Lecture 14
ICS 268

Group Key Agreement

November 17, 2001

Presented by Yongdae Kim

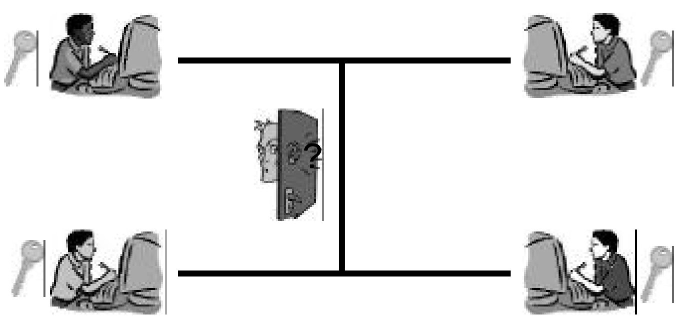
1

Outline

- Definitions and concepts
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

2/67

Background



3/67

Group Communication Settings

- Few-to-Many
  - Single-source broadcast: Cable/sat. TV, radio
  - Multi-source broadcast: Televised debates, GPS
- Any-to-Any
  - Collaborative applications need inherently underlying peer groups.
  - Video/Audio conferencing, collaborative workspaces, interactive chat, network games and gambling
  - Rich communication semantics, tighter control, more emphasis on reliability and **security**

4/67

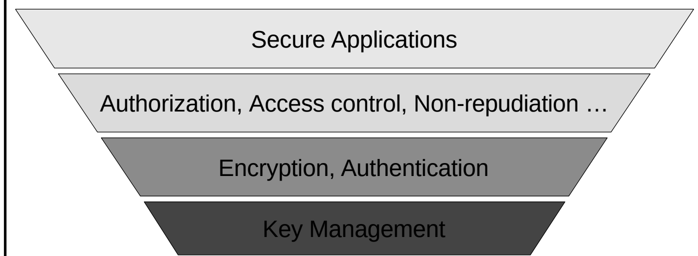
Dynamic Peer Groups (DPG)

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

My focus: key management in DPGs

5/67

Key Management is a building block



6/67

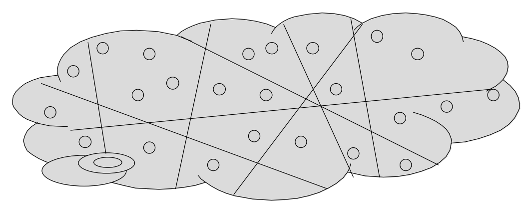
Group Key Management

- 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.

7/67

Can we use Key Distribution in DPG?

- Centralized key server
  - Single point of failure
  - Attractive attack target
- Can key server be sufficiently replicated? $\Rightarrow$ Very costly
  - Availability of a key server in any and all possible partitions
    - Network can have arbitrary faults!



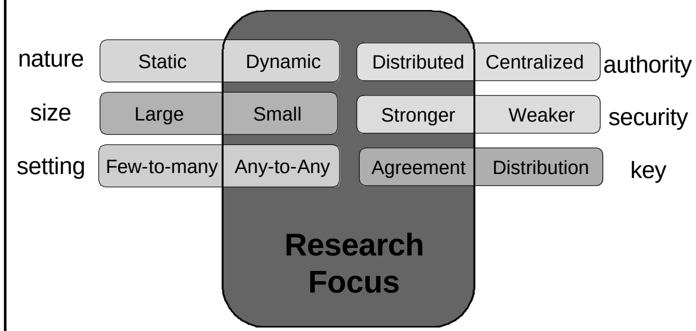
8/67

Distribution vs. Agreement

	Key Distribution	Key Agreement
Key Generation	Center	Each member's contribution
Crypto Primitive	Secret key Encryption Hash/MAC function	Extended Diffie-Hellman
Communication	Multicast or Unicast	Group communication
Computation Overhead	Small(Large for center)	Large(Similar complexity)
Group Size	> 10,000	< 100
Contributory	No	Yes
Number of round	Single	Multiple
Example	Wong and Lam OFT(McGrew, Sherman) IBM(Canetti et. al.)	BD(Burmester and Desmedt) GDH(Tsudik et. al.) TGDH(Kim et. al.) STR(Kim et. al.)

