

# Cryptography CS 555



## Lecture 24: Group Key Management

Department of Computer Sciences  
Purdue University

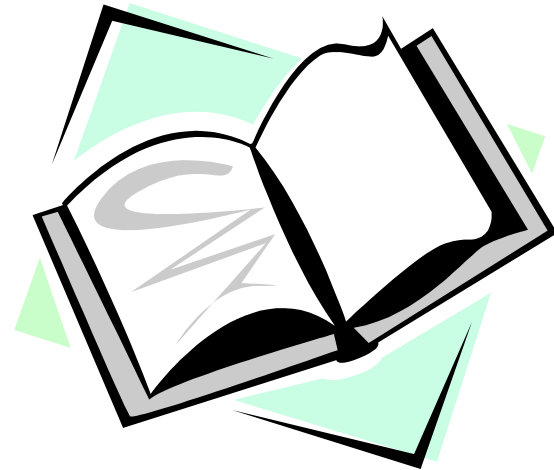
# Lecture Outline

- Key management protocols overview
- LKH
- GDH



# Recommended Reading

- Today lecture prepare mostly from research papers, email me if you are interested in reading the original papers.



# Group Key Management

- Define how to establish and maintain a secret key in a dynamic group
- One-to-many communication
  - large groups  $> 1000$
  - usually one sender
- Many-to-many communication
  - small to medium size groups
  - members are peers, anybody can be both a sender and a receiver.



# Goal and Classification

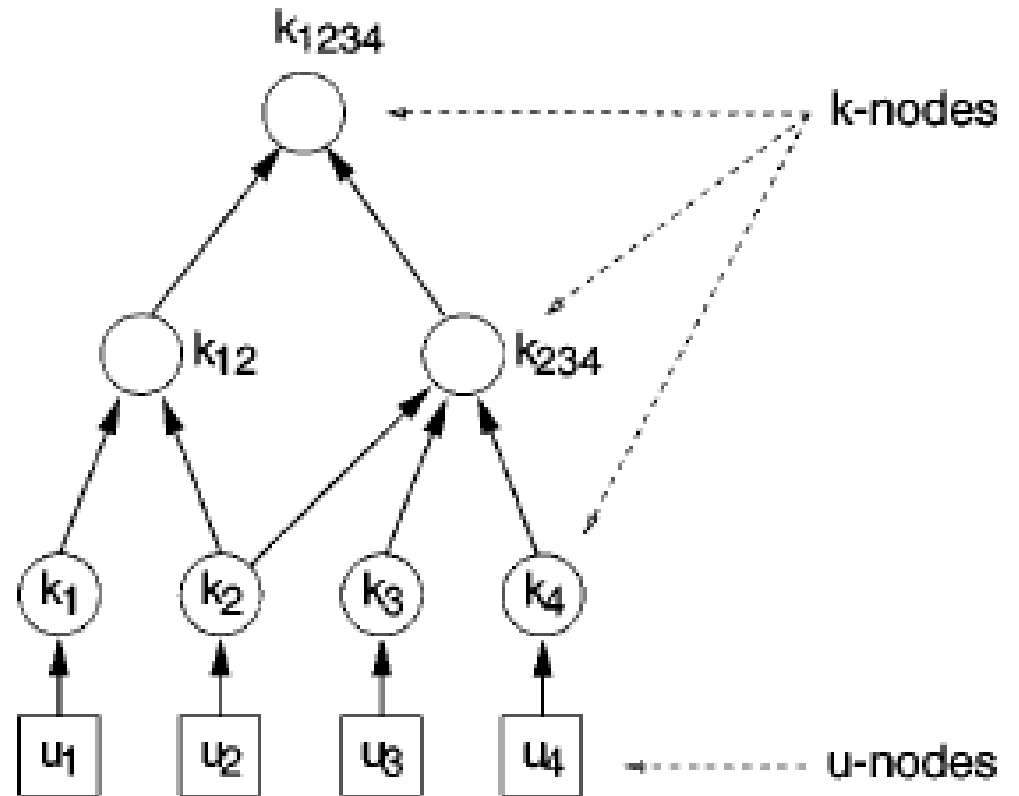
- Provide key refresh in a dynamic group setting.
- Group can change: join, leave, partition, merge.
- Security and scalability requirements.
- Key management protocols:
  - Centralized schemes, one entity (usually a key server) selects the key and distributes it everybody. Not-fault-tolerant;
  - Distributed, still key is selected by one entity, but that entity is not fixed; fault-tolerant;
  - Contributory: every group member contributes to the group key

