Simple and Fault-tolerant Group Key Agreement

ACM CCS 2000

June 20, 2001

Yongdae Kim(USC), Adrian Perrig(UCB), Gene Tsudik(UCI),

Group-oriented Settings

- 1-to-Many
  - Single source broadcast: Cable/sat., TV, radio
- Few-to-Many
  - Broadcast Multi-source: Televised debates, GPS, time
- Any-to-Any
  - Collaborative applications(Peer group)
  - Video/Audio conferencing, collaborative workspaces, interactive chat, network games and gambling
  - Rich communication semantics, tighter control, more emphasis on **security**

Dynamic Peer Group

- Relatively small(< 100 of members)
- No Hierarchy
- Frequent Membership Changes
- Any member can be sender and receiver

Our Focus: Group Key management in DPGs

Group Key Establishment

- Generating and distributing a secret quantity known only to all current group members
- Group Key Distribution
  - One party generates a secret and distributes it to other parties
  - LKH, McGraw and Sherman, Canetti et. al
- Group Key Agreement
  - A shared secret key is derived by two or more specified parties
  - As a function of information jointly contributed by each member
  - No party can predetermine the resulting value
  - CLIQUES, TGDH, Burmester-Desmedt

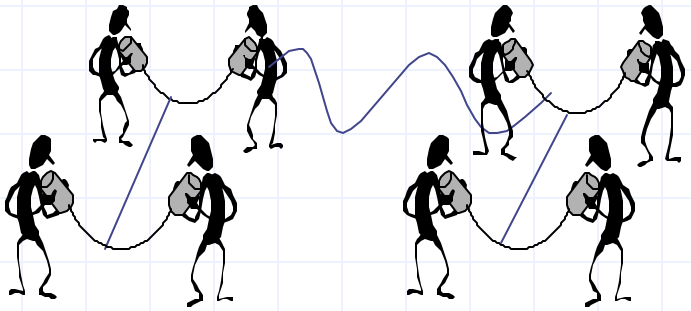
Is Key Distribution enough?

- Centralization violates the peer nature of group
  - Concentrating all key generation in a single point
  - Centralizing trust
- Centralized key server
  - A single point of failure
  - An attractive attack target
- Can key server be sufficiently replicated? ⇒ Very costly
  - Availability of a key server in any and all possible partitions
    - Network can have arbitrary faults!
  - One way is to establish a pairwise secure channel with each member [Rodeh et. al. NDSS 2000]

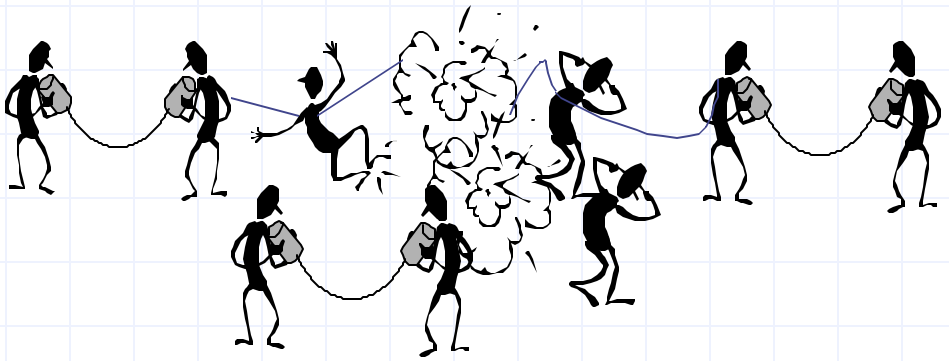
Mutual Needs between GKA and GCS

- Group communication system needs specialized security mechanisms
- Group key agreement protocols rely on the underlying group communication systems
  - Protocol message transport
  - Strong membership semantics (Notification of a group membership)

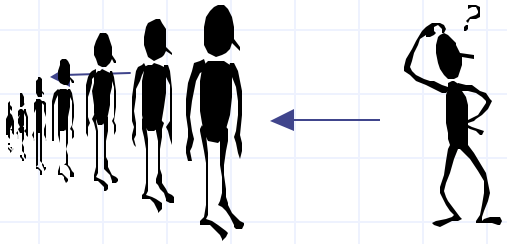
Membership Operations



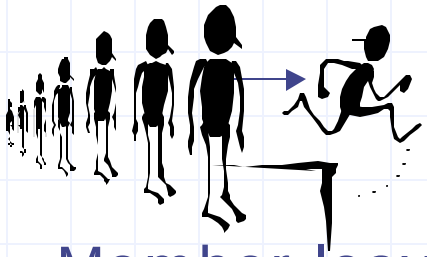
Formation



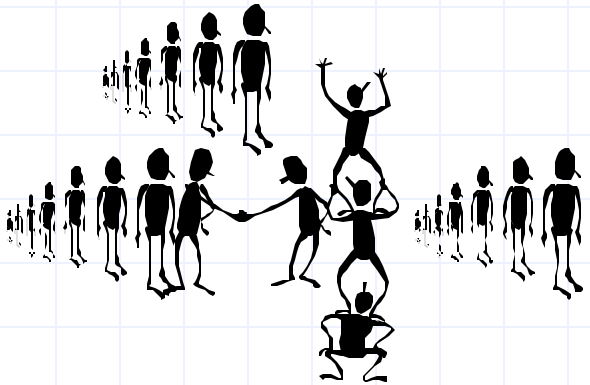
Group partition



Member add



Member leave



Group merge

Security Requirements

- Group key secrecy
  - computationally infeasible for a passive adversary to discover any group key
- Backward secrecy
  - Any subset of group keys cannot be used to discover previous group keys
- Forward secrecy
  - Any subset of old group keys cannot be used to discover subsequent group keys
- Key Independence
  - Any subset of group keys cannot be used to discover any other group keys
  - Forward + Backward secrecy

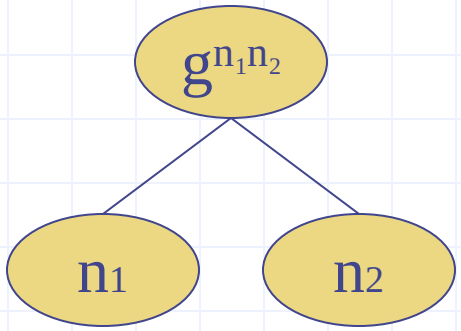
Our Contribution: TGDH

- TGDH = Tree-based Group Diffie-Hellman
- Combining Key Trees with Diffie-Hellman
- Group key agreement in dynamic peer groups
- Simple: One function is enough to implement it
- Fault-tolerant: Robust against cascaded faults
- Secure
  - Provable security
  - Key independence
- Efficient
  - Maximum number of exponentiation = 4(d - 1)
  - # of exp. in Cliques = 2N+1
  - N: Number of members, d: Height of the key tree