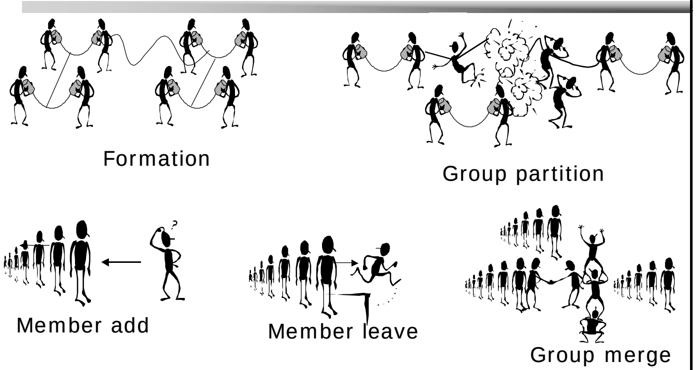
9/67

Settings for Group Key Management



10/67

Membership Operations



11/67

Membership Operations

- Join: a prospective member wants to join
- Leave: a member wants to (or is forced to) leave
- Partition: a group is split into smaller groups
  - Network failure: network event causes disconnectivity
  - Explicit partition: application decides to split the group
- Merge: two or more groups merge to form a single group
  - Network fault heal: previously disconnected partitions reconnect
  - Explicit merge: application decides to merge multiple pre-existing groups into a single group

12/67

Motivation

- We need group key agreement methods satisfying the following:
  - Strong security
  - Dynamic operation
  - Robustness
  - Efficiency in communication and computation
  - Implementation, integration, and measurement

Why care about computation overhead?

- Most group key agreement methods rely on modular exponentiation.
  - 512 bit modular exponentiation on Pentium 400 Mhz = 2 msec
  - 1024 bit modular exponentiation = 8 msec
- Most methods require a lot of modular exponentiations for each membership operation.
  - Cliques: When current group size is *n* , join of a member to this group requires *2 n + 1* modular exponentiation.

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

Outline

- Definitions and notions
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

Related Work

- Cliques
  - *Key Agreement in Dynamic Peer Groups (1996, 1997, 2000)*
    - Steiner, Tsudik and Waidner
    - Group Diffie-Hellman key agreement protocols
    - Dynamic membership operations
  - *New Multi-party Authentication Services and Key Agreement Protocols (1998, 2000)*
    - Ateniese, Steiner and Tsudik
    - A notion of group key authentication is considered
  - Drawbacks
    - Slow computation: O(n) computation for each membership event
    - Communication overhead: k rounds for merge (k: # of new members)

17/67

Related Work (Continue)

- TGDH (Tree-based Group Diffie-Hellman)
  - Y. Kim, A. Perrig, G. Tsudik
  - ACM CCS 2000, Nov. 2000
  - Computation overhead reduced from O(n) to O(log n)
  - Providing robustness against cascaded failure inherently
- STR
  - Y. Kim, A. Perrig, G. Tsudik
  - IFIP SEC 2001, accepted to publication
  - Communication overhead is lower than any other methods

18/67

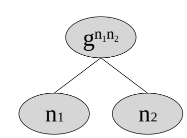
Outline

- Definitions and notions
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

19/67

Diffie-Hellman

- Setting
  - p – large prime (e.g. 512 or 1024 bits)
  - Zp* = {1, 2, … , p – 1}
  - g – base generator
- $A \rightarrow B : N_A = g^{n1}$ mod p
- $B \rightarrow A : N_B = g^{n2}$ mod p
- $A : N_B^{\ n1} = g^{n1n2}$ mod p
- $B : N_A^{\ n2} = g^{n1n2}$ mod p



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

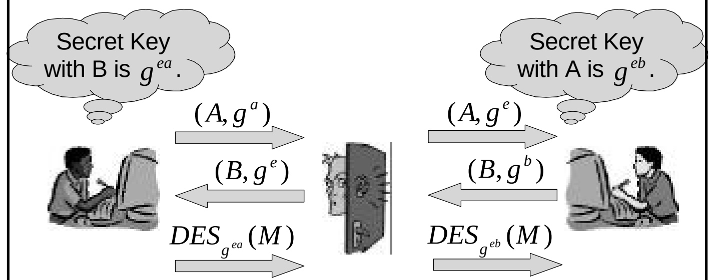
20/67

Diffie-Hellman Problem

- Computational Diffie-Hellman Assumption (CDH)
  - Loose Definition: Having known 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
    Knowing g^a and g^b, and guessing g^c, can you check $g^c = g^{ab}$?
  - Stronger than CDH