# Logical Key Graphs (LKH)

- Centralized scheme: relies on a trusted key server that generates and distributes keys to the group.
- Every group member has a secret key shared with the server.
- In addition, there are other keys shared by subgroups with the key server.
- Relies only on symmetric encryption.
- Minimizes rekeying by using a tree structure (maintained by the server).
- Without the tree,  $n$  encryptions required ( $n$  is the group size).

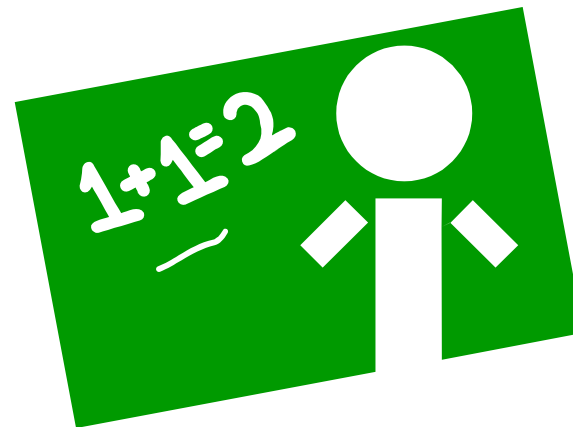
# LKH (cont.)

- Root represents group key
- Leaves represent users
- $k_i$  is the key shared by user  $i$  with server
- An intermediate node represent the key shared by all users in its subtree.



