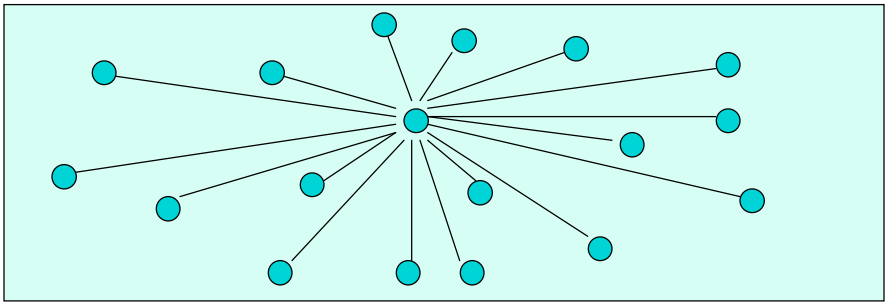
NC STATE UNIVERSITY Computer Science

CSC 774 Advanced Network Security

Topic 5 Group Key Management

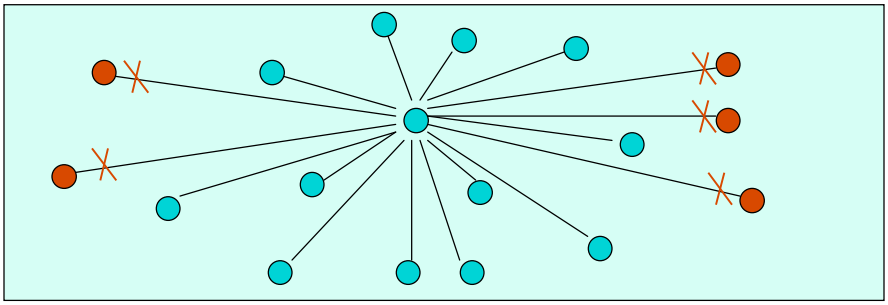
Group Communication

- A group consists of multiple members
- Messages sent by one sender are received by all the other group members
- Example application: Pay per view



Secure Group Communication

- Messages sent by a valid group member can only be understood by the other valid members
  - Others may receive the messages, but are unable to understand them
  - Typical approach: Encrypt the group messages with a key only known to the valid group members



Group Key Management

- Group key management
  - Ensure only valid group members have access to the group key
  - The REAL problem for secure group communication

Desired Properties of Group Key Management

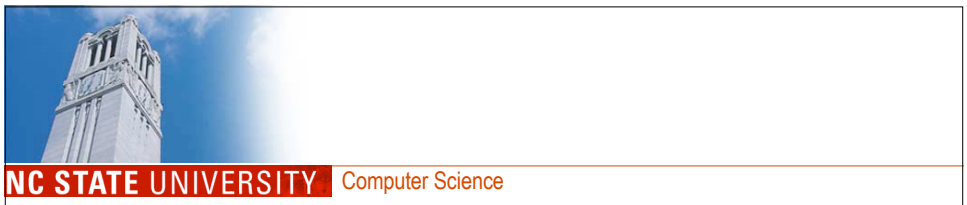
- Group key secrecy
  - It is at least computationally infeasible for an adversary to discover any group key
- Forward secrecy
  - A passive adversary who knows a contiguous subset of old group keys cannot discover subsequent group keys
  - Do not confuse with PFS
- Backward secrecy
  - A passive adversary who knows a contiguous subset of group keys cannot discover preceding group keys
- Group key independence
  - The combination of forward and backward secrecy.

Statefule v.s. Stateless

- Stateful
  - Decryption of new key depends on previous keys
  - Group member should keep track of all rekeying messages
  - Members should be online
- Stateless
  - Decryption of new key depends on establishment key set that is assigned when member join
  - Group members don't need to keep track of rekeying messages
  - Members can be offline

Types of Group Key Management

- Group key agreement
  - Group keys are determined collectively by all group members
  - Usually extended from D-H key exchange
- Group key distribution
  - Group keys are determined and distributed by a group key manager



NC STATE UNIVERSITY Computer Science

CSC 774 Advanced Network Security

Topic 5.1 Group Diffie-Hellman Protocols

Outline

- Review of the basic two-party D-H key exchange
- Generic n-party D-H key agreement
- Three specific protocols
  - GDH.1
  - GDH.2
  - GDH.3

Two-Party Diffie-Hellman Key Exchange

Alice	Bob
Pick secret S_a randomly	Pick secret S_b randomly
Compute $T_A = g^{S_a} \bmod p$	Compute $T_B = g^{S_b} \bmod p$
Send T_A to Bob	Send T_B to Bob
Compute $T_B^{S_a} \bmod p$	Compute $T_A^{S_b} \bmod p$

