Communication-efficient Group Key Agreement

June 20, 2001

Gene Tsudik, UC Irvine
gts@ics.uci.edu

Joint work with:

- Adrian Perrig, CMU/UC Berkeley
- Yongdae Kim, USC/UC Irvine

Outline

- Definitions/concepts
- Related work
- Background/Motivation
- Protocols

Group Communication Settings

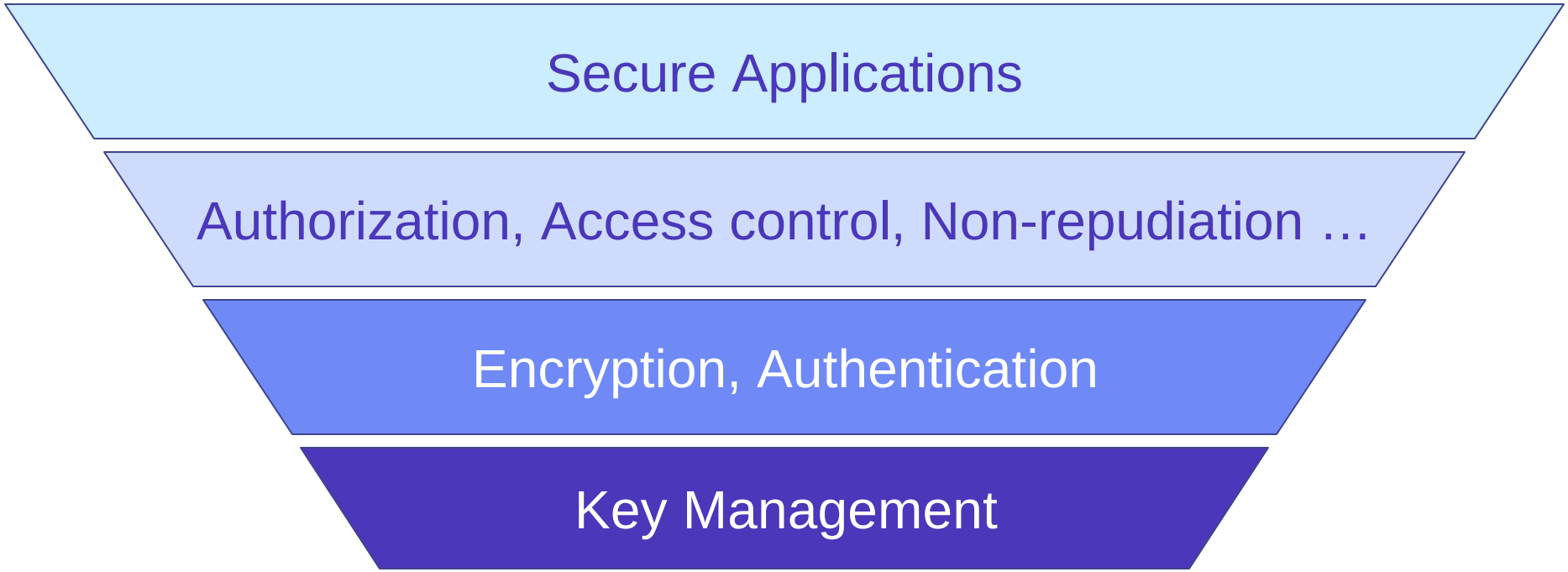
- Few-to-Many
  - Single-source broadcast: Cable/sat. TV
  - Multi-source: Televised debates, GPS
  - In general, Internet-style IP multicast
- Any-to-Any
  - Collaborative applications (peer groups)
  - Video/Audio conferencing, collaborative workspaces, interactive chat, network games, distributed database replication, etc.
  - Rich communication semantics, tighter control, more emphasis on synchronization, reliability and **security**

Dynamic Peer Groups (DPG)

- Relatively small (<100 members)
- No hierarchy
- Frequent membership changes
- Any member can be sender and receiver

