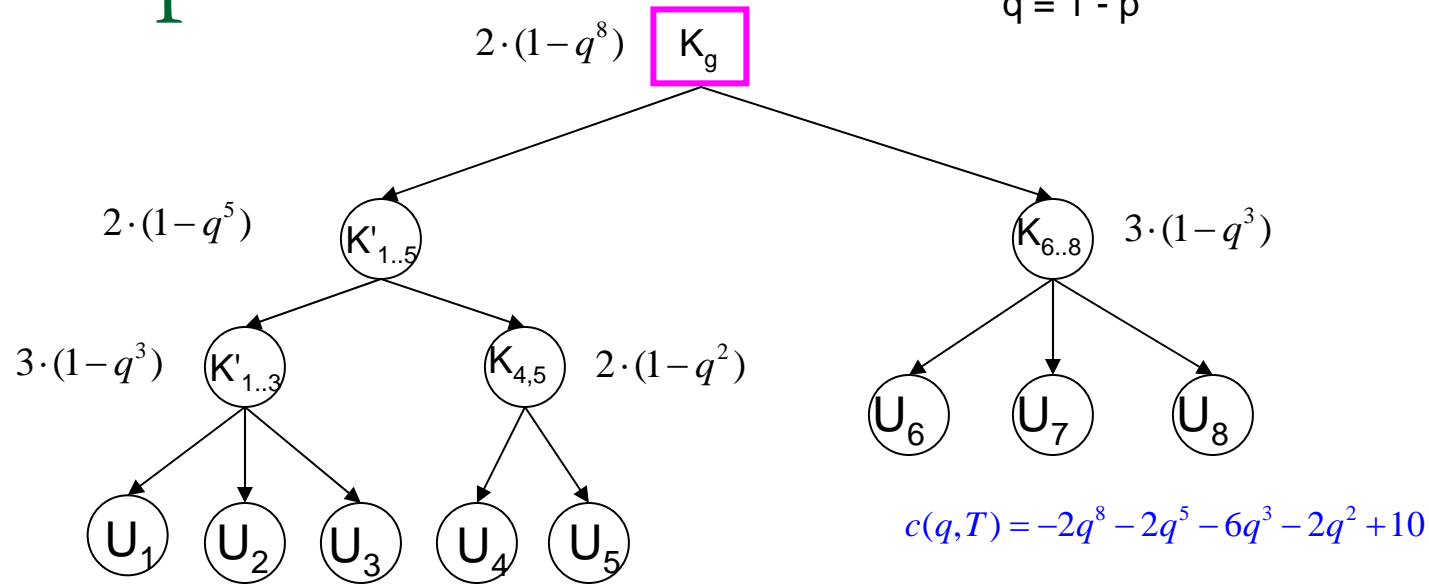
N_v is the number of leaves under the key node K_v .

d_v is the outgoing degree of the key node K_v .

For each K_v , Exp No. of Messages to update $K_v = d(v) \cdot (1 - q^{N(v)})$

Example

p: the prob. of each member changing
 $q = 1 - p$

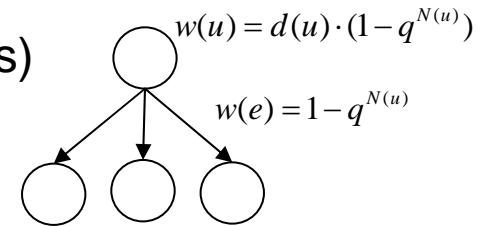


For any tree structure, assuming each leaf has probability p of changing.

Cost of the tree = Expected number of encrypted rekeying-messages.

(Expected number of encryptions needed to reassign the keys)

$$\text{Cost of Tree: } C(q, T) = \sum_{u \in V} d(u) \cdot (1 - q^{N(u)}) = \sum_{e \in E} (1 - q^{L(e)})$$



$d(u)$ = degree of u ; $N(u)$ = number of leaves under u ; for $e = (u, v)$, $L(e) = N(u)$

Introduction

Jumping Sequence Problem

GLR Algorithm

LR Algorithm

Summary



Mathematical Model

Math Model

- Given n , and n // n is the number of leaves
- q , where $q = 1-p$,
- cost of tree structure T

$$C(q, T) = \sum_{u \in V} d(u) \cdot (1 - q^{N(u)})$$

- Objective:
Find a tree T with minimum cost among all the trees with n leaves.

GKM Problem

- n Members in a group
- Each member changes with probability p .
- Expected number of rekeying messages to be sent.
- Objective:
How to arrange the members to minimize the expected number of rekeying messages.

Contributions

- Jumping sequence problem
 - Properties of optimal jumping sequences
 - Properties of optimal GKM trees
 - Approximation algorithms to build GKM trees
 - Properties of optimal trees as $n \rightarrow \infty$
 - The GLR Algorithm
 - GKM problem as $q \rightarrow 1$
 - Properties of optimal jumping sequences as $q \rightarrow 1$
 - The LR Algorithm
-

Definitions of Jumping Sequences

■ Definition (Jumping Sequence Problem)

➤ $S_n = (1 = a_0, a_1, a_2, \dots, a_{k-1}, a_k = n)$, $1 = a_0 < a_1 < \dots < a_k = n$, a_i is a real number.