21/67

Man-in-the-Middle Attack for DH



Authentication is required

22/67

Authenticated Diffie-Hellman

- Implicit Authentication
  - Using Long-term Key
- Explicit Authentication
  - Using signature or MAC

23/67

Authenticated Diffie-Hellman

- A
  - Public Key = g^A, secret key = A (Long-term Key)
  - Computes $K_{AB} = g^{AB}$
  - Generates a, computes $g^{a K_{AB}}$
  - A ⇒ B: $g^{a K_{AB}}$
- B
  - Public Key = g^B, secret key = B
  - Computes $K_{AB} = g^{AB}$
  - Generates b, computes $g^{b K_{AB}}$
  - Computes $K = g^{ab} = (g^{a K_{AB}})^{K_{AB}^{-1} b}$
  - B ⇒ A: $g^{b K_{AB}}$
- A can compute $K = g^{ab} = (g^{b K_{AB}})^{K_{AB}^{-1} a}$

24/67

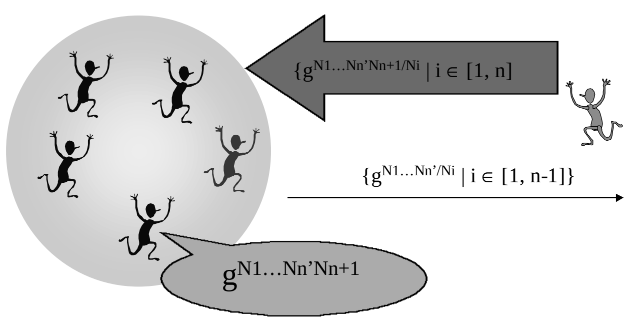
Outline

- Definitions and notions
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

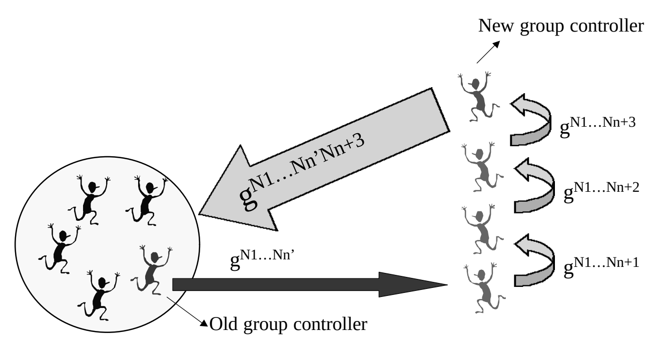
Background Intuition

- What should be the natural extension of Diffie-Hellman protocol to n members?
  - What will be the form of group key?
    $g^{N_1N_2\ldots N_n}$ where N_i is member i's secret share
  - Which information is required to compute the group key for each member i?
    $g^{N_1N_2\ldots N_n/N_i}$
  - How can we build this information?