Our focus: key management in DPGs

Key Management is a building block

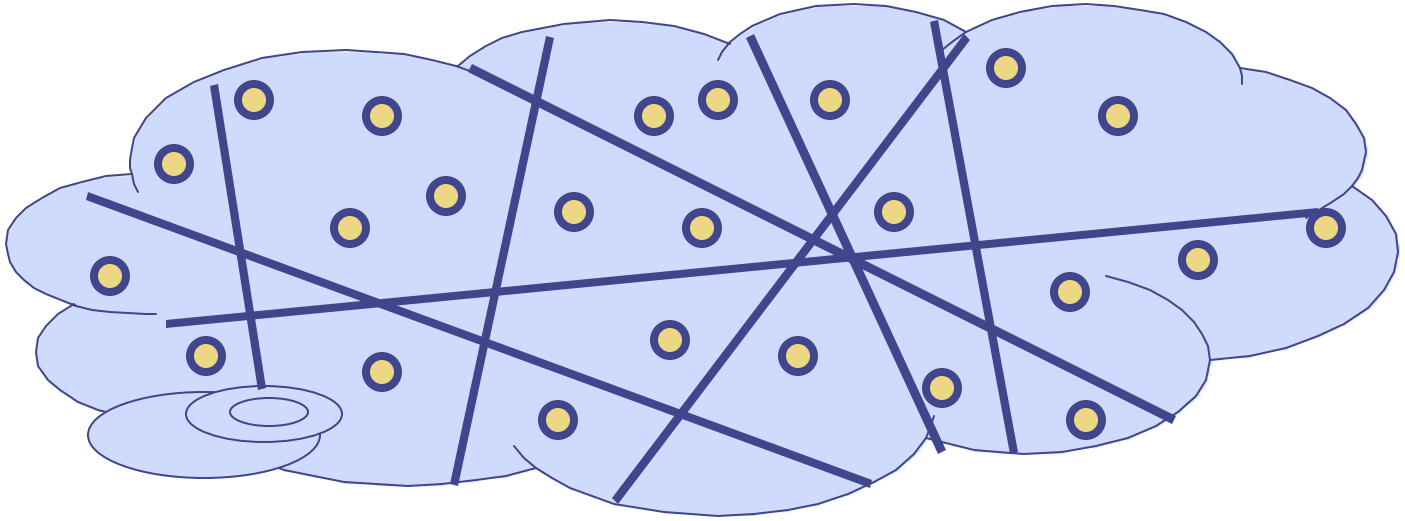


Group Key Management

- Group key: a secret quantity known only to current group members
- Group Key Distribution
  - One party generates a secret key and distributes to others.
- Group Key Agreement
  - Secret key is derived jointly by two or more parties.
  - Key is a function of information contributed by each member.
  - No party can pre-determine the result.

Key Distribution in DPG?

- Centralized key server
  - Single point of failure
  - Attractive attack target
- Can key server be sufficiently replicated?
  - Must be available in all possible partitions
  - Network can have arbitrary faults (eg, ad hoc)

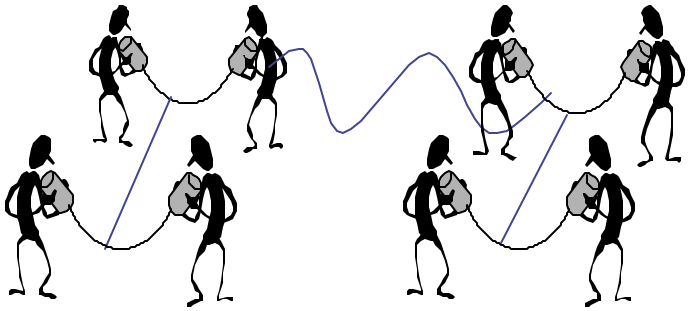


Need for Reliable Group Communication

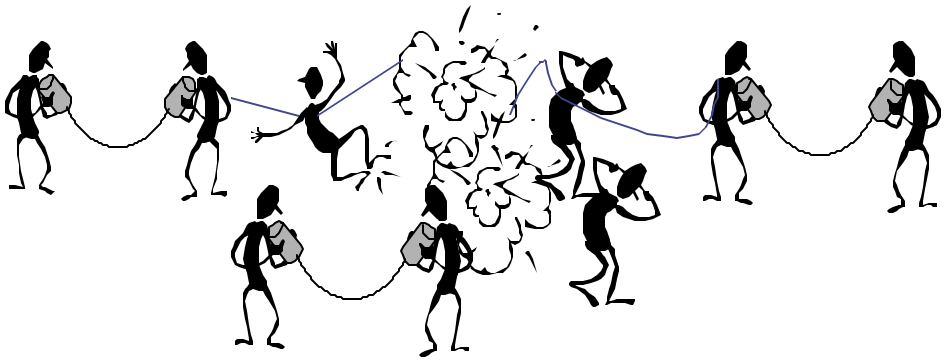
- Group key agreement protocols rely on the underlying group communication systems.
  1. Protocol message transport
  2. Strong membership semantics (notification of a group membership)
     - Not for security reasons
- Group communication system needs specialized security mechanisms.

Mutual benefit and interdependency

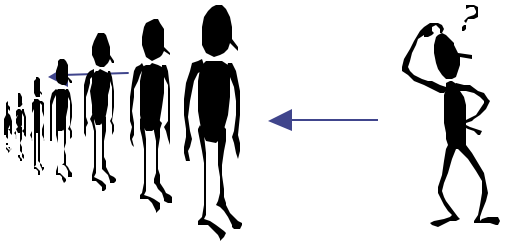
Membership Operations



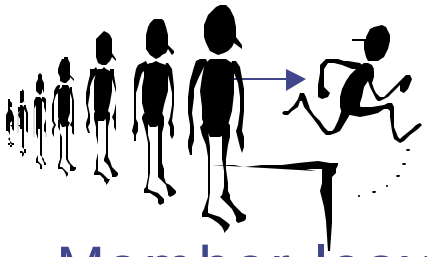
Formation ???