➤ The jump from a_{i-1} to a_i costs $\frac{1 - q^{a_i}}{a_{i-1}}$

➤ Cost of S_n is: $c(q, S_n) = \sum_{i=1}^k \frac{1 - q^{a_i}}{a_{i-1}}$

➤ Find the best sequences from 1 to n with the minimum cost.

Note: Jumping Sequence Problem is the real number version of GKM problem, and $\text{opt}(q, n) \geq n f(q, n)$.

$f(q, n)$ denotes the minimum cost from 1 to n and $F(q, n)$ denotes the corresponding optimal real number sequences. .

$\text{OPT}(q, n)$ denotes the optimal GKM tree and $\text{opt}(q, n)$ denotes the minimum cost.

Introduction

Jumping Sequence Problem

GLR Algorithm

LR Algorithm

Summary



Properties of Jumping Sequences

■ Proof Outline

➤ Recursive Property:

- If $1 < a_1 < a_2 < \dots < a_{k-1} < n$ is an optimal jump sequence for q and n , then for any $1 \leq i \leq k$, the sequence $1 < a_{i+1}/a_i < a_{i+2}/a_i < \dots < n/a_i$ is an optimal jump sequence for $q' = q^{a_i}$ and $n' = n/a_i$.
- If an optimal jump sequence starts $1 < b \dots < n$, then $b < 4.75$.
- If an optimal jump sequence starts $1 < a < b \dots < n$, then $a \geq \sqrt{2}$, and $b \geq 2$.

Introduction

**Jumping Sequence
Problem**

GLR Algorithm

LR Algorithm

Summary



Properties of Jumping Sequences

- If $n \geq 2$, then

$$f(q, n) \geq 1 - q^{\sqrt{2}}, \text{ and } \text{opt}(q, n) \geq n \cdot (1 - q^{\sqrt{2}}).$$

$f(q, n)$ denotes the minimum cost from 1 to n .

$\text{opt}(q, n)$ denotes the cost of the optimal GKM tree.

- For any n , k is the number of jumps from 1 to n , then

$$k \geq \min(0.64 \ln(n) - 0.04, -0.64 \ln(\ln \frac{1}{q}) - 0.45),$$

$$k \leq \min(2.89 \ln(n), -2.89 \ln(\ln(\frac{1}{q})) - 0.89).$$

Introduction

**Jumping Sequence
Problem**

GLR Algorithm

LR Algorithm

Summary



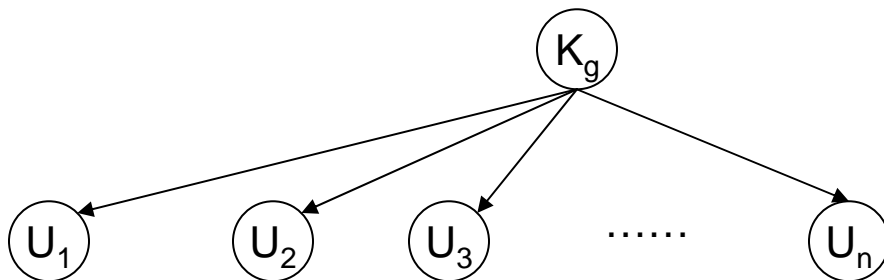
Outline

- Secure group key management overview
 - Jumping sequence problem
 - Properties of optimal jumping sequences
 - Properties of optimal GKM trees
 - Approximation algorithms to build GKM trees
 - The GLR Algorithm
 - The LR Algorithm ($q \rightarrow 1$)
 - Summary
-

Related work ('07 Graham, Li, Yao)

■ Optimal GKM trees

- When $0 \leq q < 3^{-1/3}$, the n-star is strictly better than any other tree structure. ($3^{-1/3} \approx 0.69336$, $q = 1-p$)



The cost of an n-star:

$$n \cdot (1 - q^n)$$

- We only consider $3^{-1/3} \leq q \leq 1$.

Introduction



Jumping Sequence Problem



GLR Algorithm



LR Algorithm



Summary



Idea of GLR Algorithm

- First, we show there is always a “good” and “symmetric” subtree T_{DS} connected to the root, when n is large.
 - The size of the T_{DS} is bounded;
 - The structure of T_{DS} only depends on q .
- Second, for a fixed n and q , to build an approximate GKM tree structure, we use T_{DS} as often as possible.
- Computer simulation shows GLR is efficient. The approximation ratio of GLR is still open.