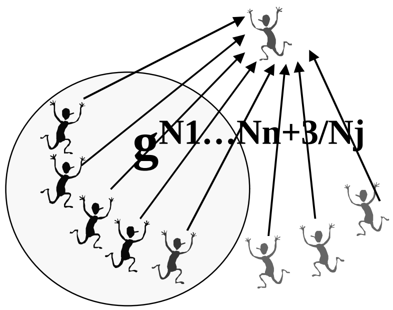
Joins



Merge I

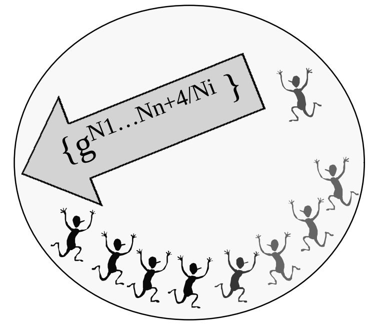


Merge II



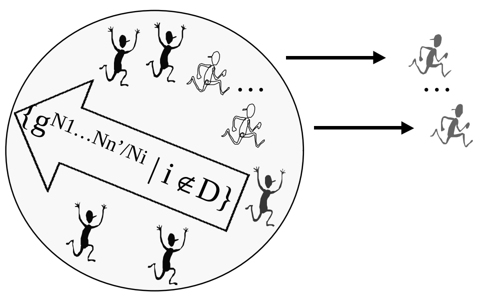
29/67

Merge III



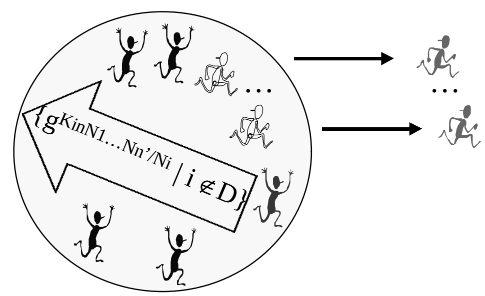
30/67

Leave or Partition



31/67

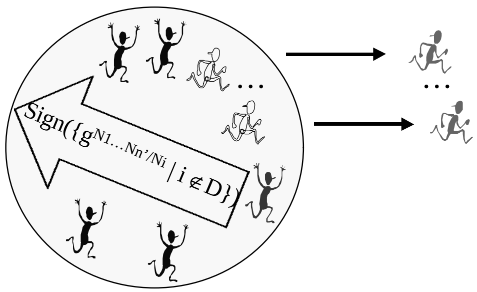
Authenticated GDH (Implicit, A-GDH)



32/67

Authenticated GDH (Explicit)

- Using signature or MAC
- Current Implementation uses signature



33/67

Discussion

- Security
  - Equivalent to 2-Party Decision Diffie-Hellman problem: If we can differentiate Cliques group key with a random number, then we can differentiate 2-party Diffie-Hellman key with a random number
- Efficiency
  - O(n) computation
  - k+3 communication round
- Robustness
  - What if a token lost?
  - Complex steps are required to achieve robustness against cascaded failure.

34/67

Outline

- Definitions and notions
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

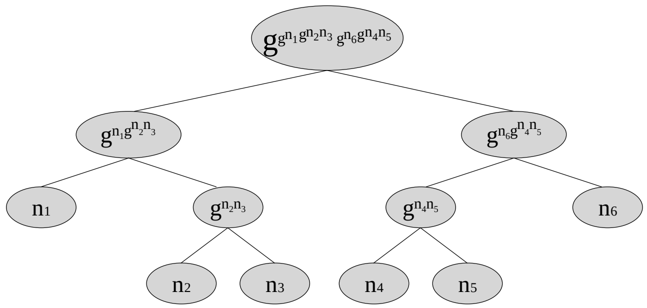
35/67

TGDH

- Simple: One function is enough to implement it
- 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)
  - # of exp. in Cliques = 2N+1

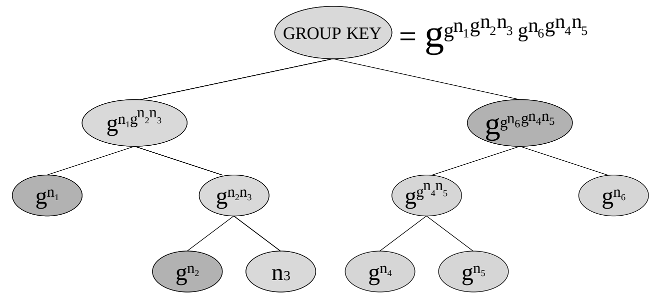
36/67

Key Tree (General)



37/67

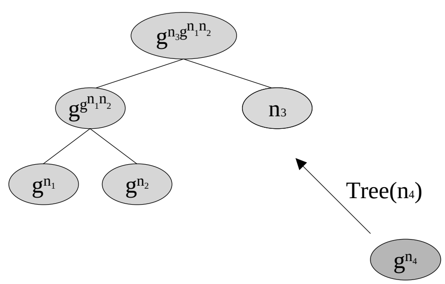
Key Tree (n_3's view)



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

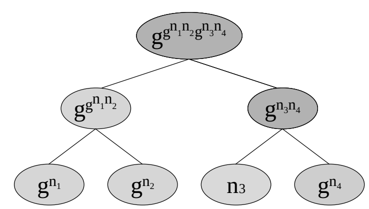
38/67

Join (n_3's view)



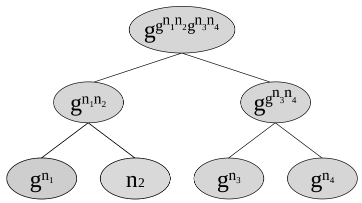
39/67

Join (n_3's view)



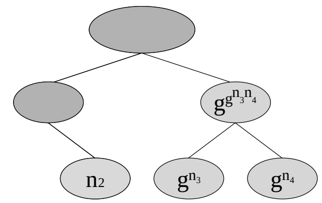
40/67

Leave (n_2's view)