Shared key is reached at both parties: $g^{S_a S_b} \bmod p$

Notations

- n: number of participants in the protocol
  - α: exponentiation base
  - q: order of the algebraic group
  - M_i: i-th group member, i is the index
  - N_i: random exponent generated by group member M_i
  - S: subsets of $\{N_1, \ldots, N_n\}$
  - $\Pi(S)$: product of all elements in subset S
  - K_n: group key shared among n members

Generic n-Party D-H Key Agreement

- Setup
  - All n participants agree on a cyclic group G, of order q and the base α
  - Each member M_i chooses a random value $N_i \in G$

Generic n-Party D-H Key Agreement (Cont'd)

- Generic Protocol:
  - Distributively revealing and computing a subset of $\{\alpha^{\Pi(S)} | S \subset \{N_1, \ldots, N_n\}\}$
  - From these subsets, member M_i computes

$$\alpha^{N_1 \ldots N_{i-1} N_{i+1} \ldots N_n} \bmod q$$

  - Finally, M_i computes the shared key

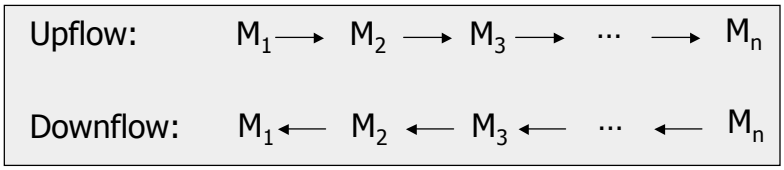
$$K = \alpha^{N_1 \cdots N_n} \bmod q$$

Generic n-Party D-H Key Agreement (Cont'd)

- Security
  - The generic n-party D-H protocol is secure if the 2-party D-H protocol is security
  - Proof: by induction on n
- Remaining problem
  - Consider $\{\alpha^{\Pi(S)} | S \subset \{N_1, \ldots, N_n\}\}$
  - What (S) to distribute, and how?

GDH.1

- Consists of an upflow stage and a downflow stage



GDH.1 (Cont'd)

- Upflow
  - M_i receives the set $\{\alpha^{N_1}, \alpha^{N_1N_2}, \ldots, \alpha^{N_1 \cdots N_{i-1}}\}$ and forwards to M_{i+1} $\{\alpha^{N_1}, \alpha^{N_1N_2}, \ldots, \alpha^{N_1 \cdots N_i}\}$, $i \in [1, n-1]$
- Example
  - M_4 receives the set $\{\alpha^{N_1}, \alpha^{N_1N_2}, \alpha^{N_1N_2N_3}\}$
  - and forwards to M_5 $\{\alpha^{N_1}, \alpha^{N_1N_2}, \alpha^{N_1N_2N_3}, \alpha^{N_1N_2N_3N_4}\}$

GDH.1 (Cont'd)

- Downflow
  - M_i uses the last intermediate value to compute K_n ($1<i<=n$)
  - M_i then raises all remaining values to the power of N_i and forwards the resulting set to M_{i-1}
- Example
  - M_4 receives the set
    $\{\alpha^{N_5}, \alpha^{N_1N_5}, \alpha^{N_1N_2N_5}, \alpha^{N_1N_2N_3N_5}\}$
  - and forwards to M_3
    $\{\alpha^{N_5N_4}, \alpha^{N_1N_5N_4}, \alpha^{N_1N_2N_5N_4}\}$

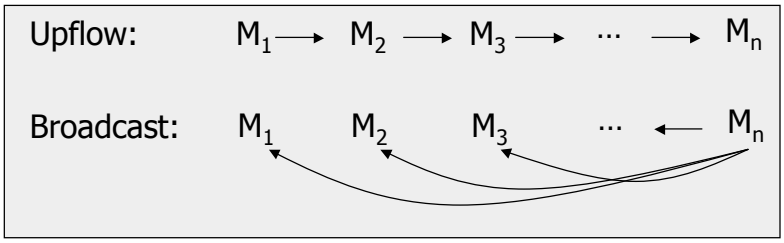
GDH.1 (Cont'd)

- How many rounds?
  - _____
- How many messages in GDH.1?
  - _____
- How many exponentiations per M_i?
  - _____

Upflow:	M_1	→	M_2	→	M_3	→	···	→ M_n
Downflow:	M_1	←	M_2	←	M_3	←	···	← M_n

GDH.2

- Consists of an upflow stage and a broadcast stage
  - Use broadcast to reduce communication overhead



GDH.2 (Cont'd)