Introduction



Jumping Sequence
Problem



GLR Algorithm



LR Algorithm



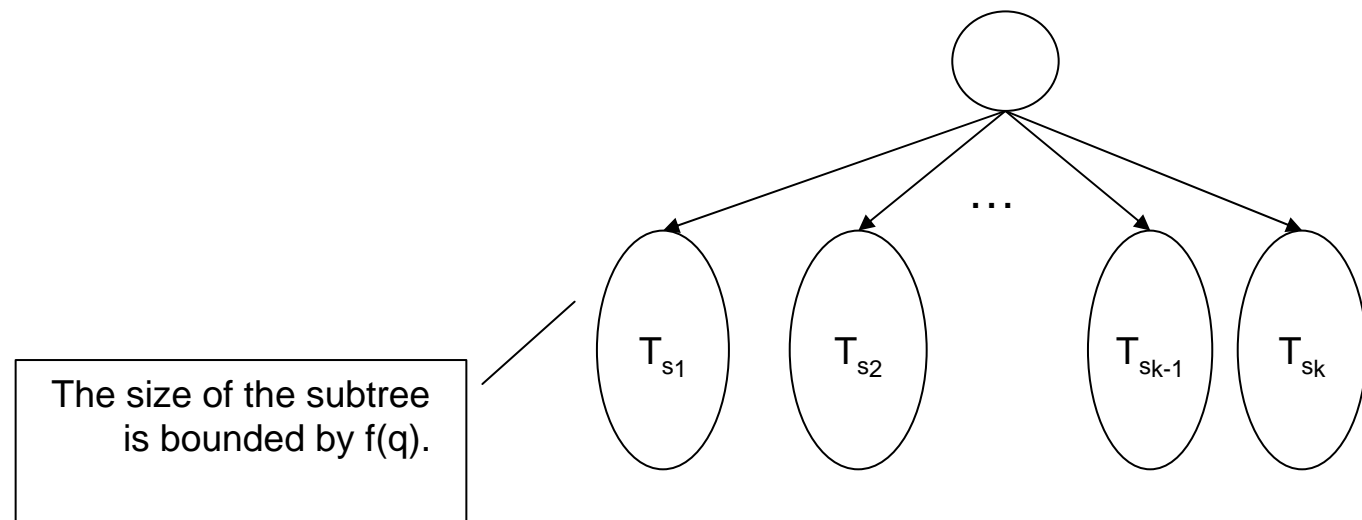
Summary



Properties of optimal GKM trees

Property ('07 Graham, Li, Yao)

- As $n \rightarrow \infty$, the size of each subtree connected to the root is bounded by $\max\{4(\log(1/q)) - 1, 1\}$
- Consistent with my results: As $n \rightarrow \infty$, the number of jumps is bounded by $O(\ln(1/\ln(1/q)))$, and each jump is bounded by $9/4$.
- As $n \rightarrow \infty$, the degree of the root is unbounded.



Introduction

Jumping Sequence
Problem

GLR Algorithm

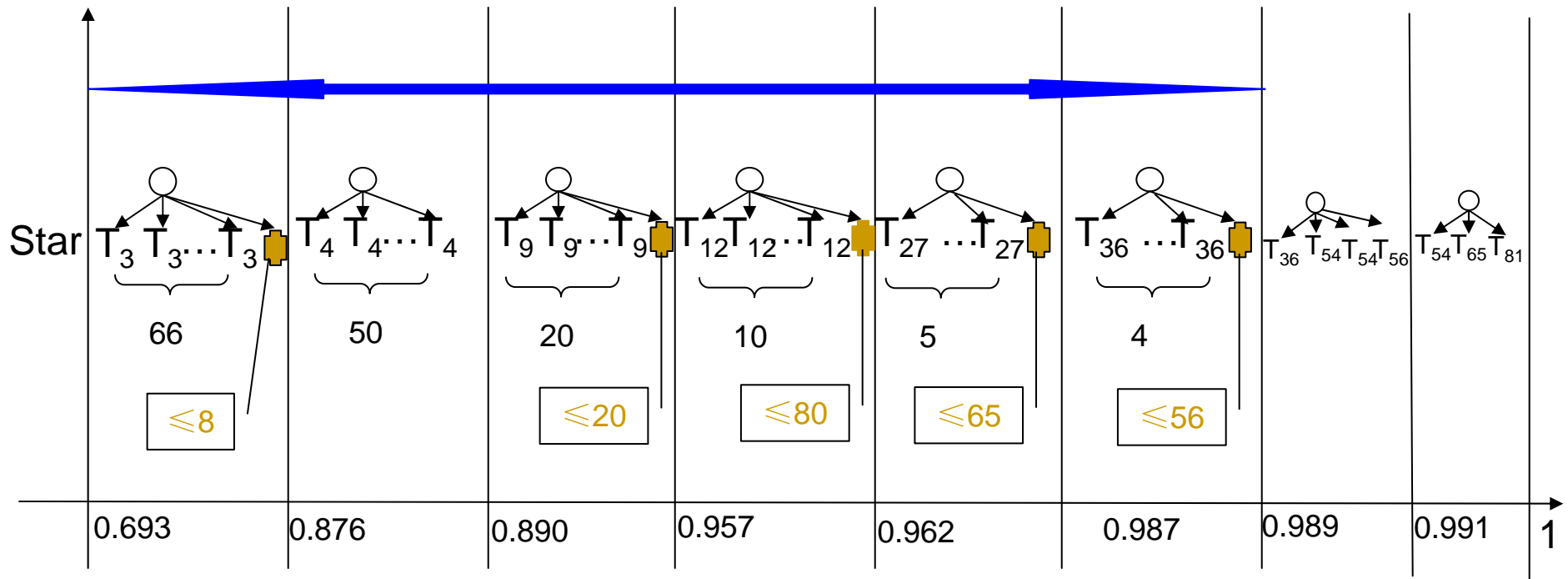
LR Algorithm

Summary



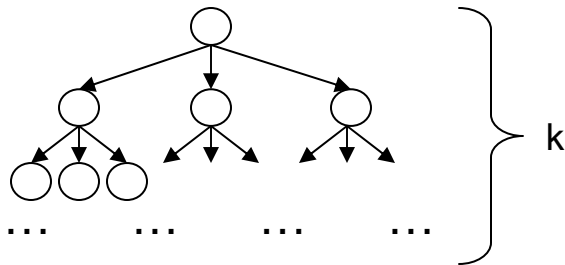
Experimental results

Opt tree structures (n=200)