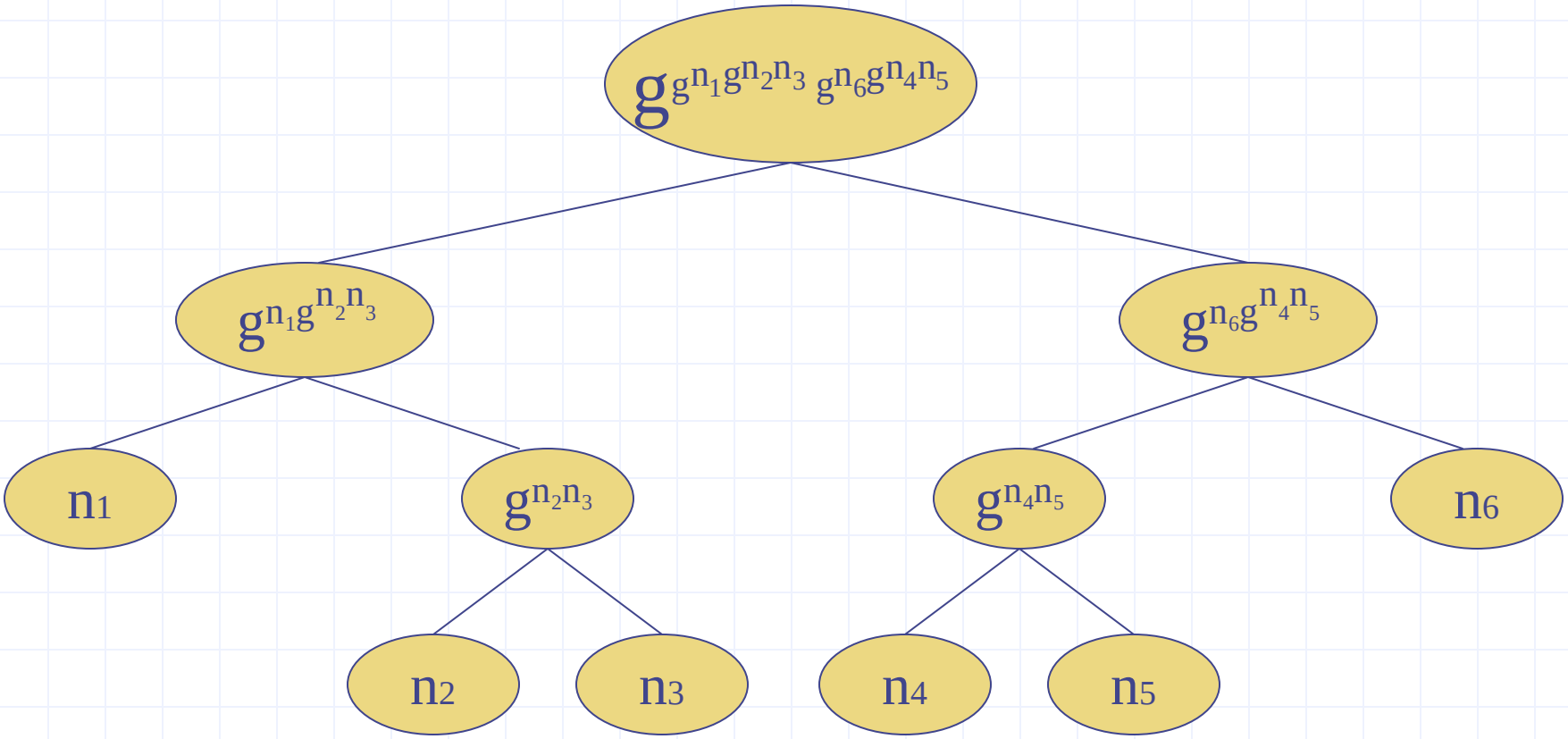
Diffie-Hellman

- $A \rightarrow B : g^{n1}$
- $B \rightarrow A : g^{n2}$
- $A : g^{n1n2} = (g^{n2})^{n1}$
- $B : g^{n1n2} = (g^{n1})^{n2}$
- Any member who knows
  - blinded key(g^{n1}) of other member
  - and its secret n2

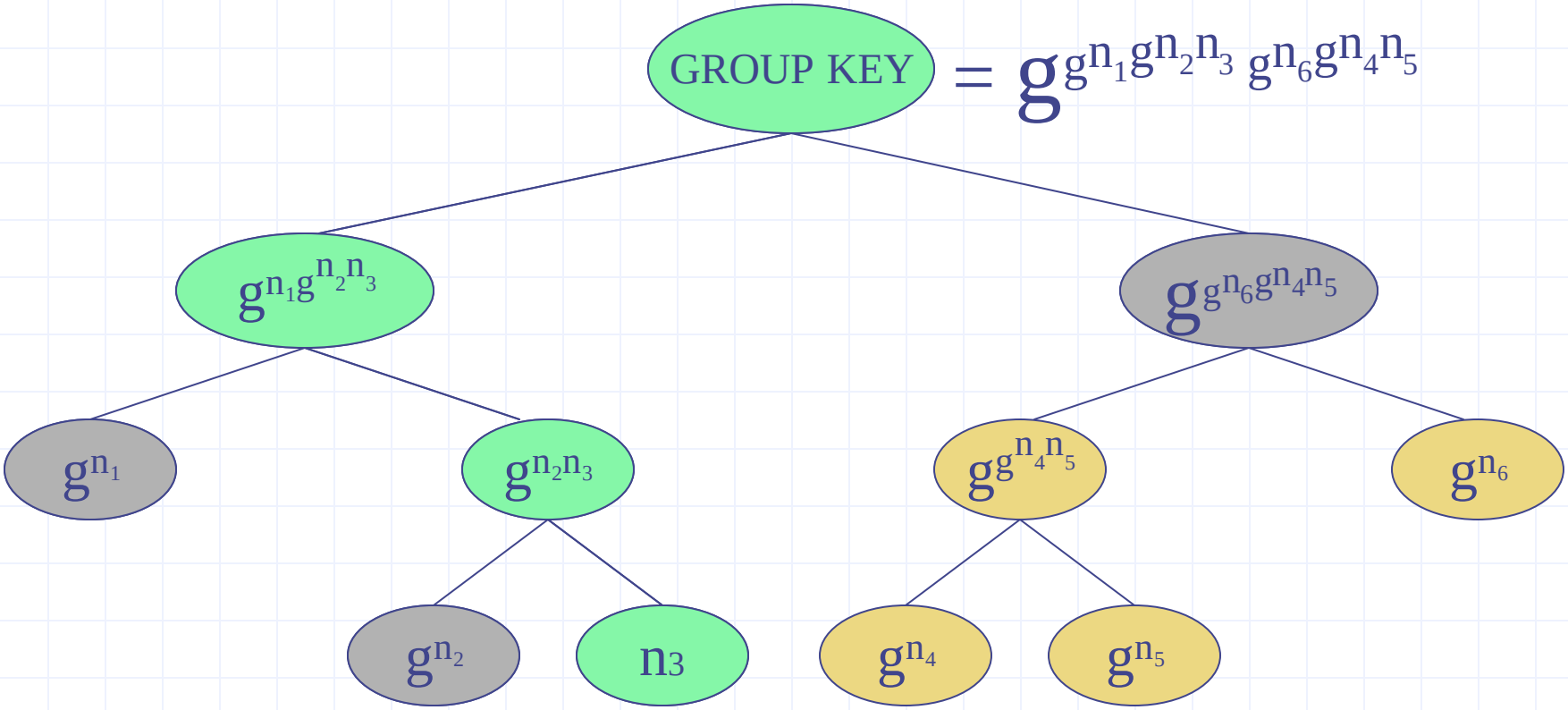
  can compute group key $(g^{n1})^{n2}$



Key Tree(General)