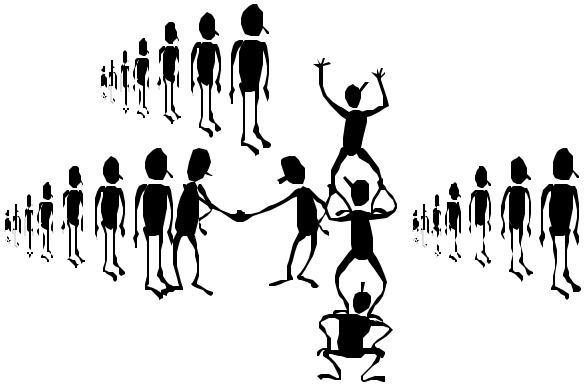
Group partition



Member join



Member leave



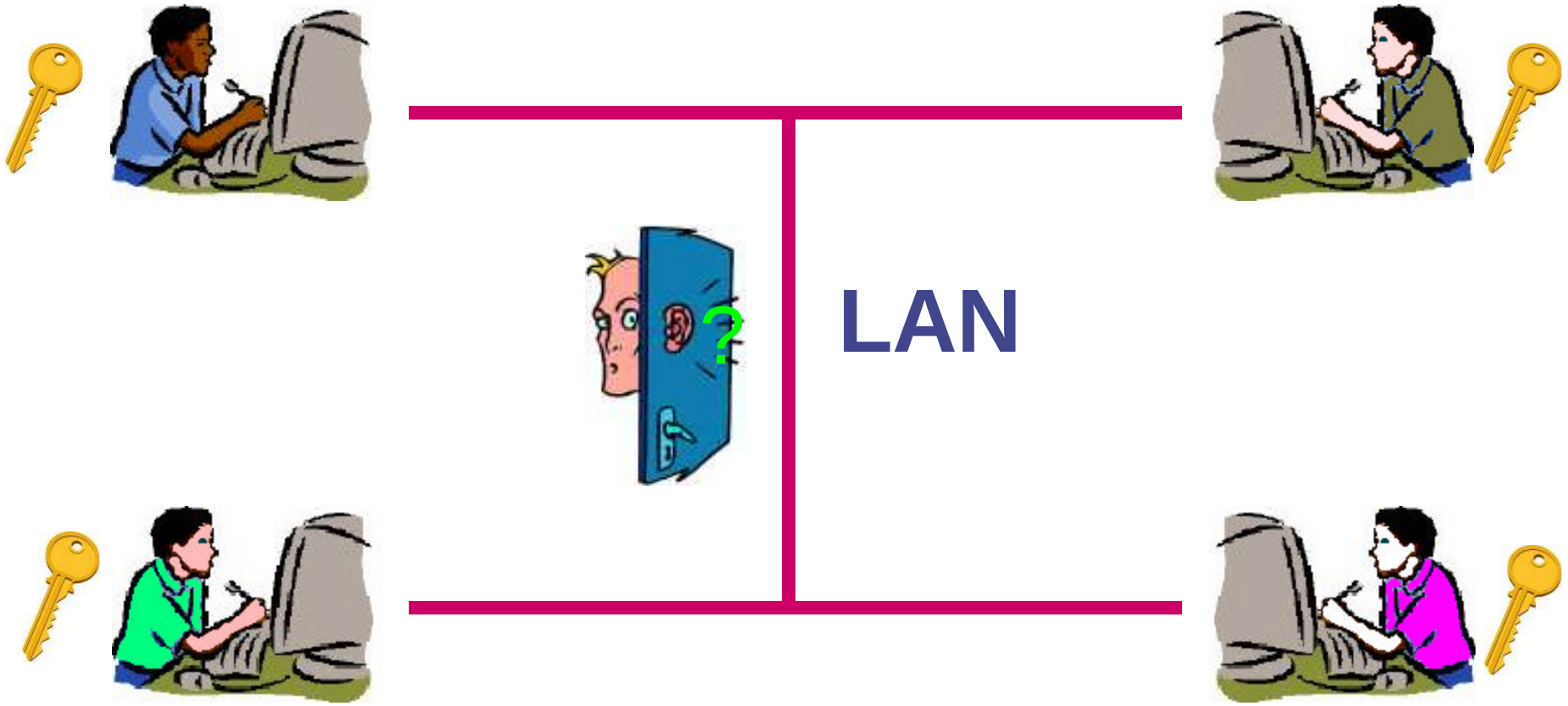
Group merge

Motivation

- need group key agreement with:
  - Strong security
  - Support for dynamic membership
  - Robustness
  - Efficiency in
    - **communication**