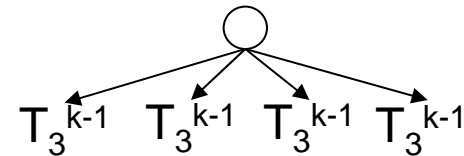


The changing points are calculated by binary search.

T_3^k



$T_{4.3}^{k-1}$



Introduction

Jumping Sequence Problem

GLR Algorithm

LR Algorithm

Summary



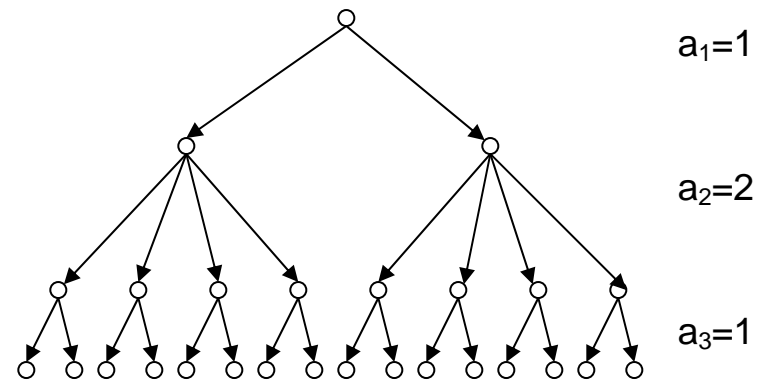
Related work ('03 Zhu, Chan, Noubir)

■ A special set of key trees

- The number of members $N = 2^k$
- $T^2(a_1, a_2, \dots, a_t)$
 - A tree has t levels
 - The node at the i -th level has 2^{a_i} branches

$$T^2(1, 2, 1)$$

$$t = 3; N = 2^4 = 16$$



■ Main results

- The optimal tree must be a star; or, can be written as $T^2(a_1, 2, \dots, 2, a_t)$, where $a_1 \geq 2$, and $a_t = 1$ or 2 .
 - Almost a 4-ary tree
 - The bottom level internal nodes can have outdegree 2 or 4.

Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary



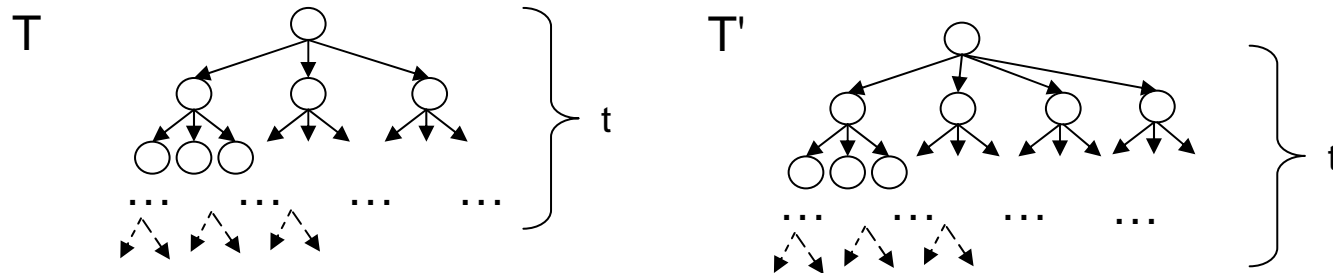
Definition of Uniform Property

■ Uniform Property

- All nodes at the same level have the same degree.
- The tree can be written of the form $T(a_1, a_2, \dots, a_t)$
 - The tree has t levels.
 - The node at the i -th level has outdegree a_i .

■ Theorem

As $n > n_0(q)$, if the **dominant subtree** has the **Uniform Property**, then the dominant subtree can be written as $T((4), 3, \dots, 3, (2))$. [(x) means the level might be missing]