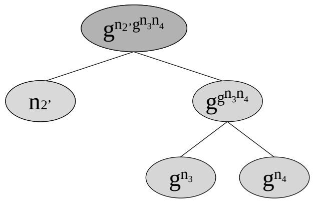
41/67

Leave (n_2's view)



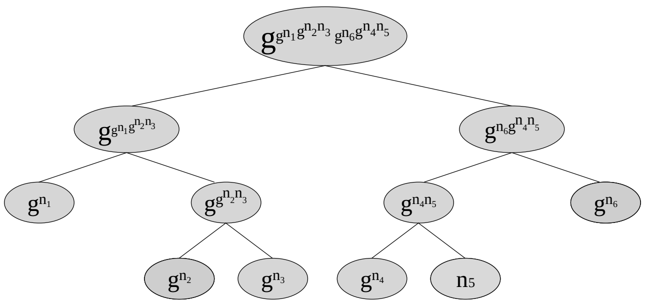
42/67

Leave (n_2's view)



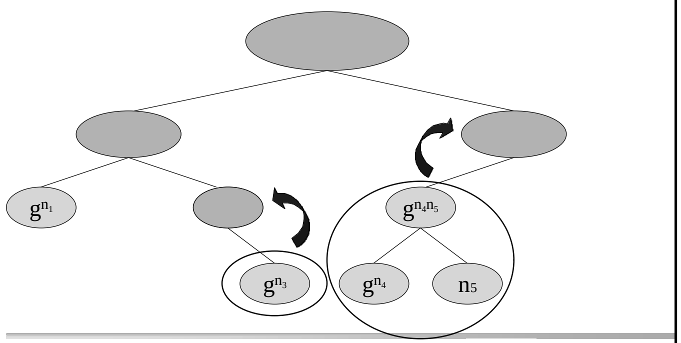
43/67

Partition (n_5's view)



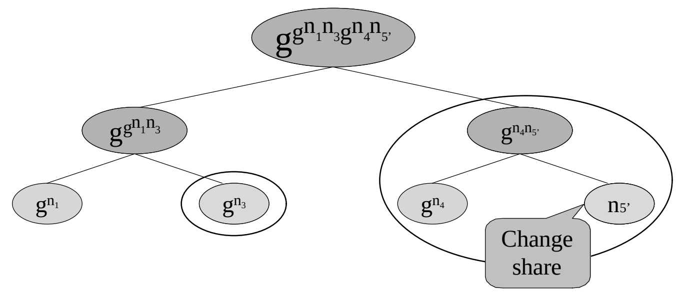
44/67

Partition (n_5's view)



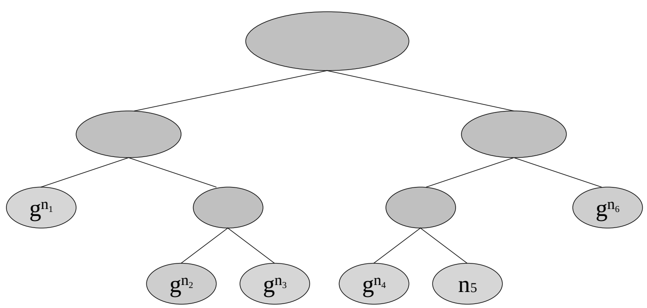
45/67

Partition (n_5's view)