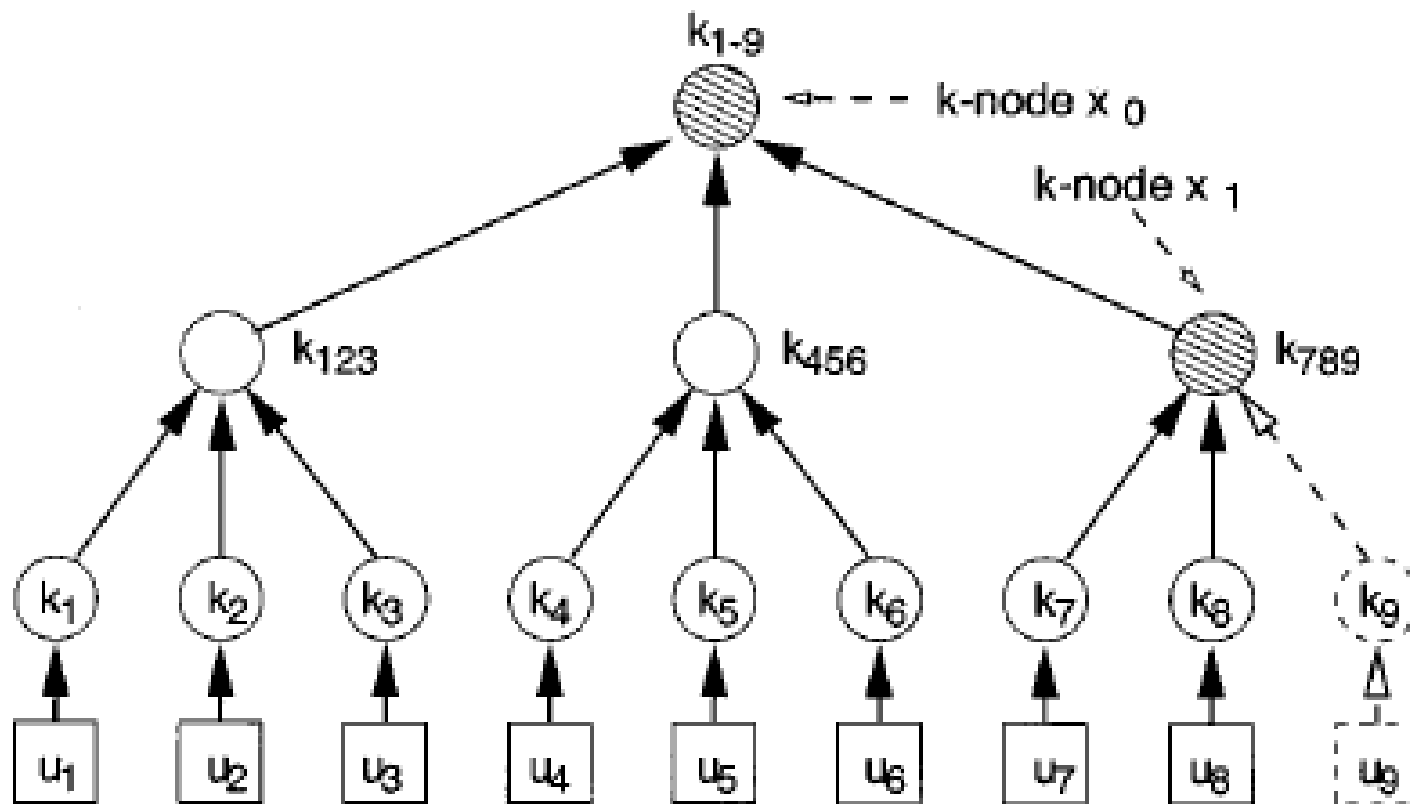
# LKH: Adding a New Member

- Server authenticates the user and establishes a secret key with it.
- Server selects for a joining point in the tree
- All keys along the path from the joining point are changed by the server
- Server distributes these keys using a minimal set of existing keys



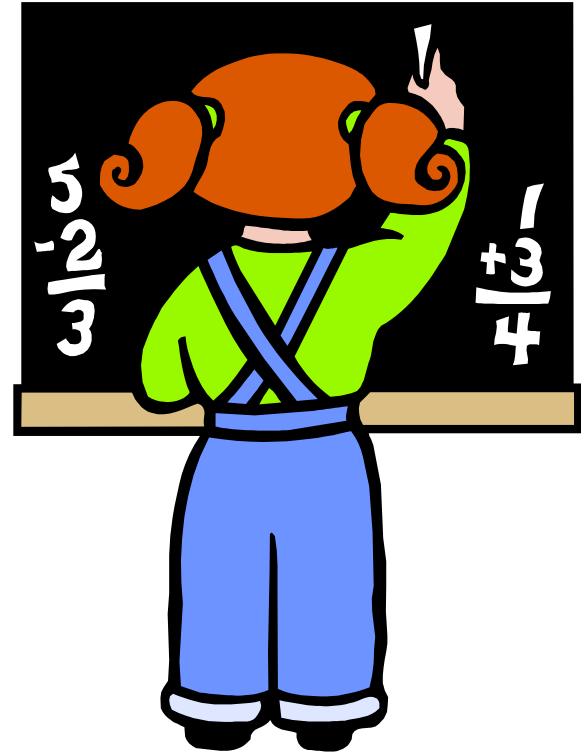
# LKH: Adding a New Member

- Old key  $k_{1-8}$  used to distribute new key  $k_{1-9}$  to users 1-6
- Key  $k_{7-8}$  used to distribute new key  $k_{1-9}$  and  $k_{789}$  to user 7 and 8
- Key  $k_9$  used to distribute new key  $k_{1-9}$  to user 9