Introduction

Jumping Sequence
Problem

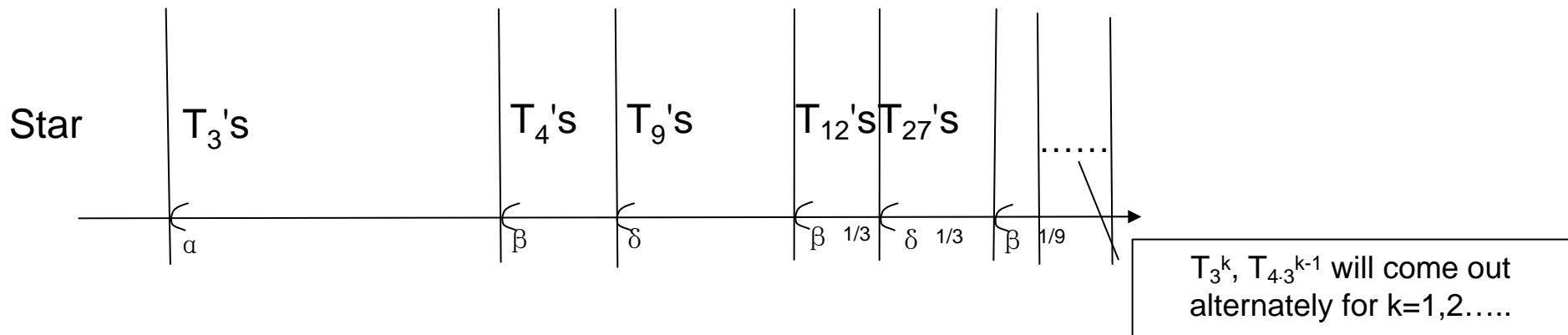
GLR Algorithm

LR Algorithm

Summary



How to find the best dominant tree



Jump points: $\alpha = 1/3^{(1/3)} \approx 0.6934$; $\beta \approx 0.8760$; $\delta \approx 0.8902$.

β is the root of $x^4 - x^3 + 1/12 = 0$; δ is the root of $1/3x^9 + x^4 - x^3 + 7/36 = 0$

β and δ are both in $[1/3^{1/3}, 1]$.

Theorem

Among the trees of the form $T((4), 3, 3, \dots, 3)$:

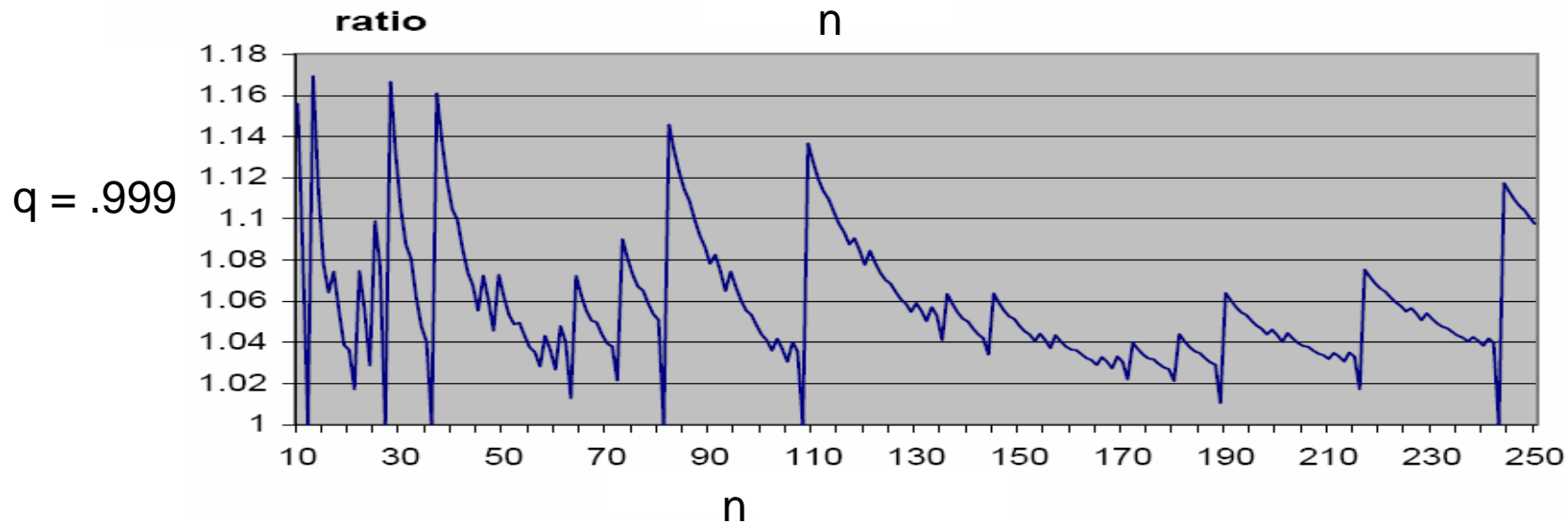
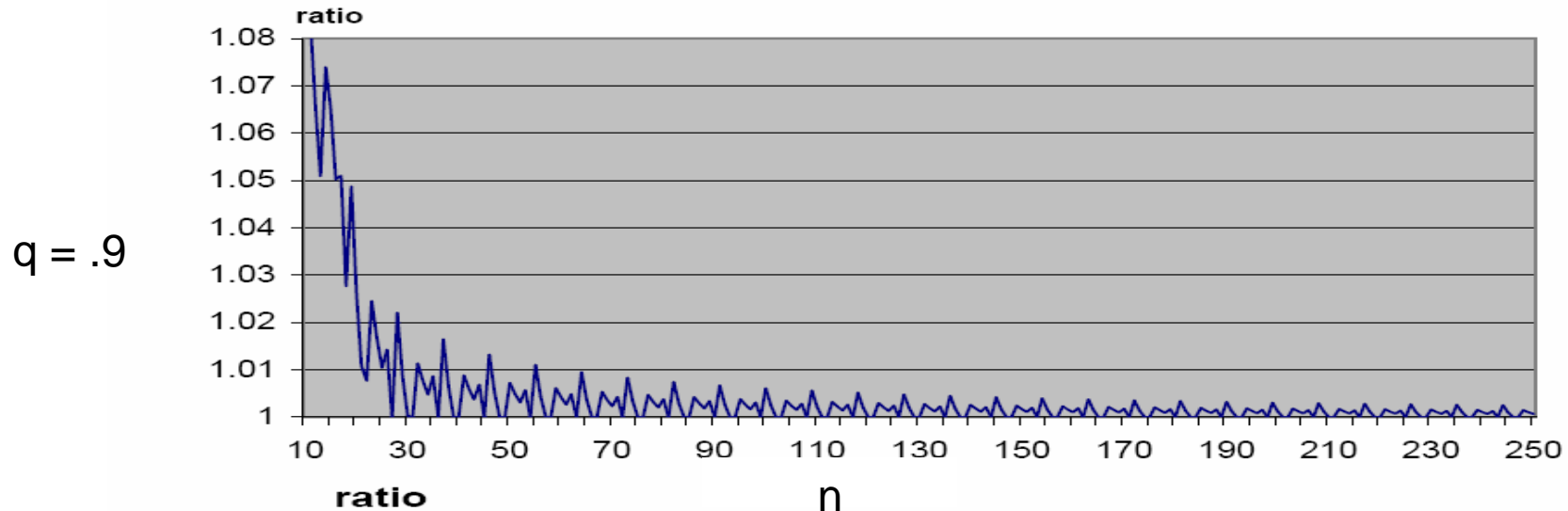
1. If q is in $(\alpha, \beta]$, T_3 is the best.
2. If q is in $(\beta \cdot 1/3^k, \delta \cdot 1/3^k]$, $T_{4 \cdot 3^k}$ is the best dominant subtree. ($k = 0, 1, 2, \dots$)
3. If q is in $(\delta \cdot 1/3^k, \beta \cdot 1/3^{k+1}]$, $T_{3^{k+2}}$ is the best dominant subtree. ($k = 0, 1, 2, \dots$)