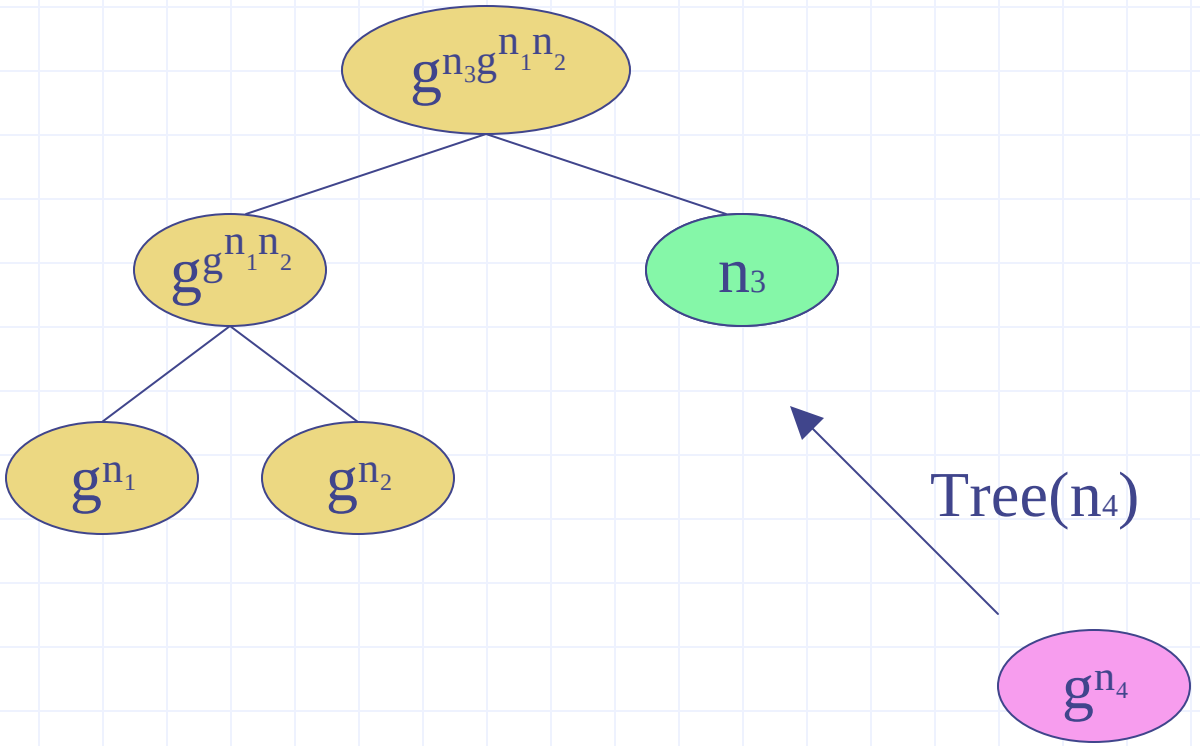


Key Tree(n_3's view)

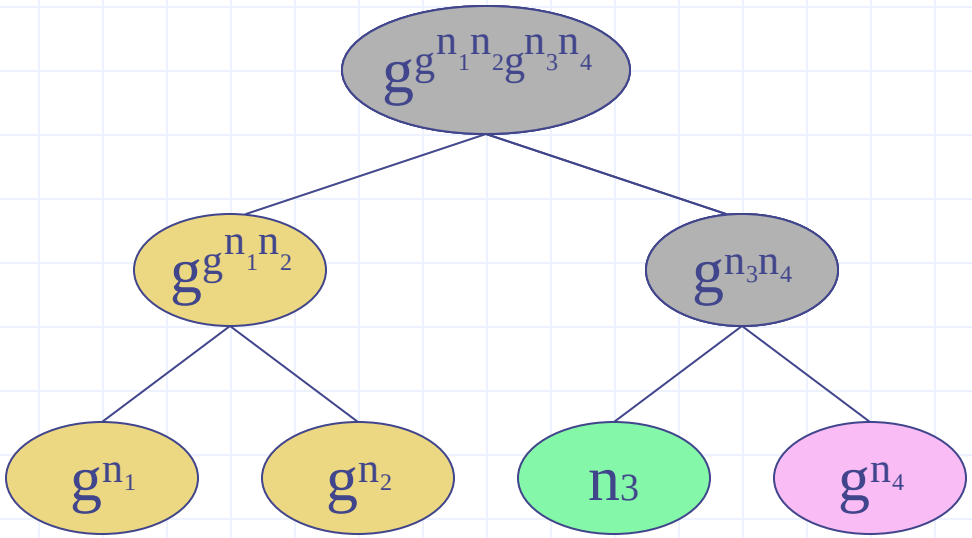


Any member who knows blinded keys on every nodes and
Its session random can compute the group key.

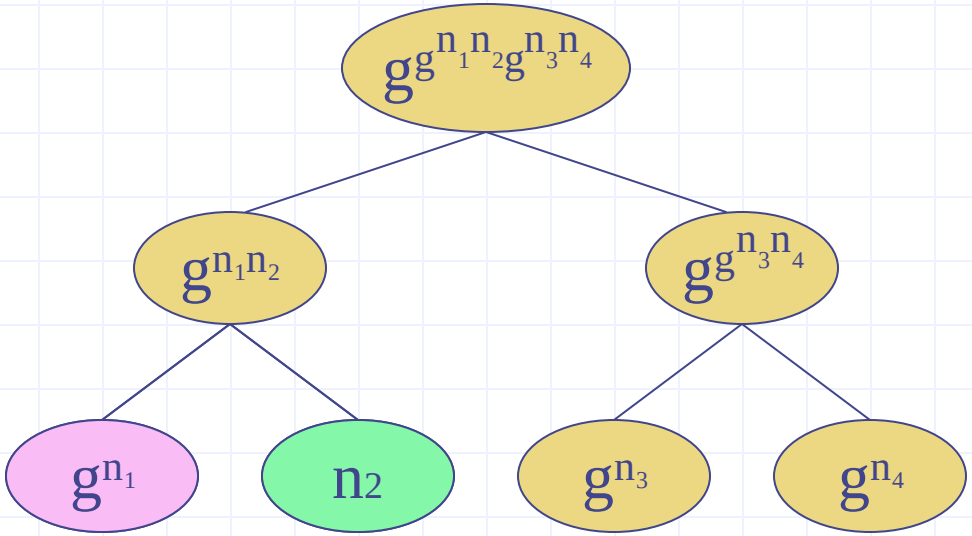
Join(n_3's view)