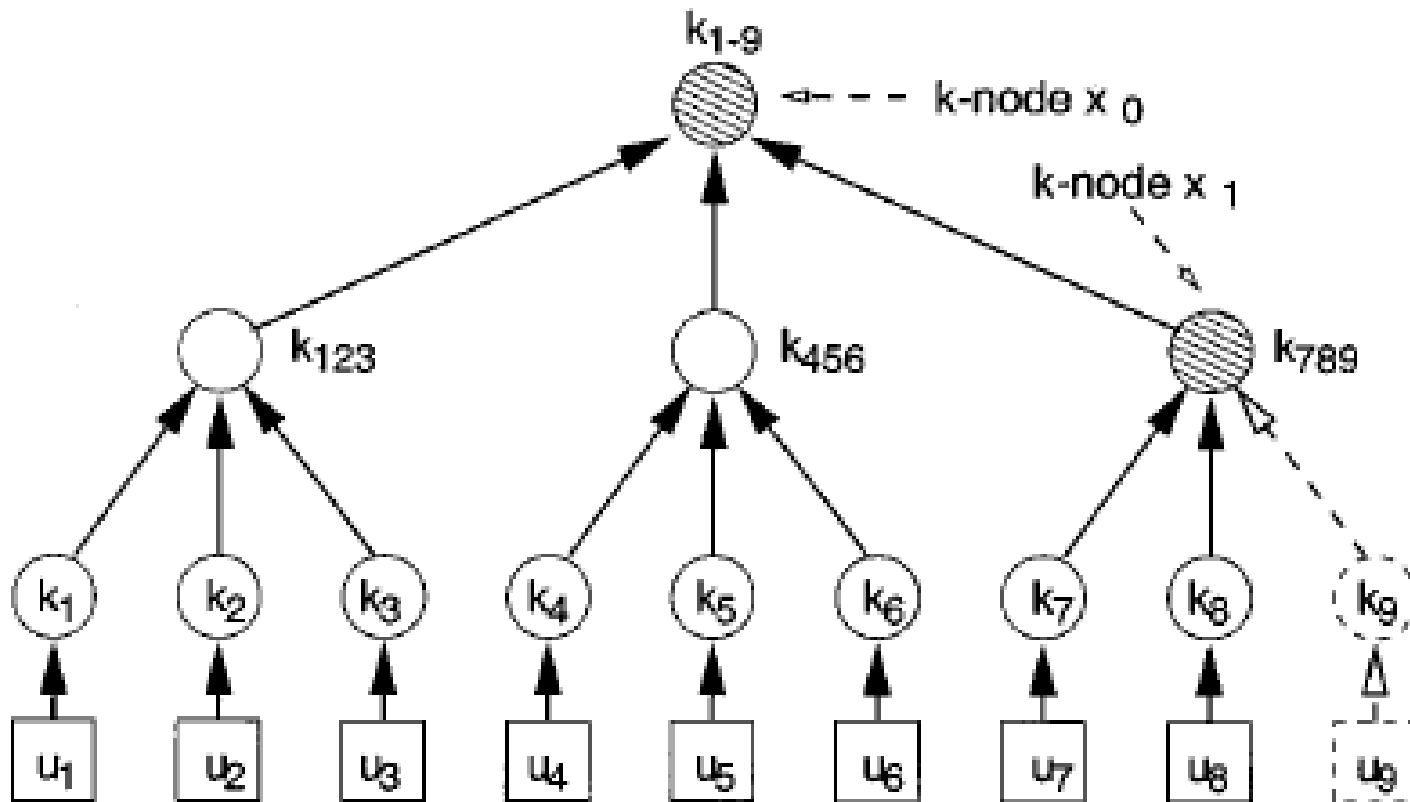
# LKH: Removing a Member

- We want to change every key that was known by the user that leaves.
- Each user receives a new message that contains all the keys in the graph, that changed, including the new group key.
- Goal is to minimize the number of encryptions and messages.



# LKH: Removing a Member

- Key  $k_{123}$  used to distribute new key  $k_{1-8}$  to users 123
- Key  $k_{456}$  used to distribute new key  $k_{1-8}$  to users 456
- Key  $k_7$  used to distribute  $k_{1-8}$  and key  $k_{7-8}$  to user 7
- Key  $k_8$  used to distribute  $k_{1-8}$  and key  $k_{7-8}$  to user 8



# Contributory Key Agreement Protocols

- Distributed, no centralized server.
- Each group member contributes a share to the group key.
- Goal is to provide key independence and perfect forward secrecy.
- Most of such protocols rely on Diffie-Hellman.