T_4 is a subtree with 4 branches, T_9 is a complete ternary tree.

T_3^k is a complete ternary tree.

$T_{4 \cdot 3^{k-1}}$ is a subtree with 4 branches on the first level, the degree of the nodes on the other level is 3.

Simulation Results of Ratio $\text{glr}(q, n)/\text{opt}(q, n)$



Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary

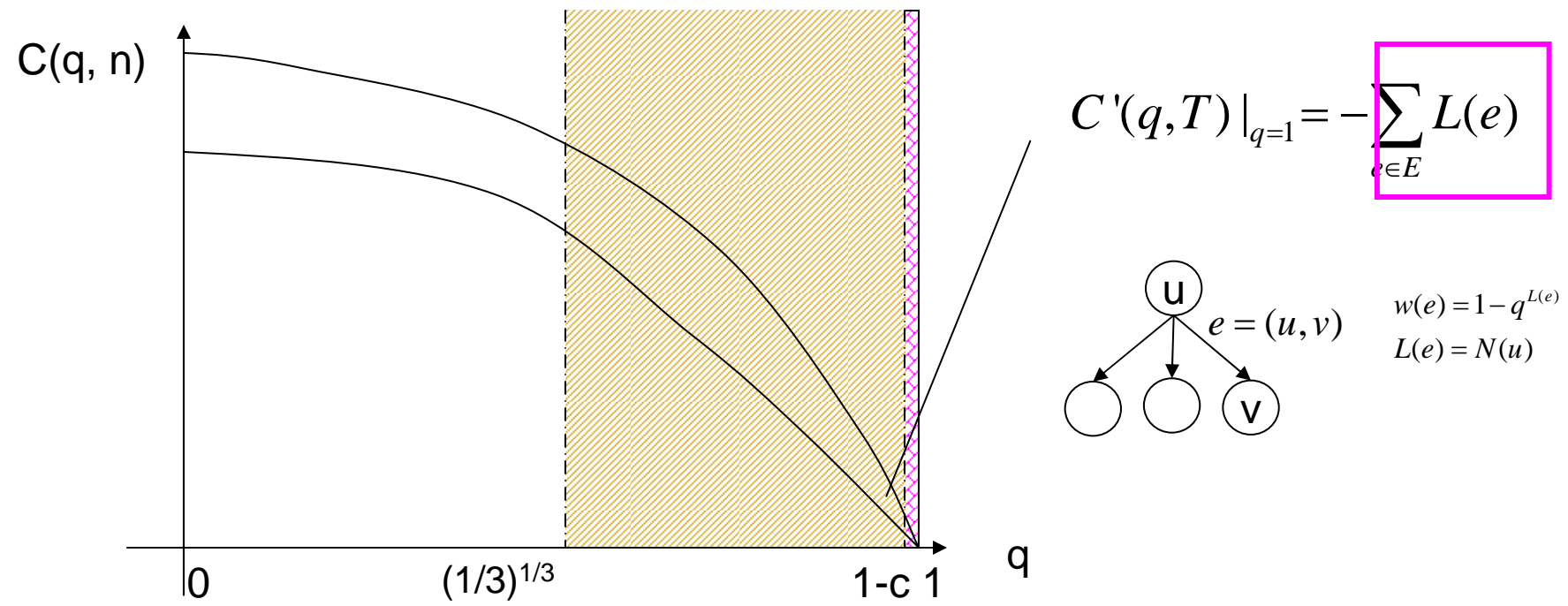


Outline

- Secure group key management overview
 - Jumping sequence problem
 - Properties of optimal jumping sequences
 - Properties of optimal GKM trees
 - Approximation algorithms to build GKM trees
 - The GLR Algorithm
 - The LR Algorithm ($q \rightarrow 1$)
 - Summary
-