      and
    - **computation**

Common DPG setting



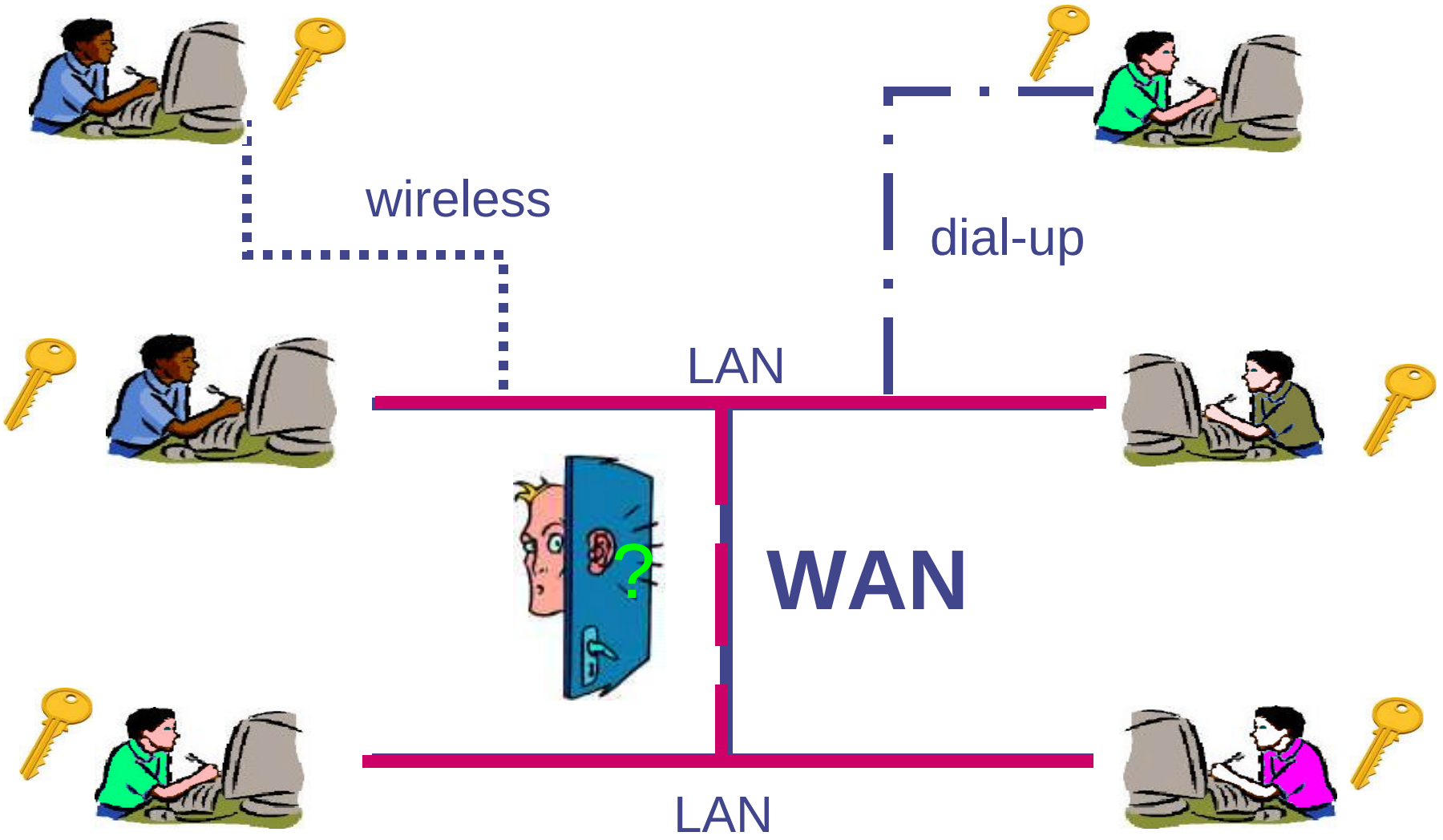
Computation overhead

- Most group key agreement methods involve modular exponentiation.

1024-bit mod exp	
Pentium II 450	8 ms
Pentium III 800	4 ms
Sun Ultra 250	20 ms

- Contrast with typical LAN roundtrip delay < 2ms
- On paper, communication overhead is negligible
- Number of protocol rounds?

Another DPG setting



Motivation: minimize rounds and messages

- Over WAN (and wireless, dial-up, etc.) communication is more expensive than computation
- Communication has an upper bound (speed of light)
  - Computation speed increases much fast than communication
- Too many messages → some might be lost/corrupted
  - Retransmissions
- Many rounds → cascaded events (protocol interruption)

Communication roundtrip (Ping)	
UCI ↔ Columbia U	88 ms / 20(ls)
UCI ↔ Thailand	420 ms
UCI ↔ Mozambique	670 ms

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 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

Functional Requirements

- Minimize communication and round complexity
- Robustness against cascaded failures
- Maintain strong security, of course…

Related Work

- Focused mainly on security and/or computation overhead
- Diffie-Hellman extensions
  - Burmester and Desmedt (BD, 1993): fast comp-n, many broadcasts
  - Steiner et al. (Cliques, 1996): slow join, fast leave
  - Becker and Wille (BW 1998): *log n* rounds, hi computation overhead
  - Tzeng and Tzeng (1999, 2000): fast but not secure

Related Work (Cont.)