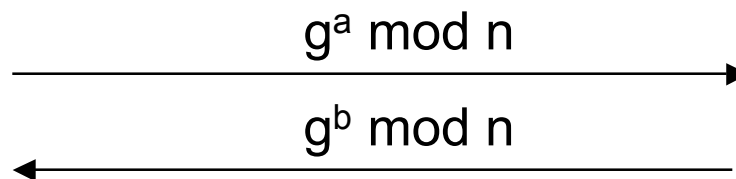
# Key Agreement: Diffie-Hellman Protocol

- Key agreement protocol, both A and B contribute to the key
- Setup  $Z_n$ ,  $n$  prime and  $g$  generator,  $n$  and  $g$  public.



Pick random, secret  $a$   
Compute and send  $g^a \bmod n$

$$K = (g^b \bmod n)^a = g^{ab} \bmod n$$



Pick random, secret  $b$   
Compute and send  $g^b \bmod n$

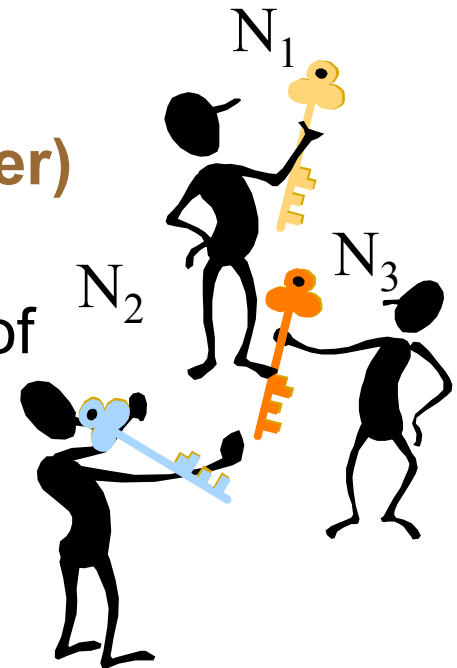
$$K = (g^a \bmod n)^b = g^{ab} \bmod n$$

# Group Diffie-Hellman (GDH)

- Contributory group key agreement protocol suite based on Diffie-Hellman key exchange.
- Each member uses a list of partial keys  $g_i^{\frac{N_1 * \dots * N_n}{N_i}} \text{ mod } p$

