Provably Authenticated Group Diffie-Hellman Key Exchange:

The Dynamic Case

Olivier Chevassut
(Université Catholique de Louvain - Lawrence Berkeley National Lab)

Emmanuel Bresson and David Pointcheval
(École normale supérieure)

Outline

- Motivation
- The Problem
- Related Work
- Security Model
- Security Definitions
- A Secure Authenticated Group Diffie-Hellman Protocol
- Security Theorem
- Conclusion
Motivation

- An increasing number of distributed applications need to communicate within groups, e.g.
  - collaboration and videoconferencing tools
  - replicated servers
  - stock market and air traffic control
  - distributed computations (Grids)
- An increasing number of applications have security requirements
  - privacy of data
  - protection from hackers (public network)
  - protection from viruses and trojan horses
- Group communication must address security needs

The Problem

- Group Diffie-Hellman Characteristics
  - group relative small (up to 100 members)
  - no centralized server
  - members have similar computing power
  - membership is dynamic (members join and leave the group at any time)

- Goals for Group Key Exchange
  - Authenticated Key Exchange (AKE)
    - implicit authentication: only the intended partners can compute $sk$
    - semantic security: a session key is indistinguishable from a random string
  - Mutual Authentication (MA)
Prior Work: The Static Case

• “Provably Authenticated Group DH Key Exchange”, ACM CCS’01
  — static membership (all the members join the group at once)
  — model of computation in the Bellare-Rogaway style
    • players are modeled via oracles
    • adversary controls all interactions among the players
    • adversary’s capabilities are modeled by queries to the oracles
    • adversary plays a game against the players
  — an authenticated group Diffie-Hellman key exchange protocol

Model of Communication

• A set of $n$ players
  — each player is represented by an oracle
  — each player holds a long-lived key (LL)
• A multicast group consisting of a set of players

Multicast Group with sk
Modeling the Adversary

- Adversary’s capabilities modeled through queries
  - setup: initialize the multicast group
  - remove: remove players from multicast group
  - join: add players to the multicast group

Freshness Related Queries

$sk$ is Fresh if it is known by the players but not the adversary

(\texttt{LL})

\texttt{corrupt}

(\texttt{sk})

\texttt{reveal}
A Secure Authenticated Group Diffie-Hellman Protocol

- The session key is
  \[ sk = H(g^{x_1 x_2 \ldots x_n}) \]

- Ring-Based with flows

- Defined by three algorithms
  - SETUP
  - REMOVE
  - JOIN

- Many details abstracted out
The SETUP Algorithm

- Up-flow: $U_i$ raises received values to the power of $x_i$ and forwards to $U_{i+1}$
- Down-flow: $U_n$ processes the last up-flow and broadcasts

$sk = H(g^{x_1 x_2 x_3})$

The REMOVE Algorithm

- Down-flow of the SETUP algorithm

$sk = H(g^{x_1 x_2 x_3})$
The JOIN Algorithm

- SETUP initiated by player with highest index in group ($U_{gc}$)

\[ sk = H(g^{x_1x_2x_3x_4}) \]

Security Theorem (AKE)

- Random-oracle assumption
- Theorem
\[ \text{Adv}_{\text{ake}}(T,Q,q_s,q_h) \leq 2 \cdot n \cdot \text{Succ}_{\text{ema}}(T') + 2 \cdot Q \cdot (n_s) \cdot s \cdot q_h \cdot \text{Succ}_{\text{goh}}(T') \]
\[ T', T'' \leq T + (Q+q_s) \cdot n \cdot T_{\text{exp}}(k) \]

- Adversary breaks AKE in two ways:
  1. Assume that the adversary forges a signature w.r.t some player's LL-key => it is possible to build a forger
  2. Assume that the adversary is able to guess the bit $b$ involved in the Test-query
     => it is possible to come up with an algo that solves an instance of the Group Diffie-Hellman problem
Conclusion and Future Work

- Summary
  - A security model for the dynamic case
  - A secure protocol
  - A proof of security in the random-oracle model
- Limitations
  - sequential executions only
  - random-oracle assumption
- "Concurrent Executions for Authenticated Dynamic Group DH Key Exchange using Crypto-Devices", Work in Progress
  - concurrent executions
  - standard model
  - weak-corruption and strong-corruption models