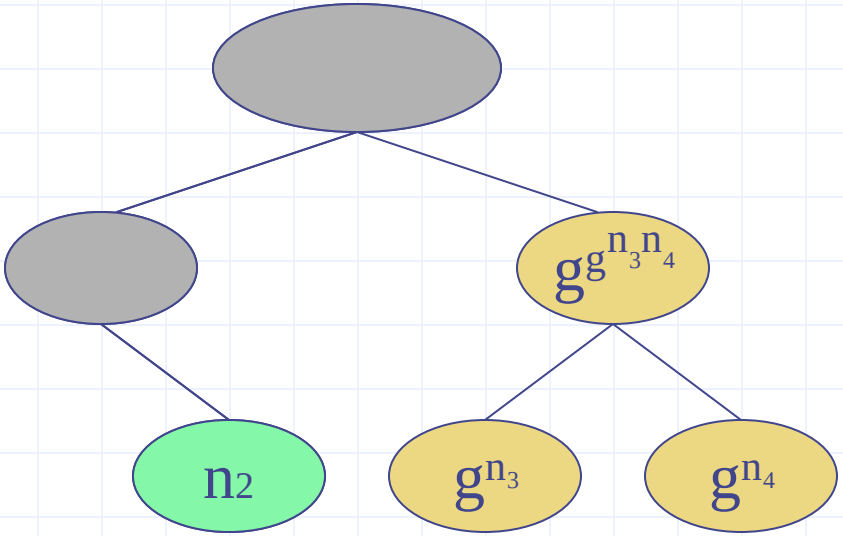
Join(n_3's view)



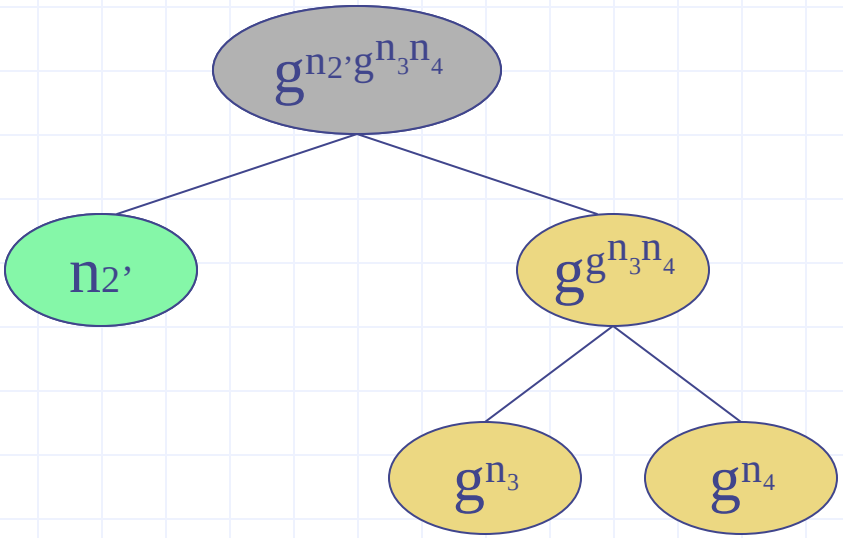
Leave(n_2's view)



Leave(n_2's view)



Leave(n_2's view)



Tree Management : Do one's best

- Join or Merge Policy
  - Join to leaf or intermediate node, if height of the tree will not increase
  - Join to root, if height of the tree increases
- Leave or Partition policy
  - No one can expect who will leave or partitioned out
  - No policy for leave or partition event

Tree Management: Etc

- Why balanced tree?
  - # of exponentiation depends on height of tree
- How good is our policy
  - With a fully balanced tree(height 8) with 256 users, generate random partition and merge for 10000 times
  - Maximum: 15
  - Average: 12
  - Still logarithmic to number of users
- Rebalancing?

Self-stabilization and Fault-tolerance

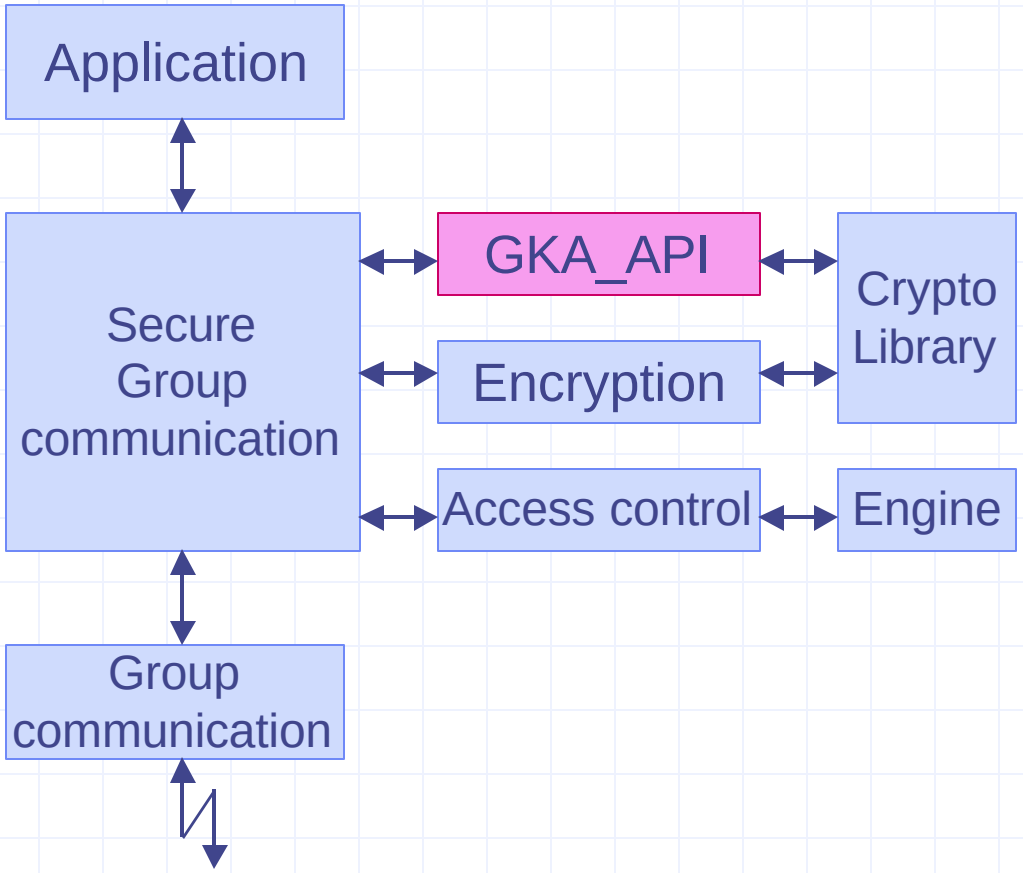
- Four protocols actually represent different strands of a single protocol
- Every protocol consists of
  - Tree management(delete or add nodes) for each membership event
  - Compute missing keys if I can
  - If I computed, broadcast my tree
- This leads us to design self-stabilizing protocol
- Cascaded event: Join, leave, merge, or partition occurs while a prior event is being handled

Security (group key secrecy)

- We have proofs using
  - Random Oracle Model
  - Becker and Wille's proof
- 3-party Tree-DDH is equivalent to 2-party DDH
  - $(g^a, g^b, g^c, g^{g^{ab}}, g^{cg^{ab}})$ and $(g^a, g^b, g^c, g^{g^{ab}}, r)$
  - We are currently extending this to general n-party case.
- We can provide key independence
  - By changing sponsor's session random on every additive event