$$K_{group} = g^{N_1 * \dots * N_n} \text{ mod } p$$

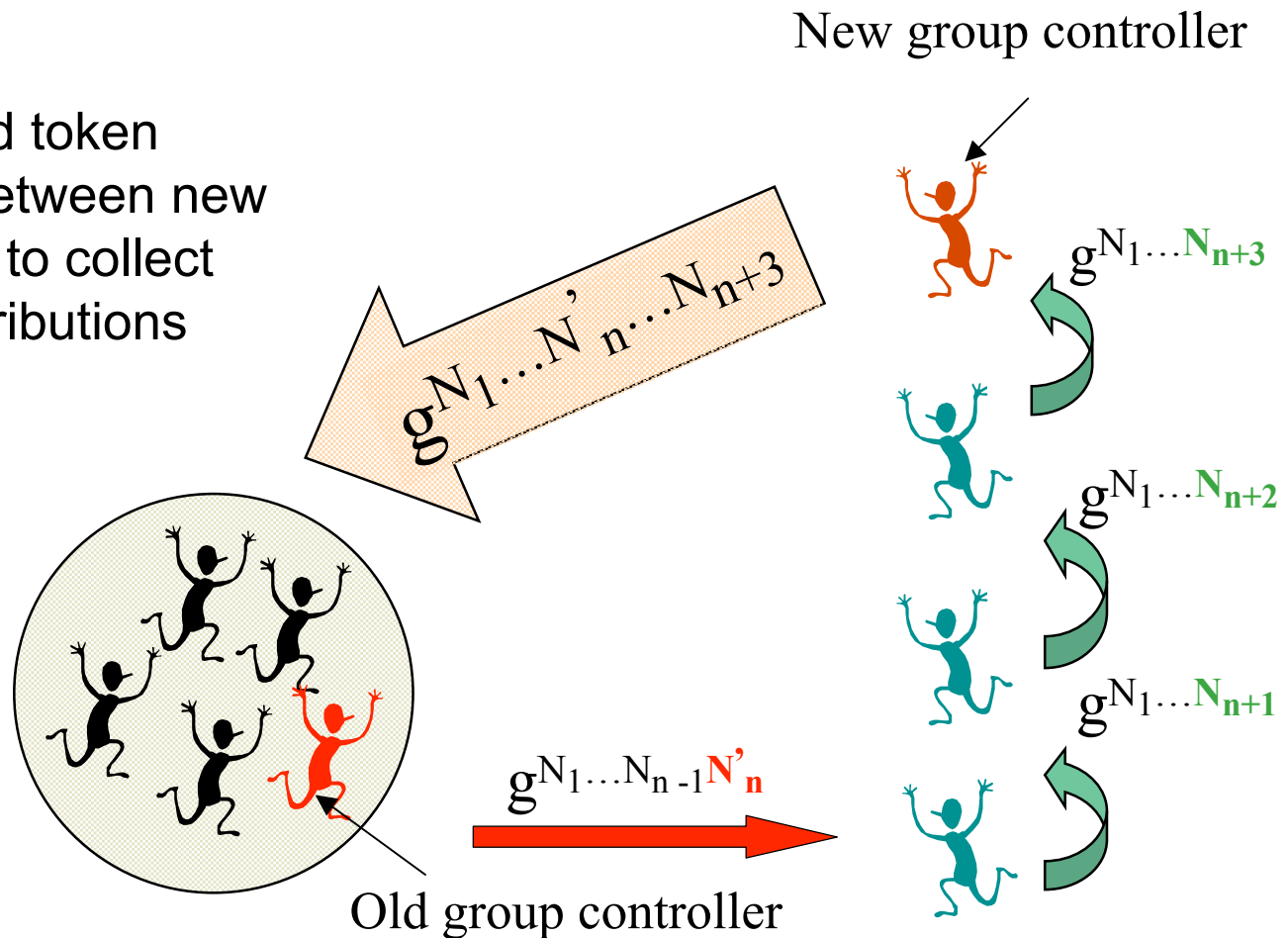
- A member of the group (**group controller**) is maintaining the list.
- Messages are authenticated by means of digital signatures.



# GDH Merge

## STEP 1

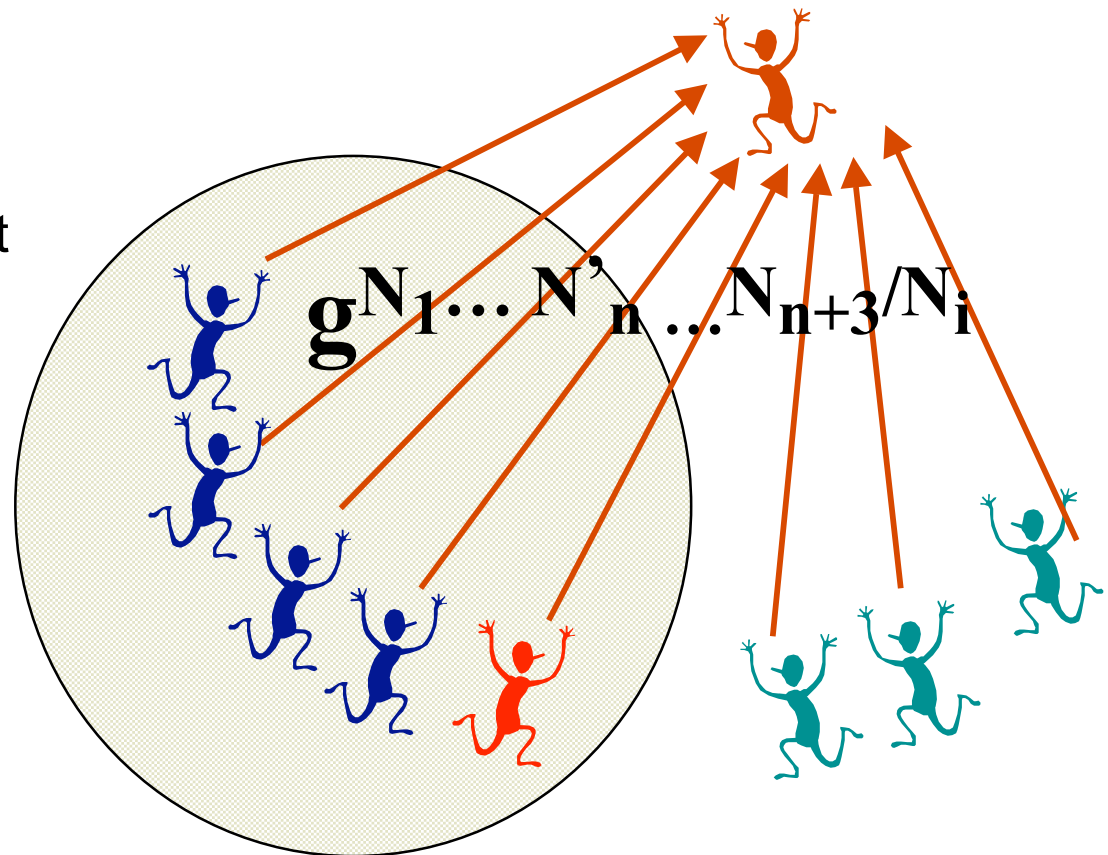
- Refreshed token passed between new members to collect their contributions