GKM tree problem as $q \rightarrow 1$

- Cost of a GKM Tree T : $C(q, T) = \sum_{e \in E} (1 - q^{L(e)})$



The opt GKM tree is the tree with $\min \sum_{e \in E} L(e)$

Idea of LR Algorithm

- As $q \rightarrow 1$, the level of an optimal GKM tree is bounded by $O(\ln n)$.
- T_{DS} is almost a complete ternary tree, except possibly the top and the bottom levels.
- As $q \rightarrow 1$, if $n = 4 \cdot 3^t$, the optimal GKM tree $T^*(n)$ has root degree 4; otherwise, $T^*(n)$ has root degree 3.
- We construct an almost balanced ternary by adding new leaves from a top-down and left-to-right order.

Introduction



Jumping Sequence
Problem



GLR Algorithm



LR Algorithm



Summary



LR Algorithm

Root ($n = 40, t = 3$)



	Leaves (Left)	Leaves (Middle)	Leaves (Right)
$3^t \leq n < 5 \cdot 3^{t-1}$	$n - 2 \cdot 3^{t-1}$	3^{t-1}	3^{t-1}
$5 \cdot 3^{t-1} \leq n < 7 \cdot 3^{t-1}$	3^t	$n - 4 \cdot 3^{t-1}$	3^{t-1}
$7 \cdot 3^{t-1} \leq n < 3^{t+1}$	3^t	3^t	$n - 2 \cdot 3^t$

Introduction

Jumping Sequence
Problem

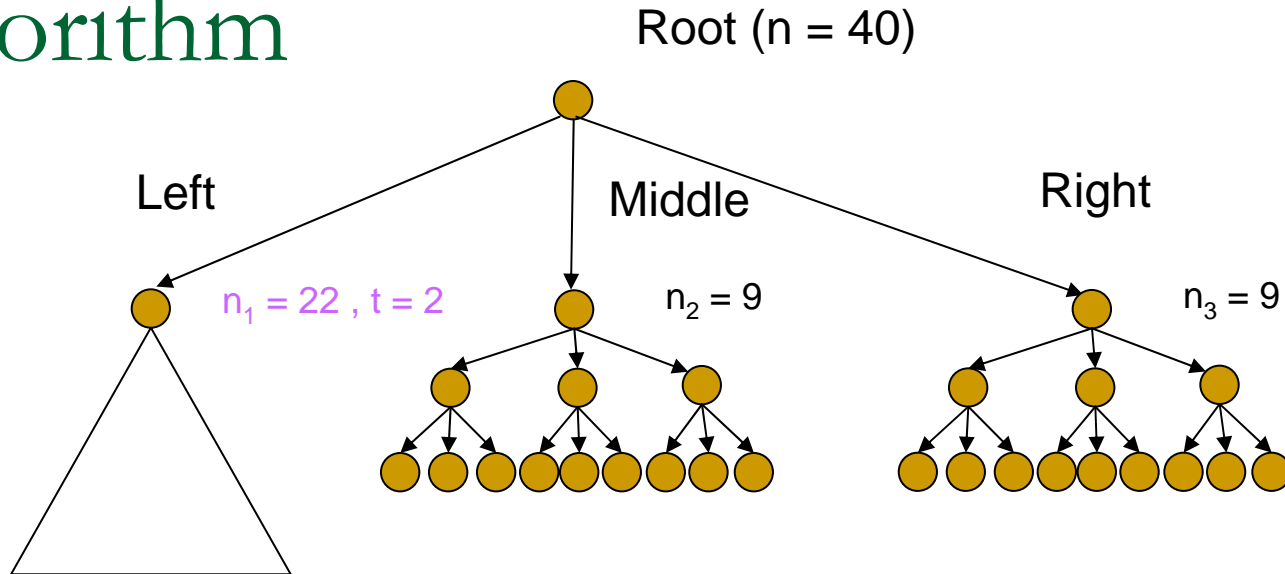
GLR Algorithm

LR Algorithm

Summary



LR Algorithm



	Leaves (Left)	Leaves (Middle)	Leaves (Right)
$3^t \leq n < 5 \cdot 3^{t-1}$	$n - 2 \cdot 3^{t-1}$	3^{t-1}	3^{t-1}
$5 \cdot 3^{t-1} \leq n < 7 \cdot 3^{t-1}$	3^t	$n - 4 \cdot 3^{t-1}$	3^{t-1}
$7 \cdot 3^t \leq n < 3^{t+1}$	3^t	3^t	$n - 2 \cdot 3^t$