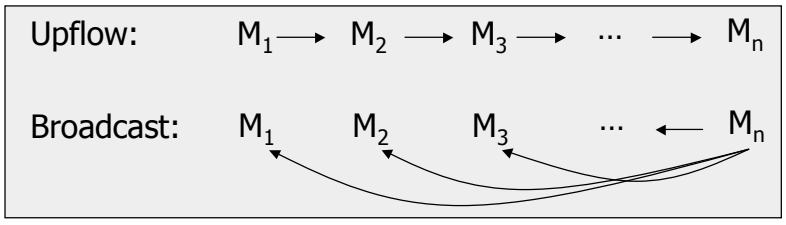
- Upflow
  - M_i composes *i* intermediate values and one cardinal value and forwards the resulting set to M_{i+1} ($i < n$)
- Example:
  - M_4 receives the set
    $\{\alpha^{N_1N_2N_3}, \alpha^{N_1N_2}, \alpha^{N_1N_3}, \alpha^{N_2N_3}\}$
  - and forwards to M_5
    $\{\alpha^{N_1N_2N_3N_4}, \alpha^{N_1N_2N_3}, \alpha^{N_1N_2N_4}, \alpha^{N_1N_3N_4}, \alpha^{N_2N_3N_4}\}$

GDH.2 (Cont'd)

- Downflow
  - M_n raises every intermediate value to the power of N_n broadcasts the resulting values to all group members, in another word
  - M_n broadcasts the set $\{\alpha^{N_1 \cdots N_{i-1} N_{i+1} \cdots N_n}\}$ to M_i ($i < n$)
- Example
  - M_4 receives the set $\{\alpha^{N_1 N_2 N_3 N_5}\}$ from M_5 (Assume n=5)

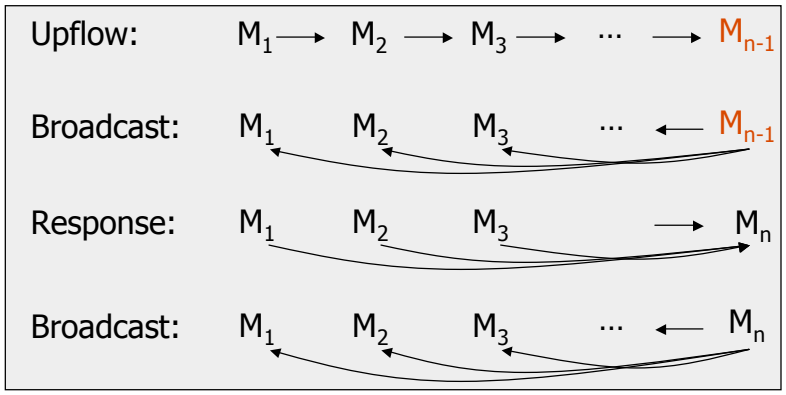
GDH.2 (Cont'd)

- How many rounds?
  - _____
- How many messages in GDH.2?
  - _____
- How many exponentiations per M_i?
  - _____



GDH.3

- Consists of an upflow stage, a broadcast stage, a response stage, and final broadcast stage
  - Reduce the number of exponentiations per group member.



GDH.3 (Cont'd)

- Upflow
  - M_i ($i \in [1, n\text{-}2]$) receives $\alpha^{N_1 \cdots N_{i-1}}$, and
  - forwards to M_{i+1} $\alpha^{N_1 \cdots N_i}$,
- Broadcast
  - M_{n-1} broadcasts $\alpha^{N_1 \cdots N_{n-1}}$ to M_i ($i \neq n\text{-}1$)

GDH.3 (Cont'd)

- Response
  - M_i ($i < n$) factors out its own component and forwards $\alpha^{N_1 \cdots N_{i-1} N_{i+1} \cdots N_{n-1}}$ to M_n
- Broadcast
  - M_n raises every input to the power of N_n and broadcasts the resulting set $\{\alpha^{N_1 \cdots N_{i-1} N_{i+1} \cdots N_n}\}$ to M_i ($i < n$)

GDH.3 (Cont'd)

- How many rounds?
  - _____
- How many messages in GDH.2?
  - _____
- How many exponentiations per M_i?
  - _____

Comparison

	GDH.1	GDH.2	GDH.3
Rounds	2(n-1)	n	n+1
Messages	2(n-1)	n	2n-1
Total message size	n(n-1)	(n-1)(n/2+2)-1	3(n-1)
Exp ops per M_i	i+1, n	i+1, n	4, 2, n
Total exp ops	(n+3)n/2-1	(n+3)n/2-1	5n-6

Alteration of Group Membership

- GDH.1 does not support efficient member addition/deletion.
- GDH.2 & GDH.3
  - Member addition
    - Consider the new member as the new M_{n+1}
  - Member deletion
    - M_n regenerates its secret N_n and re-executes the protocol from the second stage.