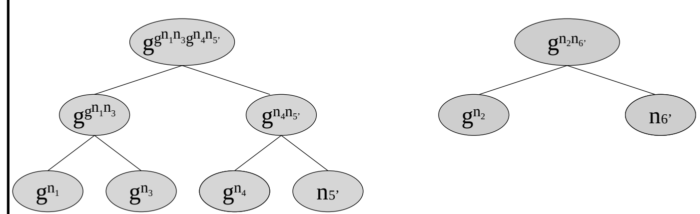
46/67

Partition: Both Sides



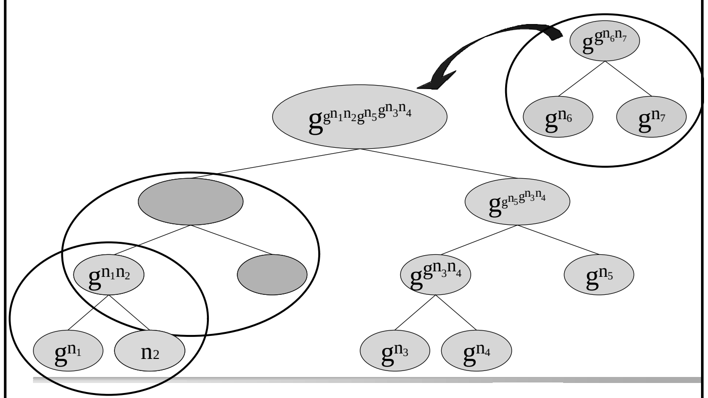
47/67

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



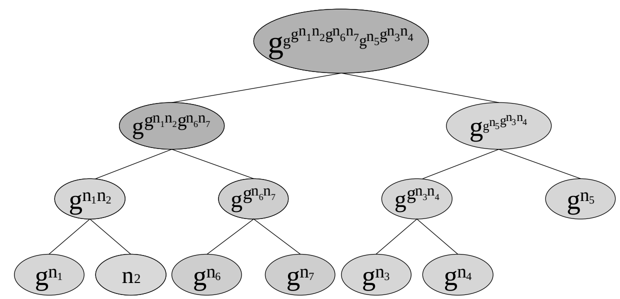
48/67

Merge (to intermediate node, N2's view)



49/67

Merge (to intermediate node)



50/67

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 be partitioned out.
  - No policy for leave or partition event
- Successful
  - Still maintaining logarithmic (height < 2 $\log_2$ N)
- Future Work
  - Other tree management technique
  - Rebalancing

51/67

Security

- Group key secrecy
  - Intuitive Definition

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

  If we can distinguish, we can distinguish 2-party DDH.
- We can provide key independence.
  - By changing session random of a member on every additive event

52/67

Discussion

- Efficiency
  - Average number of mod exp: $2 \log_2 n$
  - Maximum number of round: $\log_2 n$
- Robustness is easily provided due to self-stabilization property

53/67

Outline

- Definitions and notions
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

54/67

STR

- Communication efficient
  - Maximum 2 communication round
  - Maximum 2 broadcast messages
- Simple: One function is enough to implement it.
- Fault-tolerant: Easier than TGDH
- Secure
  - Contributory
  - Backward and forward secrecy
  - Provable security
  - Key independence
- Computation is bit more expensive.
  - Maximum number of exponentiation = 4(N-1)
  - N is the number of users.

55/67

Motivation

- Over WAN, communication is much more expensive than computation
  - Multi-round protocol is slow
- Communication always has upper bound (speed of light)
  - Computation speed increases much fast than communication
- Too many messages are also bad
  - May require retransmission

Computation(1024 DSA signature)		Communication (Ping)	
Pentium 800 Mhz	0.0037 secs	UCI ↔ Columbia univ.	0.0884 secs
Sun Ultra 250 MHz	0.0193 secs	UCI ↔ Mozambique	0.6687 secs

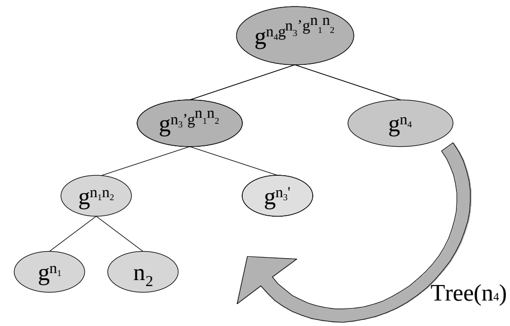
56/67

Goal

- To design a key agreement scheme which has
  - small number of round
  - Small number of message
  - But, may compute little bit more