Implementation

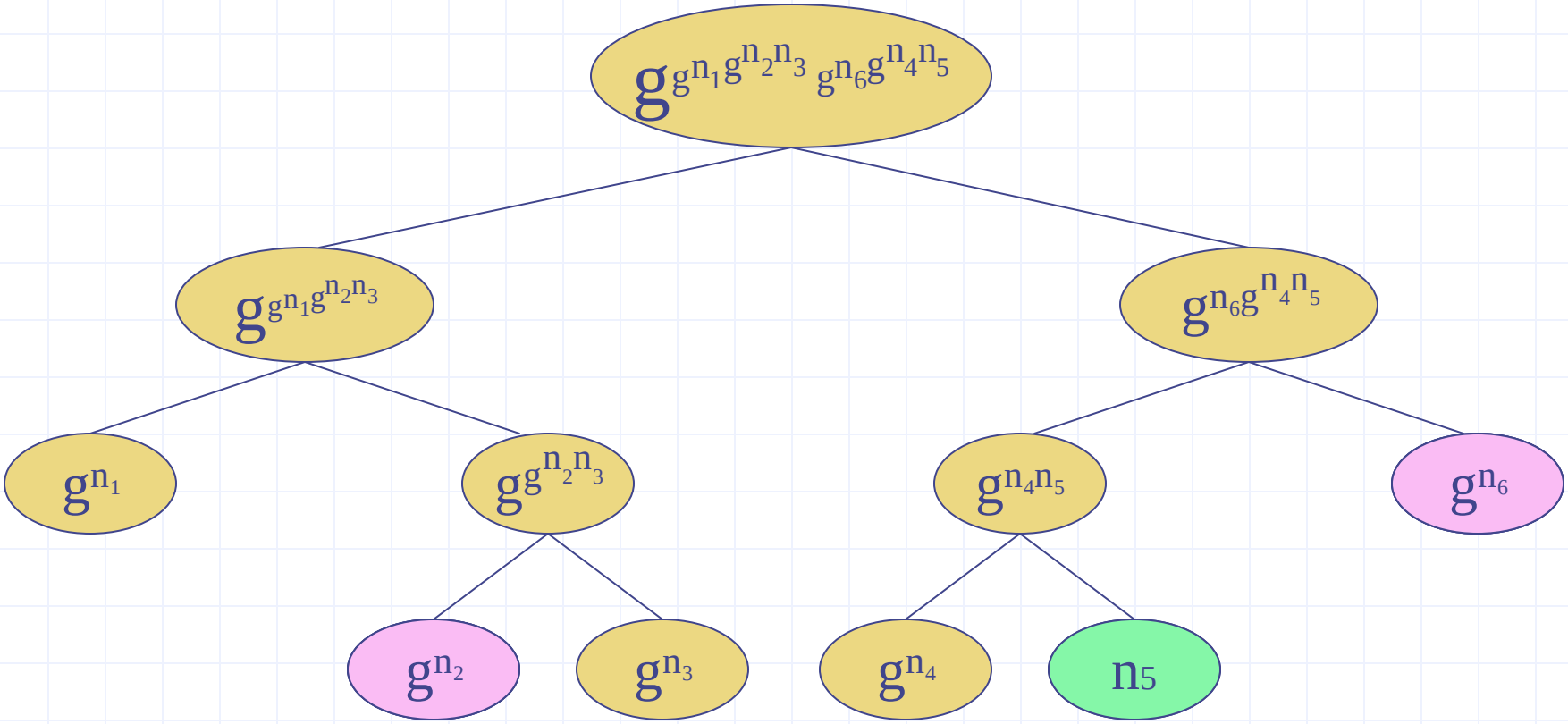
- OpenSSL
- Group communication system
  - Message transport
  - Membership
- Functions
  - new_user
  - merge_request
  - cascade
    - Implements self-stabilization