# GDH Merge (cont.)

## STEP 2

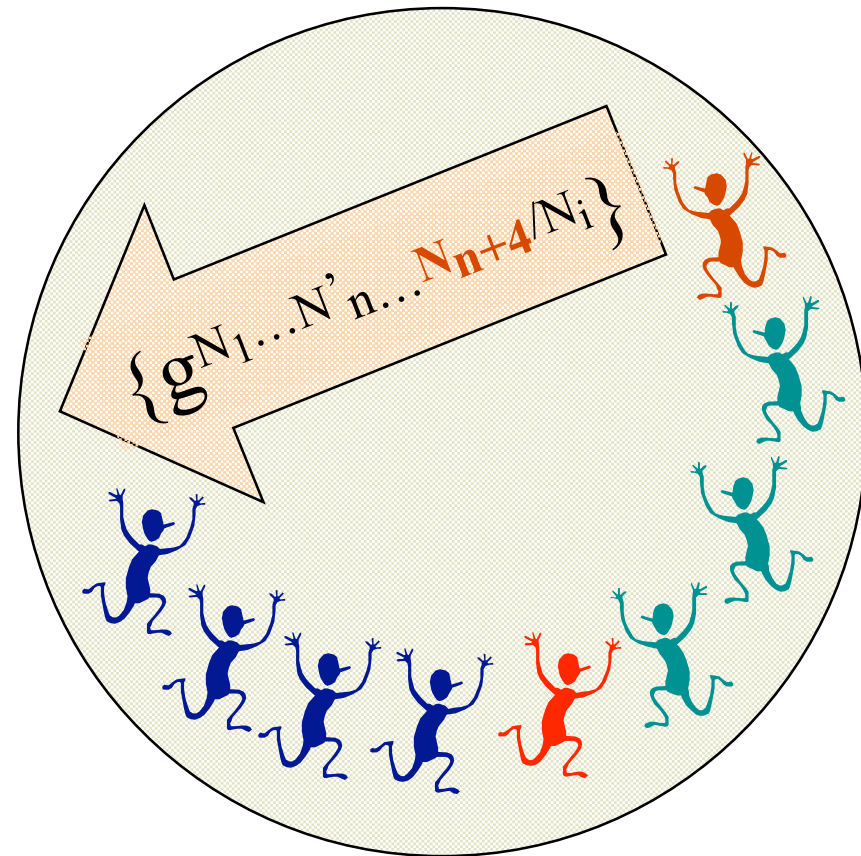
- Old and new members factor out their share and send it to the new controller.