- TGDH (Tree-based Group Diffie-Hellman)
  - Y. Kim, A. Perrig and G. Tsudik
  - ACM CCS 2000
- STR (A Secure Audio Conference System)
  - D. Steer, L. Strawczynski, W. Diffie and M. Wiener
  - CRYPTO'88
    - Static groups
    - No security proof

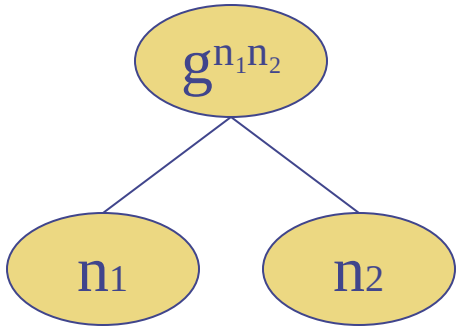
What we do

- Extend STR to dynamic groups
- Security
- Analyze, implement, integrate

Diffie-Hellman

- Setting
  - p – large prime (e.g. 512 or 1024 bits)
  - Zp* = {1, 2, … , p – 1}
  - g – base generator

- $A \rightarrow B : N_A = g^{n1} \bmod p$
- $B \rightarrow A : N_B = g^{n2} \bmod p$
- $A : N_B{}^{n1} = g^{n1n2} \bmod p$
- $B : N_A{}^{n2} = g^{n1n2} \bmod p$



- Diffie-Hellman Key : $g^{n1\ n2}$
- Blinded Key of n1 : $N_A = g^{n1} \bmod p$

Diffie-Hellman Problem

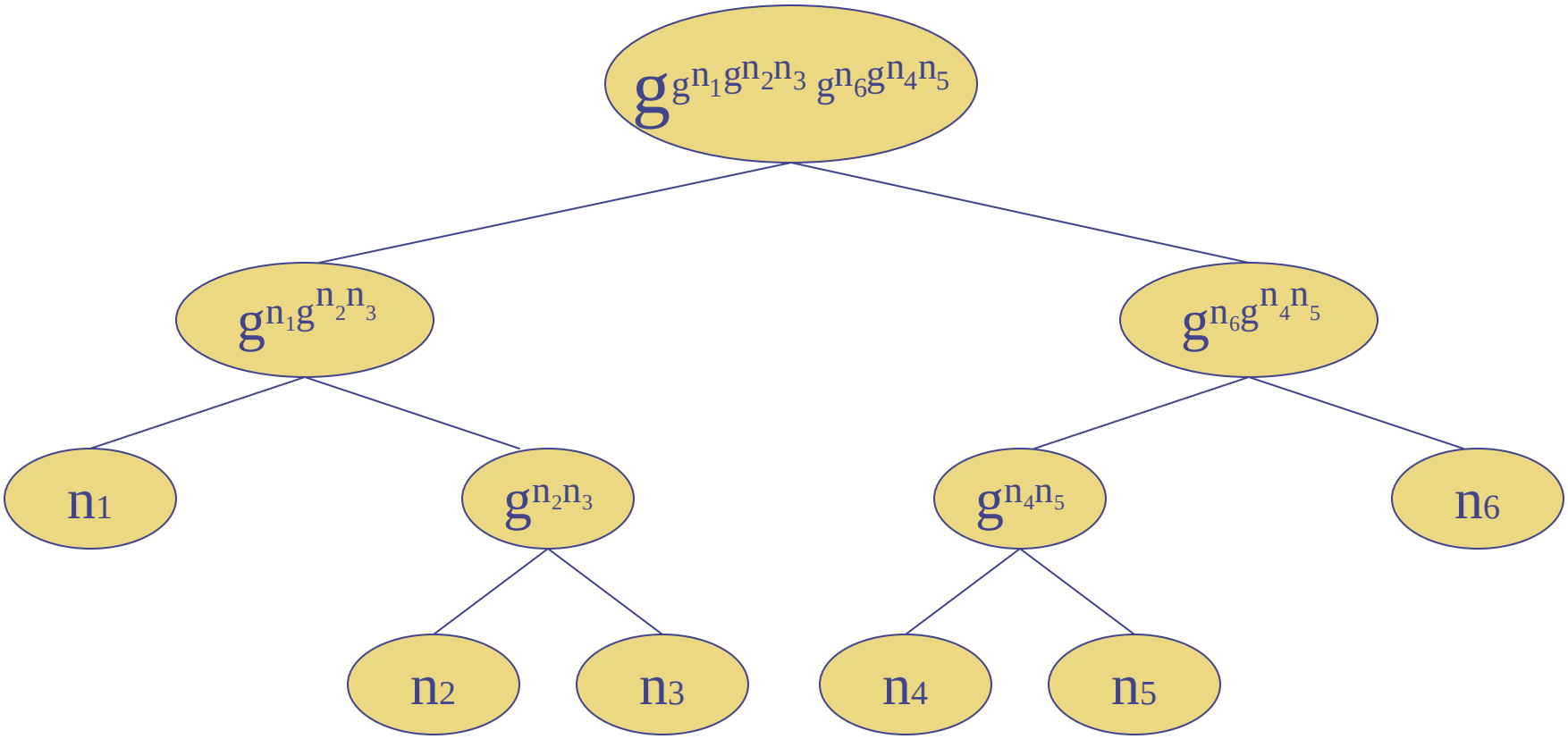
- Computational Diffie-Hellman Assumption (CDH)
  - Loose Definition: Given g^a, g^b, computing g^{ab} is hard.
  - CDH is not sufficient to prove that Diffie-Hellman Key can be used as secret key.
    - Eve may recover part of information with some confidence
    - One cannot simply use bits of g^{ab} as a shared key
- Decision Diffie-Hellman Assumption (DDH)
  - Loose Definition