Introduction

Jumping Sequence
Problem

GLR Algorithm

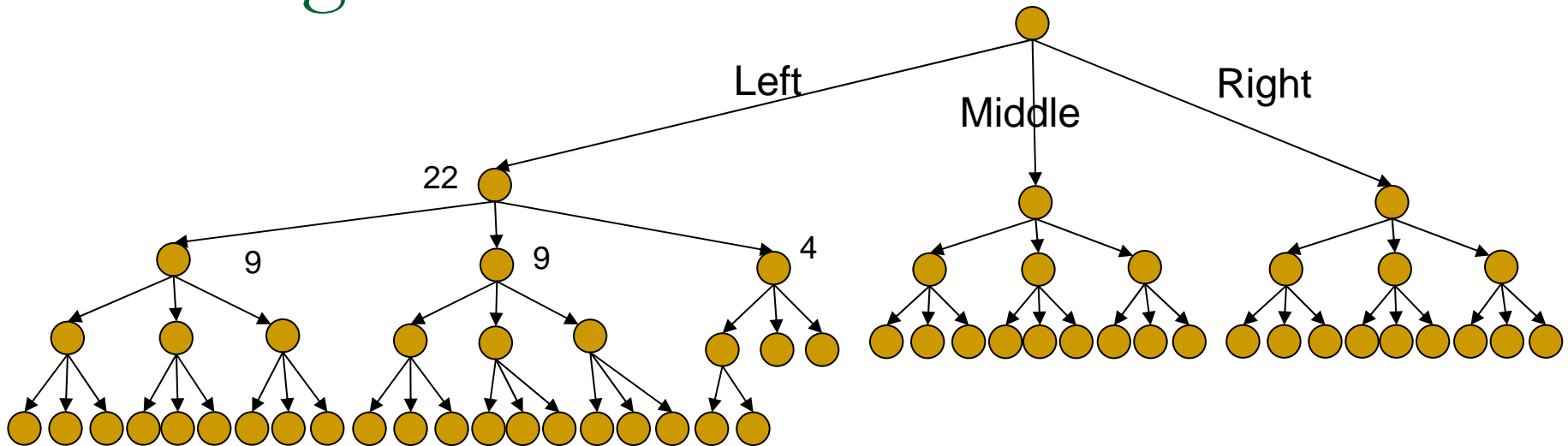
LR Algorithm

Summary



LR Algorithm

Root (n = 40)



	Leaves (Left)	Leaves (Middle)	Leaves (Right)
$3^t \leq n < 5 \cdot 3^{t-1}$	$n - 2 \cdot 3^{t-1}$	3^{t-1}	3^{t-1}
$5 \cdot 3^{t-1} \leq n < 7 \cdot 3^{t-1}$	3^t	$n - 4 \cdot 3^{t-1}$	3^{t-1}
$7 \cdot 3^{t-1} \leq n < 3^{t+1}$	3^t	3^t	$n - 2 \cdot 3^t$

Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary



Properties of Jumping Sequences $q \rightarrow 1$

■ Jumps along the integers

$g_Z(n)$ is defined as the minimum cost of jumping along integers from 1 to n , and $g_Z(n)$ satisfies:

- If $n_1 > n_2$, then $g_Z(n_1) > g_Z(n_2)$.
- If $n = 3^t$ and $S_{3^t} = (1, 3, 9, \dots, 3^t)$
 - $c(S_{3^t}) = 3t$;
 - $g_Z(3^t) \geq 0.996 \cdot 3t$.

Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary



Summary

- Analyze the properties of optimal GKM trees
- Approximation algorithms to build GKM trees
 - The GLR Algorithm
 - The LR Algorithm ($q \rightarrow 1$)
- Jumping Sequence Problems
 - Properties of the optimal Jumping Sequences
 - Case 1: $0 < q < 1$
 - Case 2: $q \rightarrow 1$

Introduction

Jumping Sequence
Problem

GLR Algorithm

LR Algorithm

Summary



References

- ['98 Wong, Gouda, Lam] C. K. Wong, M. Gouda, and S. S. Lam Secure group communications using key graphs, Proceedings of the ACM SIGCOMM '98 conference on ATAPCC.
 - ['01 Li, Yang, Gouda, Lam] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, batch re-keying for secure group communications, WWW10, 2001, Hong Kong.
 - ['03 Zhu, Chan, Noubir] F. Zhu, A. Chan, and G. Noubir, Optimal tree structure for key management of simultaneous join/leave in secure multicast, Proceedings of MILCOM, 2003.
 - ['07 Graham, Li, Yao] R. Graham, M. Li, and F. Yao Optimal tree structures for group key management with batch updates, to appear in SIAM Journal on Discrete Mathematics.
-

Related Publications

- "Approximately optimal trees for group key management with batch updates" (with Minming Li, Ze Feng, R. Graham and Frances F. Yao). Accepted for publication by Special Issue of Theoretical Computer Science.
 - "Optimal jumping patterns" (with Steve Butler and R. Graham). Submitted to Journal of Combinatorics and Number Theory.
-

Thank you!