# GDH Merge (cont.)

## STEP 3

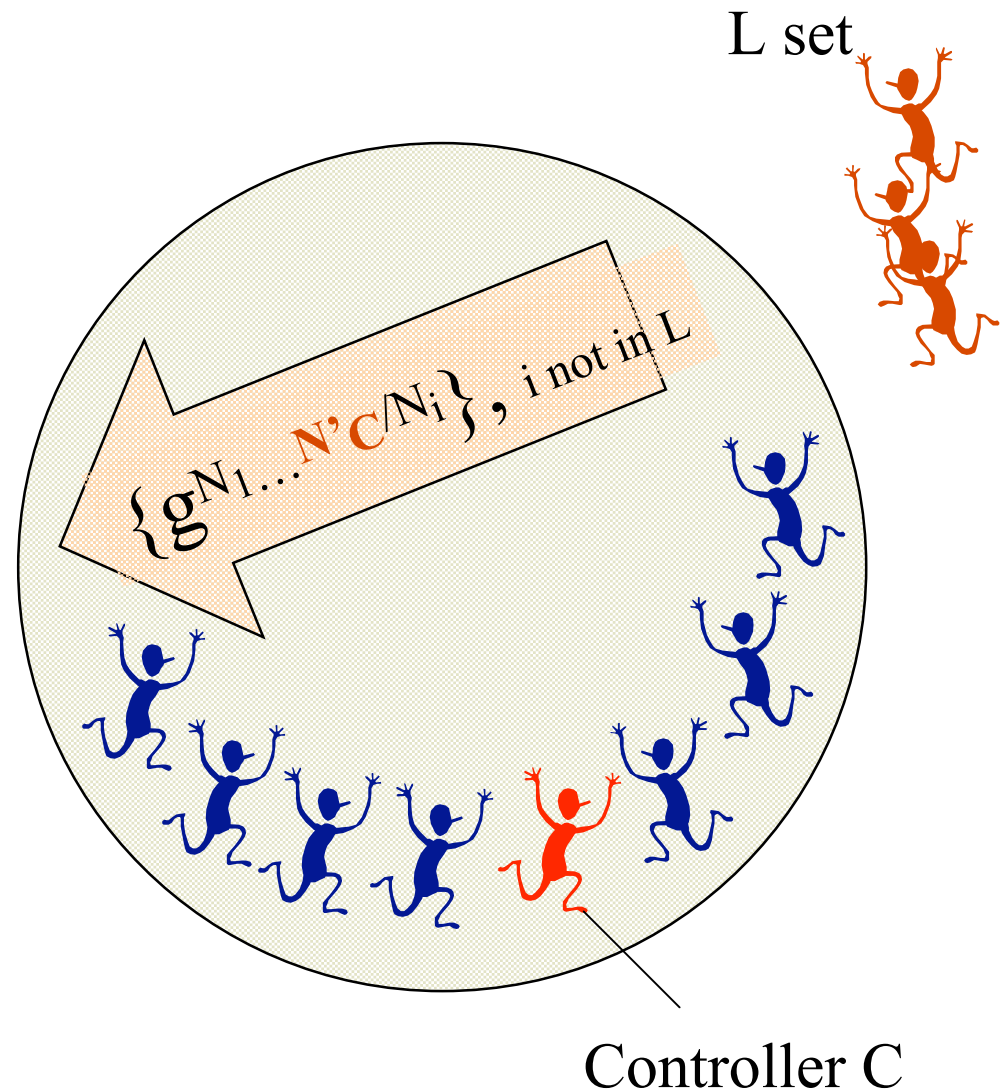
- New controller collects factored out tokens in a key list, adds its share to the key list, and broadcasts the list to the new group.



# Leave

## Group controller

1. removes from the list members that left
2. refreshes its contribution for every partial key in the list
3. Broadcast the list



# Summary

- Group Key Protocols:
  - Centralized, for large groups, usually rely on symmetric encryption, scalability is a main goal
  - Contributory, small peer groups, usually rely on Diffie-Hellman, stronger security properties, more expensive



# Next Lecture...

- Threshold cryptography
- Shamir's scheme
- RSA Threshold signature