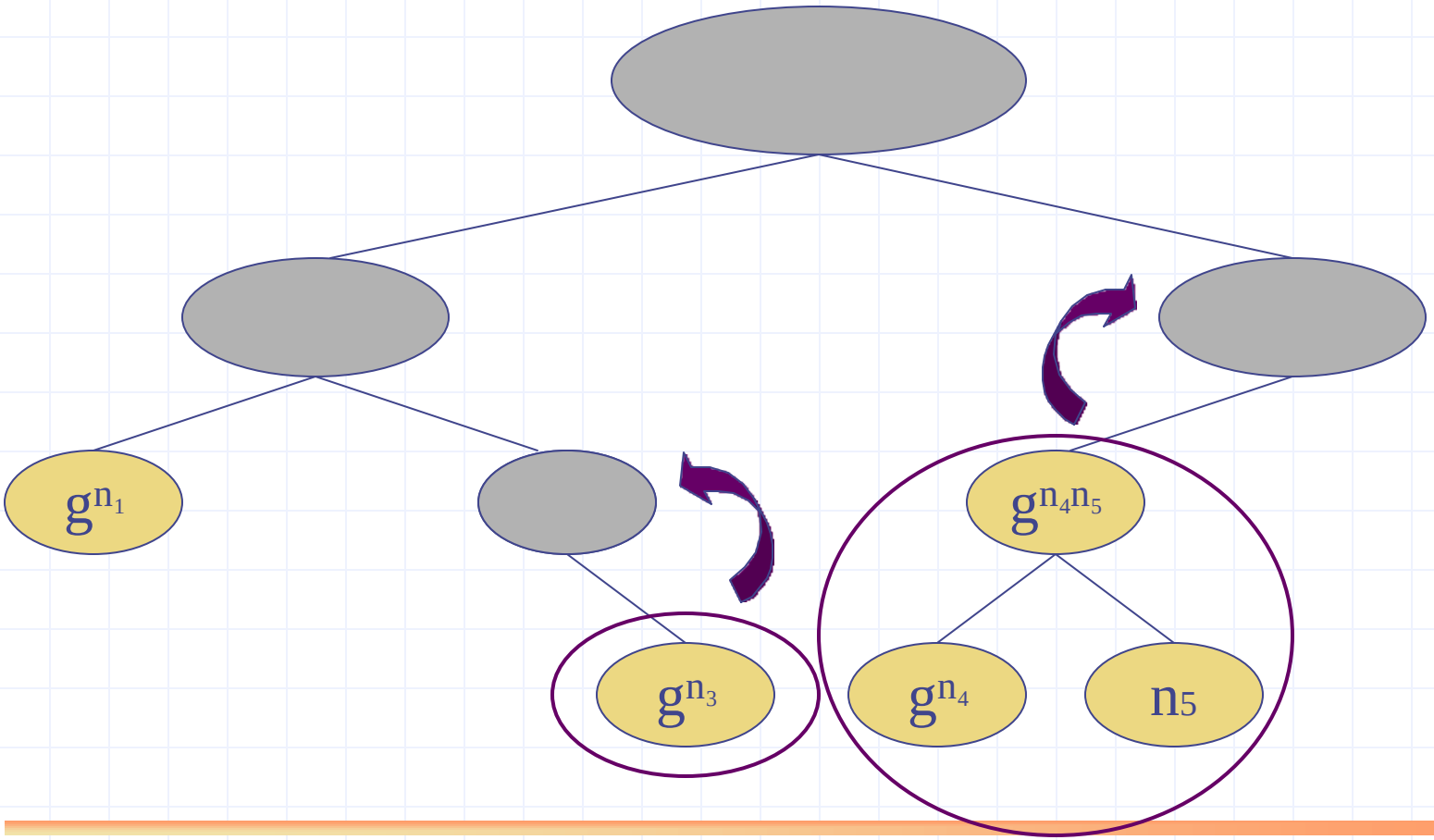
- CLIQUES webpage

  http://www.isi.edu/~gts/CLIQUES

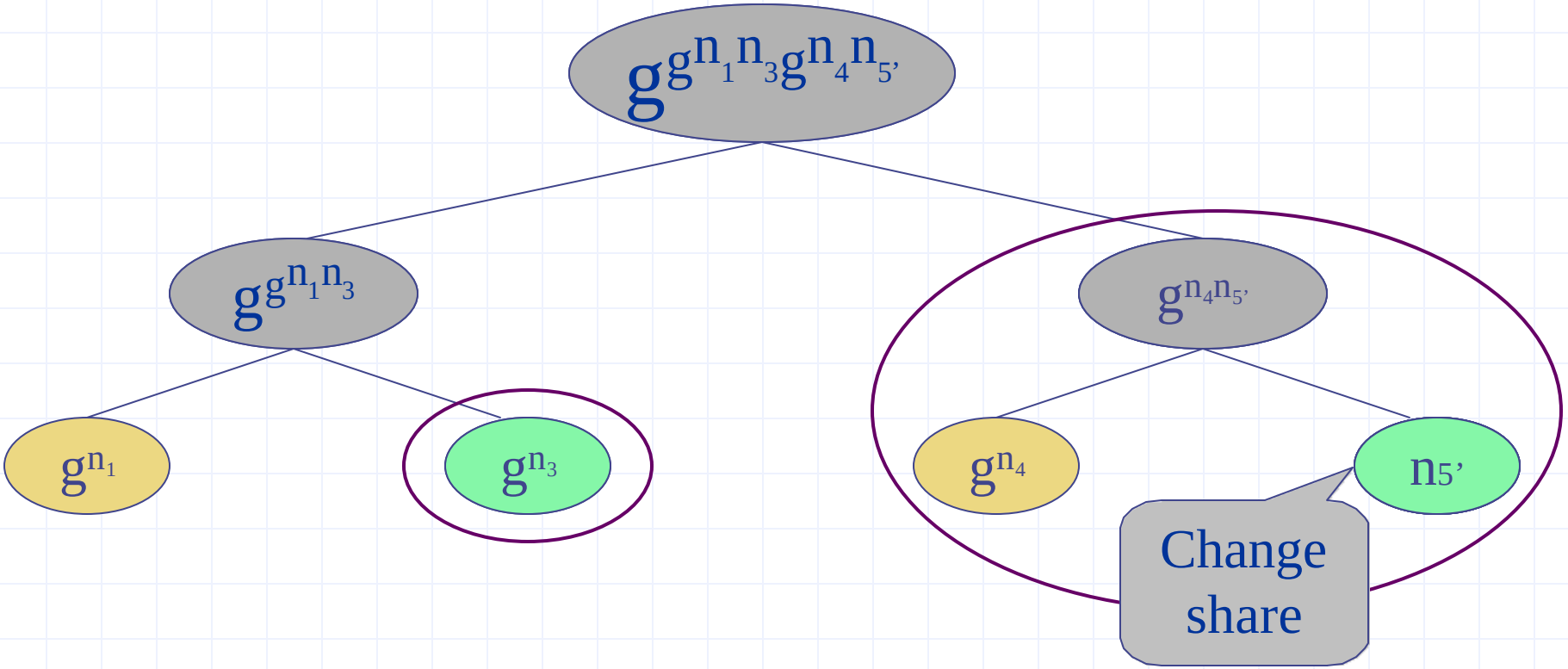
Partition(n_5's view)



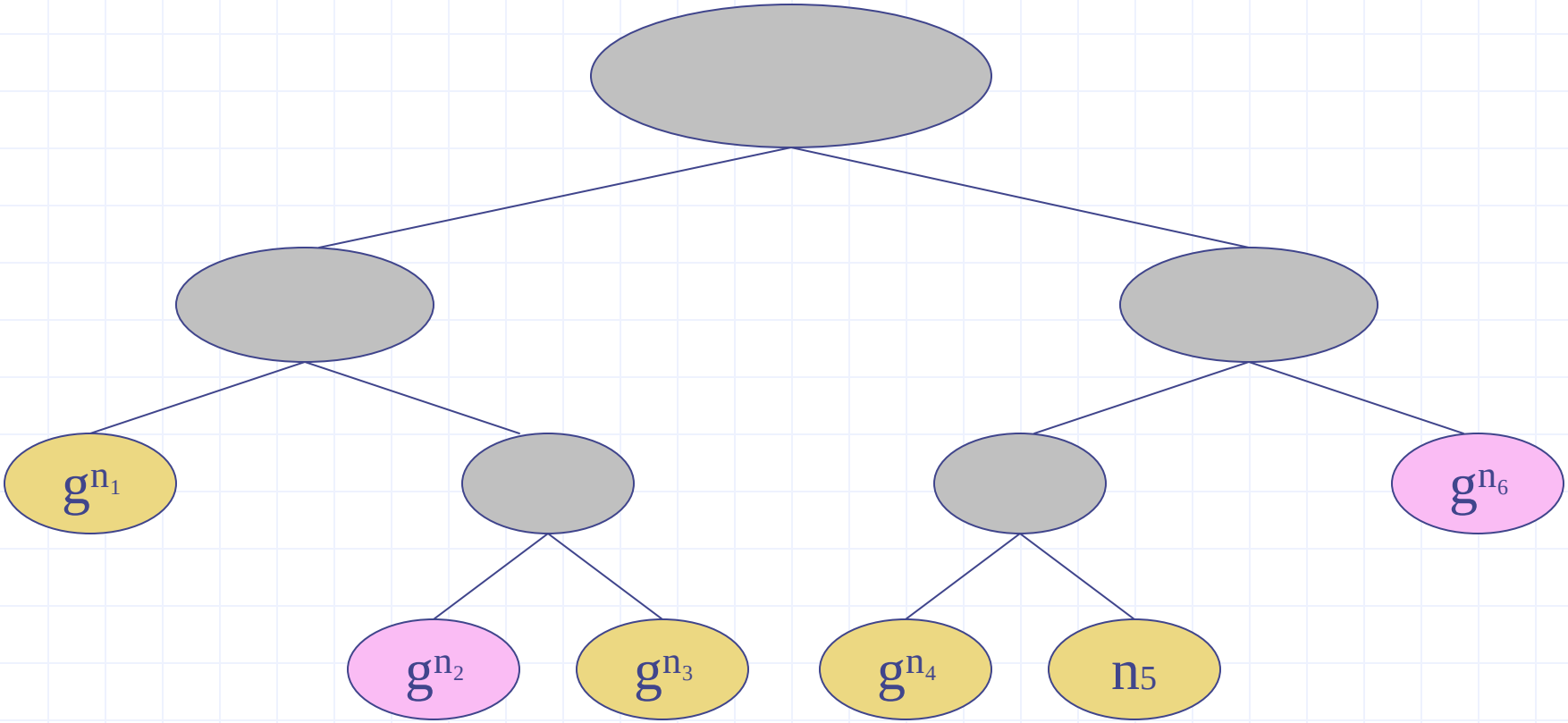
Partition(n_5's view)