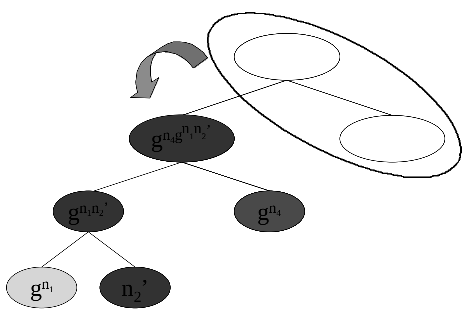
57/67

Join



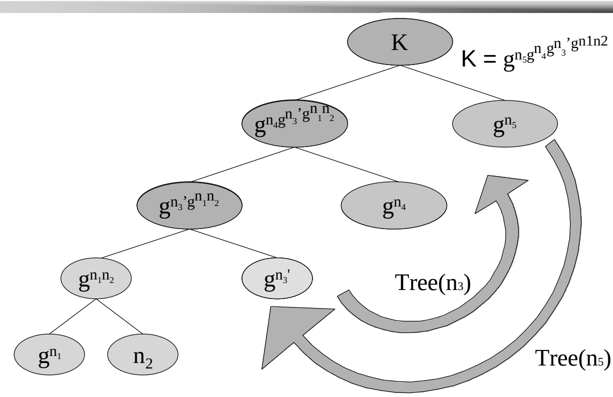
58/67

Leave or Partition



59/67

Merge



60/67

Discussion

- Security
  - Same as TGDH, since STR key tree is a special case of TGDH key tree
- Efficiency
  - Average number of mod exp: 2 n
  - Maximum number of round: 2
  - Maximum number of message: 3
- Robustness is easily provided due to self-stabilization property

61/67

Outline

- Definitions and notions
- Related work
- Background
- Protocols
  - Cliques
  - TGDH
  - STR
  - BD

62/67

BD

- Computation efficient
  - Constant 2 modular exponentiation
  - Constant 2 communication round
  - Each round requires n broadcasts
- No join, leave, merge, and partition protocol
  - Whenever new membership happens, need to build new group key
- Fault-tolerant
  - Whenever cascaded event happens, start from scratch
- Secure
  - Contributory
  - Backward and forward secrecy
  - Provable security
  - Key independence

63/67

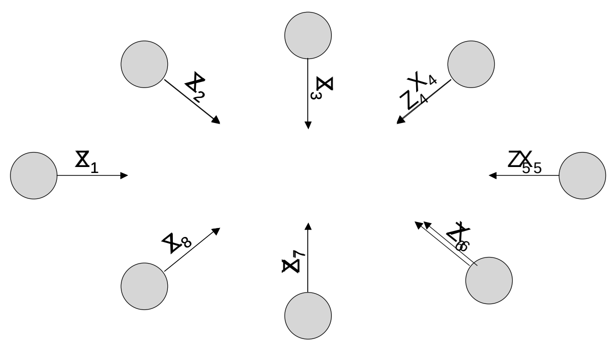
Protocol

1. Each U_i selects random integer r_i and computes and broadcasts $z_i = g^{r_i} \bmod p$
2. Each U_i computes and broadcasts
$$X_i = (z_{i+1}/z_{i-1})^{r_i} \bmod p$$
3. Each U_i computes the conference key
$$K_i = (z_{i-1})^{n\, r_i}\, X_i^{n-1}\, X_{i+1}^{n-2} \ldots X_{i-2} \bmod p$$

$$\begin{aligned} Ki &= (z_{i-1})^{n\, r_i}\, X_i^{n-1}\, X_{i+1}^{n-2} \ldots X_{i-2} \\ &= (z_{i-1})^{n\, r_i}\, (z_{i+1}/z_{i-1})^{r_i\, n-1}\, (z_{i+2}/z_i)^{r_{i+1}\, n-2} \ldots (z_{i-1}/z_{i-3})^{r_{i-2}} \\ &= (g^{r_{i-1}})^{n\, r_i}\, (g^{(r_{i+1} - r_{i-1})r_i})^{n-1}\, (g^{(r_{i+2} - r_i)r_{i+1}})^{n-2} \ldots (g^{(r_{i-1} - r_{i-3})r_{i-2}}) \\ &= g^{r_1 r_2 + r_2 r_3 + r_3 r_4 + \ldots + r_n r_1} \end{aligned}$$

64/67

BD Topology



Discussion

- Security
  - Group key is indistinguishable from random number
- Efficiency
  - Constant 2 modular exponentiation
  - Constant 2 communication round
  - Each round requires n broadcasts
- Robustness is easily provided since we start from scratch

Comparison

		Comm				Comp	Robust
		Round	Msg	Uni	Broad	Exp	
CLQ	Join	4	n+3	n+1	2	n+3	Hard
	Leave, Partition	1	1	0	1	n-1	
	Merge	k+3	n+2k+1	n+2k-1	2	n+2k+1	
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	2	3	1	3	7	Easy
	Leave, Partition	1	1	0	1	3n+6	
	Merge	2	3	0	3	4k+4	
BD		2	2n	0	2n	3	Easy