    Given g^a and g^b, and a guess g^c, check if $g^c = g^{ab}$
  - Stronger than CDH

TGDH

- Simple: all membership operations in a single function
- Fault-tolerant: robust against cascaded faults
- Secure
  - Contributory
  - Provable security
  - Key independence
- Efficient
  - d is the height of key tree ($O(\log_2 N)$), N is the number of users
  - Maximum number of exponentiation = 4(d-1)

Key Tree (General)



Security

- Group key secrecy T-DDH
  - Intuitive Definition

    Given all blinded keys of a random key tree, can we distinguish the group key from a random number?
- Proof goal

  If we can solve T-DDH, we can solve 2-party DDH.
- Key independence.
  - One member changing its contribution upon every event

Features

- Efficiency
  - Avg number of mod exp: $2 \log_2 n$
  - Max number of rounds: $\log_2 n$
- Robustness easy thanks to self-stabilization property
- Tree structure a bit complex

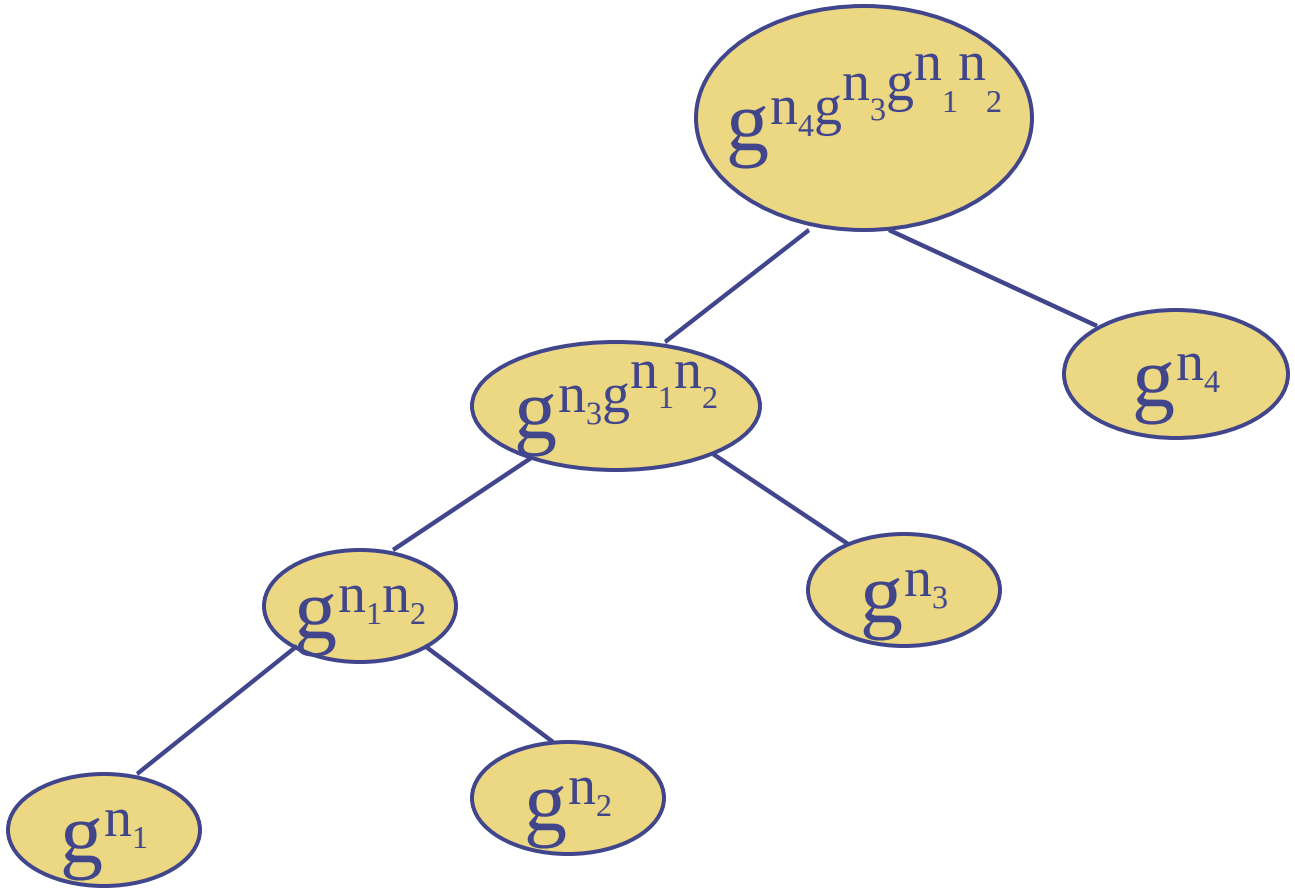
Goal:

- Group key agreement scheme with:
  - small number of rounds
  - small number of messages
  - in return for more computation

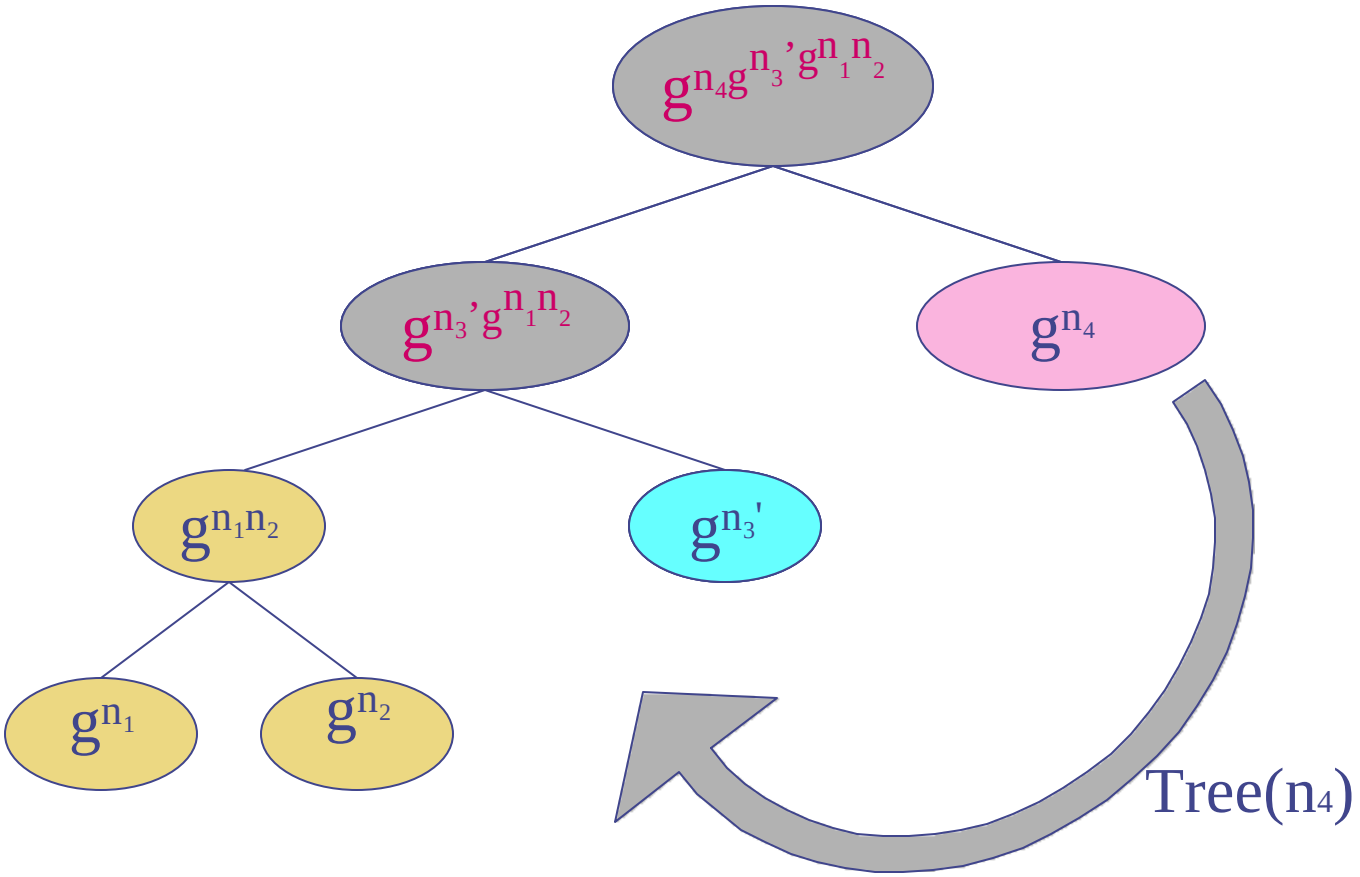
STR

- Communication efficient (not in original form)
  - Max 2 rounds
  - Max 2 broadcasts
- Simple: implemented as one function
- Fault-tolerant: Easier than TGDH
- Secure
  - Contributory
  - Provable security
  - Backward and forward secrecy => key independence
  - Provable security
- Computation cost is higher (for leave/partition)
  - Max # exponentiations z3(N-1) avg = 3N/2
  - Low for join/merge