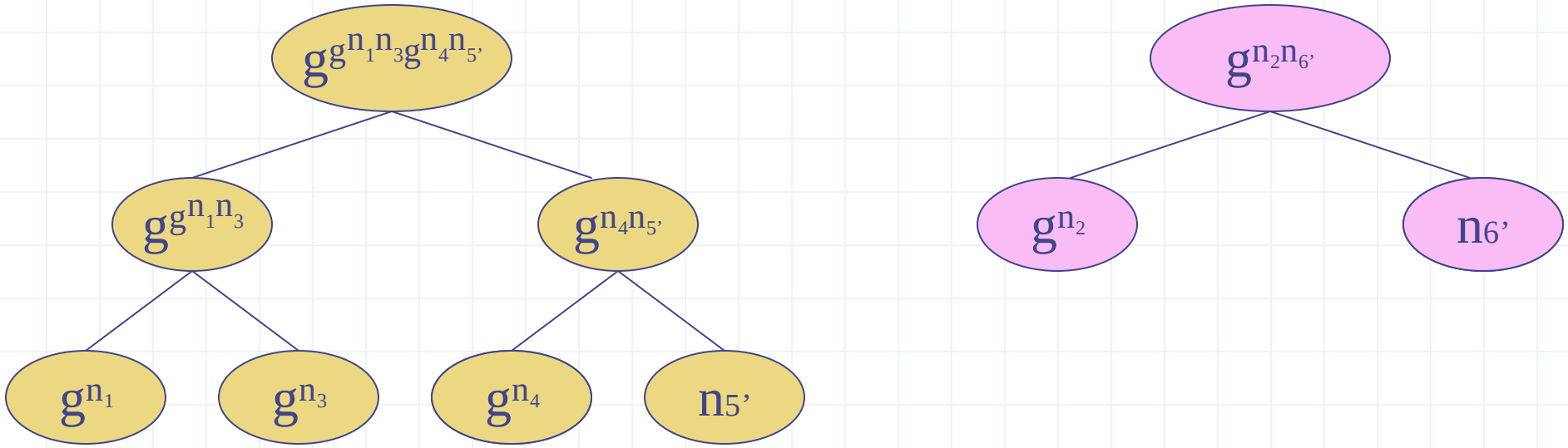
Partition(n_5's view)



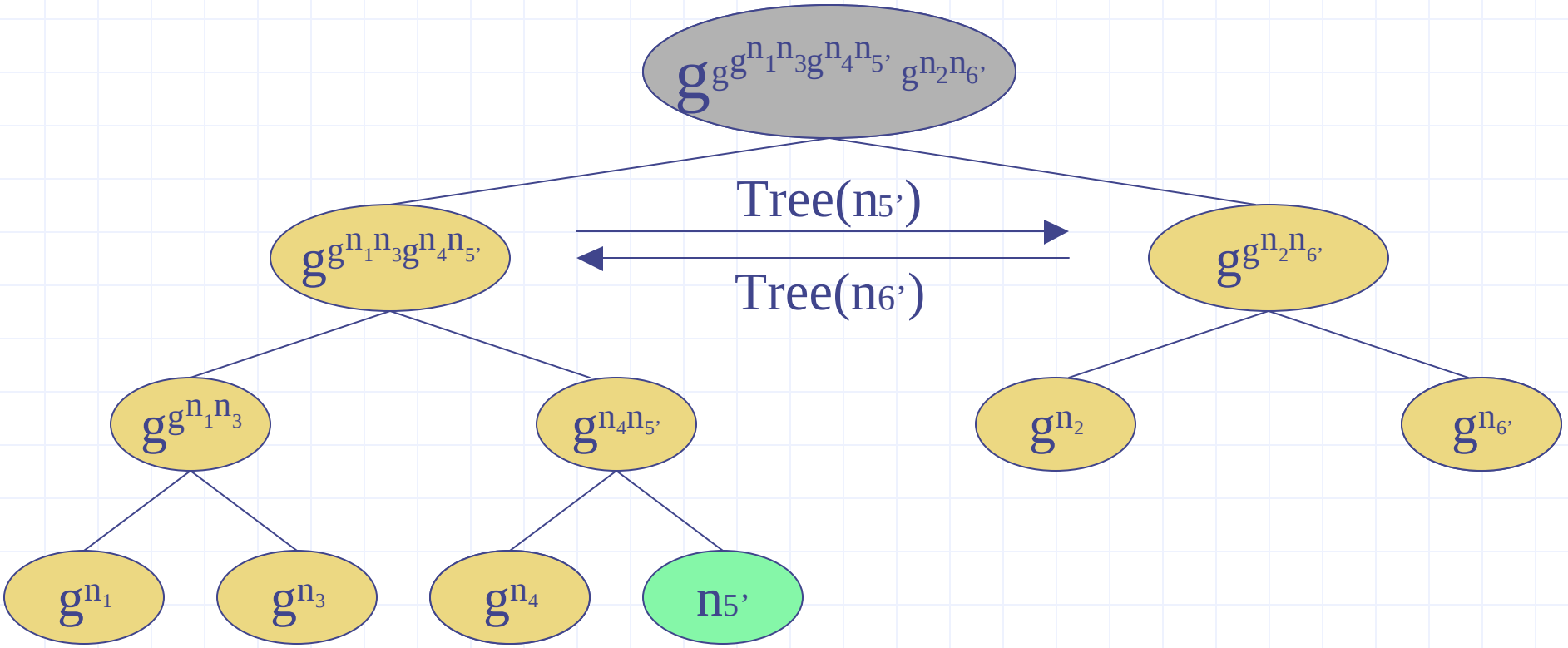
Partition: Both Sides



Partition: Both sides(N_5 and N_6's view)



