Diffie-Hellman Key Distribution
Extended to Group Communication

Qingfeng He
Department of Computer Science
North Carolina State University
April 02, 2003

Agenda
- Introduction
- Problem Statement
- Generic n-Party D-H Key Distribution
- Group Key Distribution Protocols
- Related Work
- Protocol Comparison
- Limitations and Future Work
Introduction

2-Party Diffie-Hellman Key Exchange

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick secret $S_a$ randomly</td>
<td>Pick secret $S_b$ randomly</td>
</tr>
<tr>
<td>Compute $T_A = g^{S_a} \mod p$</td>
<td>Compute $T_B = g^{S_b} \mod p$</td>
</tr>
<tr>
<td>Send $T_A$ to Bob</td>
<td>Send $T_B$ to Bob</td>
</tr>
<tr>
<td>Compute $T_B^{S_a} \mod p$</td>
<td>Compute $T_A^{S_b} \mod p$</td>
</tr>
</tbody>
</table>

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

Problem Statement

- Extend 2-Party D-H to n-Party
- Motivation
- Problem of Previous Approaches
  - Protocols are too theoretical
  - Protocol security unproven
Generic n-Party D-H Key Distribution (1)

- Notation
  - $n$: number of participants in the protocol
  - $\alpha$: 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 $\{N_1, \ldots, N_n\}$
  - $\mathcal{II}(S)$: product of all elements in subset $S$
  - $K_n$: group key shared among $n$ members

Generic n-Party D-H Key Distribution (2)

- Priori
  - All $n$ participants agree on a cyclic group $G$, of order $q$ and the base $\alpha$
  - Each member $M_i$ chooses a random value $N_i \in G$
Generic n-Party D-H Key Distribution (3)

- Generic Protocol:
  - Distributively revealing and computing a subset of \( \{a^{|S|}\mid S \subseteq \{N_1, \ldots, N_n\}\} \)
  - From these subsets, member \( M_i \) computes \( a^{N_1 \cdots N_{i-1}N_{i+1} \cdots N_n} \mod q \)
  - Finally, \( M_i \) computes the shared key \( K = a^{N_1 \cdots N_n} \mod q \)

Generic n-Party D-H Key Distribution (4)

- Protocol Security Assumption
  - 2-party D-H key distribution is secure
- Proof: by induction on \( n \)
Group Key Distribution Protocol (1)

- **Group Key Distribution: GDH.1**
  - **Upflow:** $M_i$ receives the set $\{d_{N1}, d_{N1N2}, \ldots, d_{N1N2\ldots Ni-1}\}$ and forwards to $M_{i+1}$ $\{d_{N1}, d_{N1N2}, \ldots, d_{N1N2\ldots Ni}\}$, $i \in [1, n-1]$
  - **Example:** $M_4$ receives the set $\{d_{N1}, d_{N1N2}, d_{N1N2N3}\}$ and forwards to $M_5$ $\{d_{N1}, d_{N1N2}, d_{N1N2N3}, d_{N1N2N3N4}\}$

Group Key Distribution Protocol (2)

- **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 $\{d_{N5}, d_{N1N5}, d_{N1N2N5}, d_{N1N2N3N5}\}$ and forwards to $M_3$ $\{d_{N5N4}, d_{N1N5N4}, d_{N1N2N5N4}\}$
Group Key Distribution Protocol (3)

- Group Key Distribution: GDH.2
  - **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 \{ $\alpha^{N_1N_2}$, $\alpha^{N_1N_3}$, $\alpha^{N_2N_3}$ \} and forwards to $M_5$ \{ $\alpha^{N_1N_2N_3N_4}$, $\alpha^{N_1N_2N_3}$, $\alpha^{N_1N_2N_4}$, $\alpha^{N_1N_3N_4}$, $\alpha^{N_2N_3N_4}$ \}

Group Key Distribution Protocol (4)

- GDH.2 (Cont’d)
  - **Downflow**:
    - $M_n$ raises every intermediate value to the power of $N_n$ and broadcasts the resulting values to all group members, in another word
    - $M_n$ broadcasts the set \{ $\alpha^{N_1...N_{i-1}N_{i+1}...N_n}$ \} to $M_i$ ($i < n$)
  - **Example**: $M_4$ receives the set \{ $\alpha^{N_1N_2N_3N_5}$ \} from $M_5$ (Assume $n=5$)
Group Key Distribution Protocol (5)