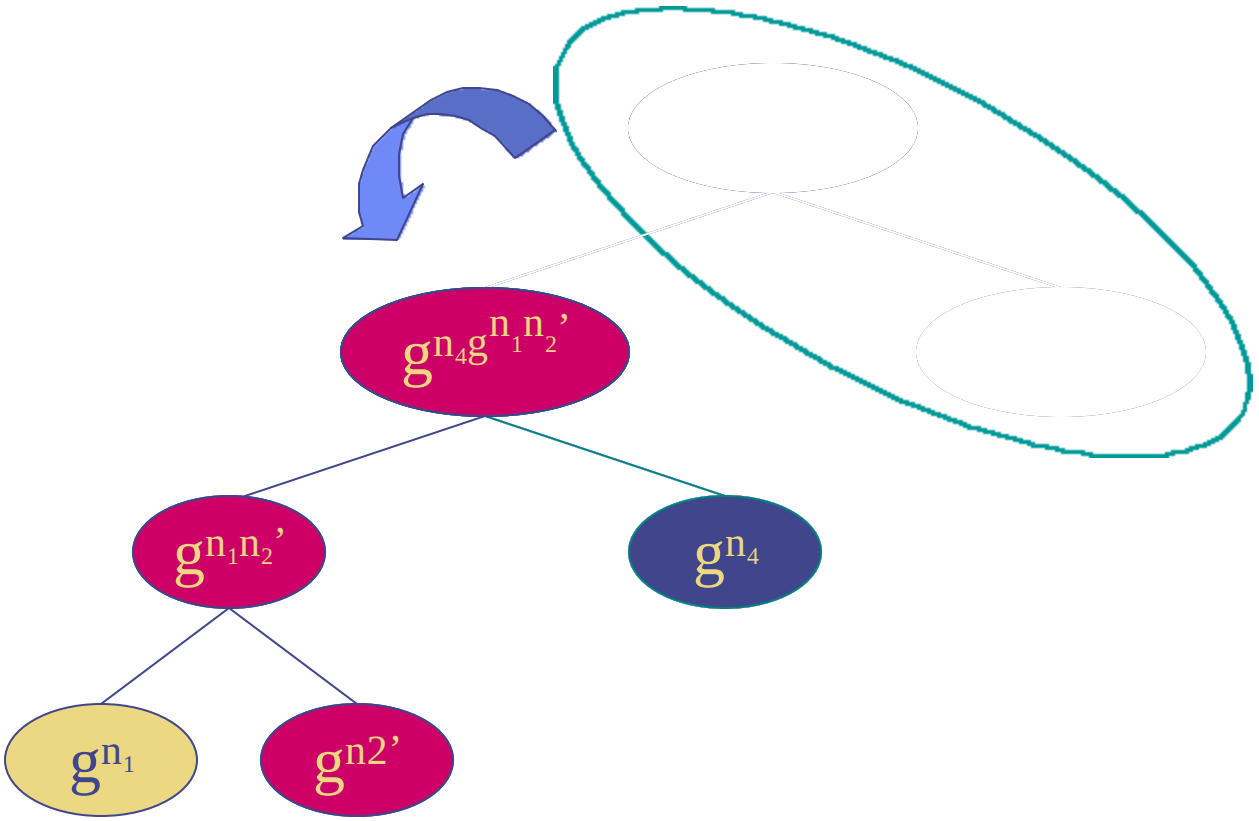
STR Key Tree



Join (Merge similar)



Leave or Partition



Features

- Security
  - Same as TGDH
- Efficiency
  - mod exp: 2 – join, 1.5n – leave
  - number of rounds: 1 – join, 1 – leave
  - number of messages: 2 – join, 1 – leave
- Robustness is provided by self-stabilization property

Comparison

		Comm				Comp	Robust
		Rounds	Msgs	Uni	Broad	Exp	
Cliques IKA.2	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	1.5n	
	Merge	2	3	2	1	2k	
BD		2	2n	0	2n	3	Easy

Finally…

Code available, part of Cliques distribution

- **STR**
- TGDH
- CKD
- BD
- GDH IKA.1
- GDH IKA.2

- http://sconce.ics.uci.edu
- Standalone or integrated with Spread group communication toolkit
- Questions?