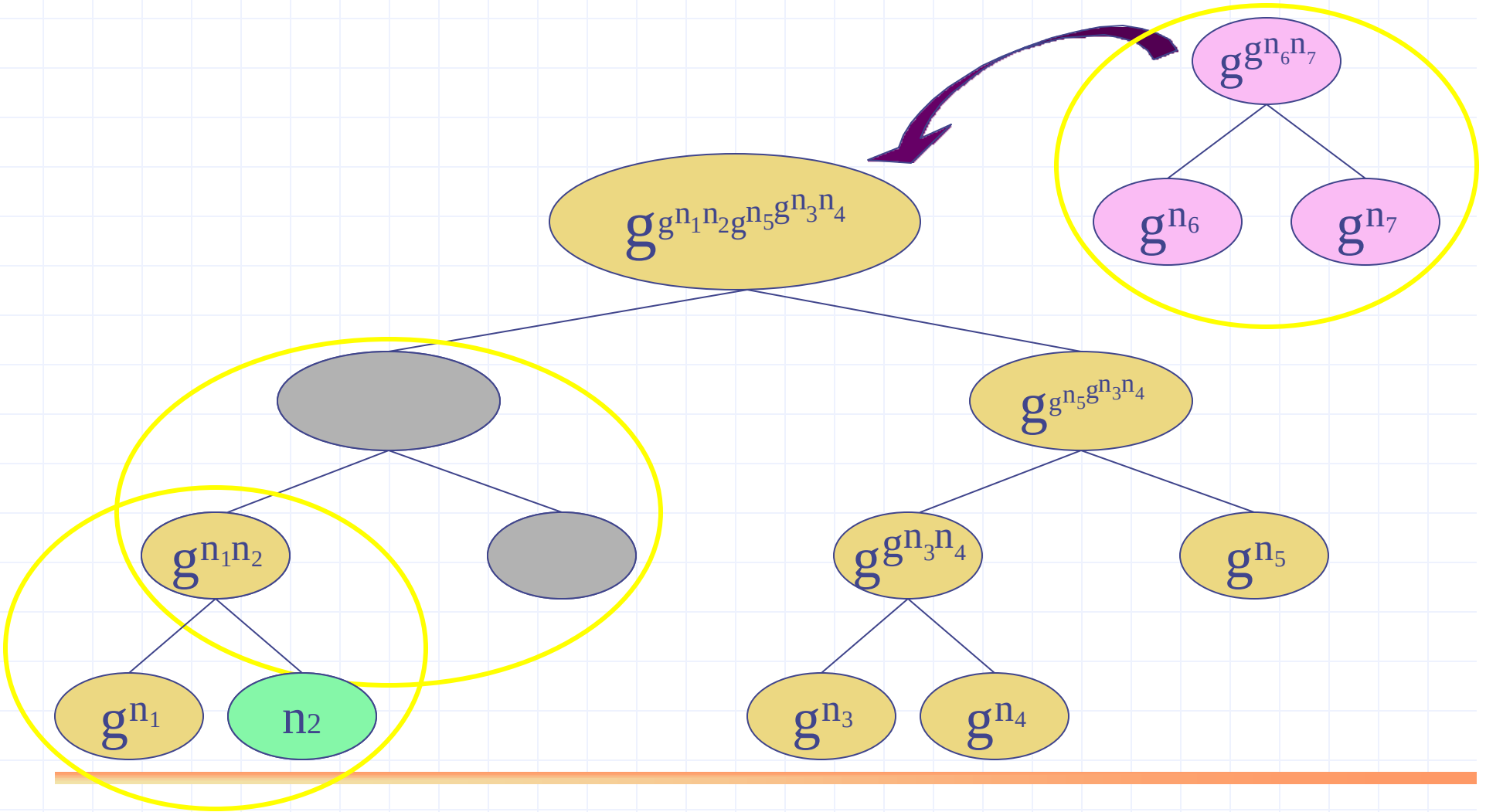
Merge(N_5's view)



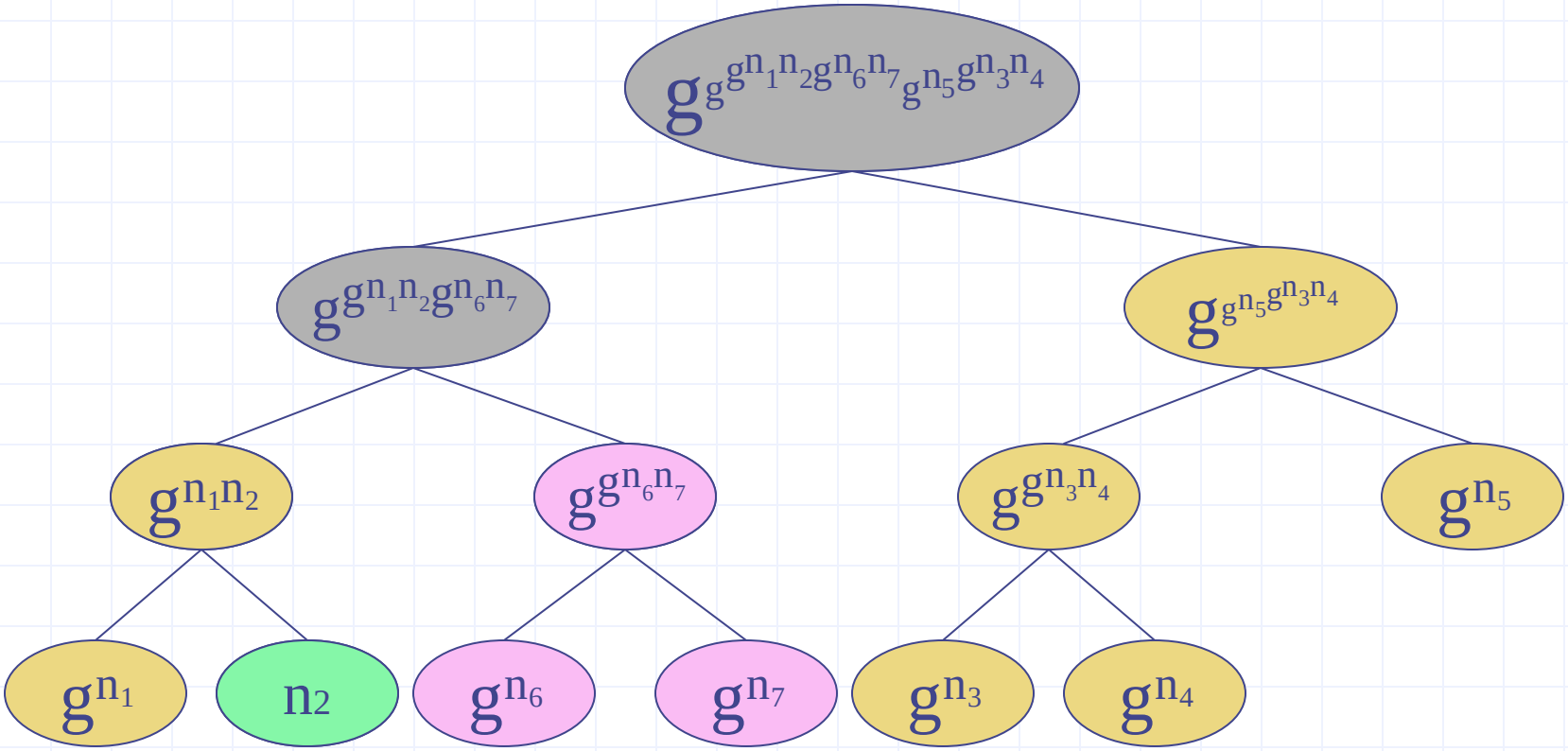
No nodes to merge without increasing height of the tree

Join to root

Merge(to intermediate node, N_2's view)



Merge (to intermediate node)



Comparison

		Comm				Comp	Robust
		Round	Msg	Uni	Broad	Exp	
CLQ	Join	2	2	1	1	2n	Hard
	Leave, Partition	1	1	0	1	n	
	Merge	k+3	n+2k+1	n+2k-1	2	n+2k	
TGDH	Join, Merge	2	3	0	3	2log n	Easy
	Leave	1	1	0	1	log n	
	Partition	log n/2	log n	0	log n	log n	
STR	Join	1	2	1	1	2	Easy
	Leave, Partition	1	1	0	1	2n	
	Merge	2	3	2	1	2	
BD		2	2n	2	2n	3	Easy

Key Distribution vs. Agreement

	Key Distribution	Key Agreement
Key Generation	Center	Each member's contribution
Crypto Primitive	Secret key Encryption Hash and MAC	Extended Diffie-Hellman
Communication	IP Multicast	Group communication
Computation Overhead	Small(Large for center)	Large(Similar complexity)
Group Size	> 10,000	< 100
Contributory	No	Yes
Number of round	Single(no acks)	Multiple
Fault-tolerance	Supported(1 round)	Required
Single point of attack?	Yes	No