- Group Key Distribution: GDH.3
  - Upflow: \( M_i \) receives the set \( \{a^{N_1}, a^{N_1N_2}, \ldots, a^{N_1\cdots N_{i-1}}\} \) and forwards to \( M_{i+1} \) \( \{a^{N_1}, a^{N_1N_2}, \ldots, a^{N_1\cdots N_{i-1}}\}, i \in [1, n-2] \)
  - Broadcast: \( M_{n-1} \) broadcasts the set \( \{a^{N_1\cdots N_{n-1}}\} \) to \( M_i \) \( (i \neq n-1) \)

Group Key Distribution Protocol (6)

- GDH.3 (Cont’d)
  - Response: \( M_i \) \((i < n)\) factors out its own component and forwards the set \( \{a^{N_{i+1}\cdots N_{n-1}}\} \) to \( M_n \)
  - Broadcast: \( M_n \) raises every input to the power of \( N_n \) and broadcasts the resulting set \( \{a^{N_{i+1}\cdots N_{n-1}\cdots N_i\cdots N_{n}}\} \) to \( M_i \) \((i < n)\)
Group Key Distribution Protocol (7)

- Properties Comparison

<table>
<thead>
<tr>
<th></th>
<th>GDH.1</th>
<th>GDH.2</th>
<th>GDH.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rounds</td>
<td>2(n-1)</td>
<td>n</td>
<td>n+1</td>
</tr>
<tr>
<td>Messages</td>
<td>2(n-1)</td>
<td>n</td>
<td>2n-1</td>
</tr>
<tr>
<td>Combined message size</td>
<td>n(n-1)</td>
<td>(n-1)(n/2+2)-1</td>
<td>3(n-1)</td>
</tr>
<tr>
<td>Exponentiations per $M_i$</td>
<td>i+1, n</td>
<td>i+1, n</td>
<td>4, 2, n</td>
</tr>
<tr>
<td>Total exponentiations</td>
<td>(n+3)n/2-1</td>
<td>(n+3)n/2-1</td>
<td>5n-6</td>
</tr>
</tbody>
</table>

Group Key Distribution Protocol (8)

- Alteration of Group Membership
  - Member addition
  - Member deletion
Group Key Distribution Protocol (9)

Protocol Advantages
- No synchronization
- Small number of exponentiations
- Minimal total number of messages (GDH.2)
- Minimal number of rounds for asynchronous operation (GDH.2)
- Minimal number of messages sent/received by each participant (GDH.2)
- Security equivalent to 2-party D-H
- Implementation simplicity

Related Work

Ingemarsson et al. (ING)
- Requires a synchronous startup
- All participants must be arranged in a logical ring

Burmester/Desmedt (BD)
- $K_n = g^{N_1N_2+N_2N_3+\ldots+N_NN_1}$
- Cheap exponentiation operations because of relatively small exponents involved in almost all operations
- Time (number of rounds: 2): simultaneous broadcasts
- BD* (without simultaneous broadcast): 2n-1 rounds
Protocol Comparison (1)

Comparison with GDH protocols

- Number of rounds
  - GDH.3: $n-1$ simultaneous unicasts to $M_n$ (less load compared with BD)
  - BD: $n$ simultaneous broadcasts
- Communication bandwidth overhead
  - GDH.2: $n$ messages
  - BD*: least total information changed

Protocol Comparison (2)

Comparison with GDH protocols (Cont’d)

- Protocol efficiency (number of messages received and sent by each participant)
  - GDH.2: least overhead with respect to the communication infrastructure
- Protocol symmetry
  - BD/BD* and ING: offer symmetric operations
  - GDH protocols: asymmetric
## Protocol Comparison (3)

<table>
<thead>
<tr>
<th></th>
<th>ING</th>
<th>BD</th>
<th>GDH.2</th>
<th>GDH.3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rounds</strong></td>
<td>n-1</td>
<td>2</td>
<td>n</td>
<td>n+1</td>
</tr>
<tr>
<td><strong>Messages</strong></td>
<td>n(n-1)</td>
<td>2n</td>
<td>n</td>
<td>2n-1</td>
</tr>
<tr>
<td><strong>Combined message size</strong></td>
<td>n(n-1)</td>
<td>2n</td>
<td>(n-1)(n/2+2)-1</td>
<td>3(n-1)</td>
</tr>
<tr>
<td><strong>Exponentiations per ( M_i )</strong></td>
<td>n</td>
<td>n+1</td>
<td>i+1, n</td>
<td>4, 2, n</td>
</tr>
<tr>
<td><strong>Total exponentiations</strong></td>
<td>n^2</td>
<td>n(n+1)</td>
<td>(n+3)n/2-1</td>
<td>5n-6</td>
</tr>
<tr>
<td><strong>Divisions per ( M_i )</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Limitations and Future Work

- Do not provide authentication of the participants
- Do not handle periodic re-keying
- Formal proof to support optimality/minimality claims
Thank you

- Questions?