Collaborative Security

Gene Tsudik
USC/ISI
4676 Admiralty Way
Marina del Rey, CA 90292
gts@isi.edu
http://www.isi.edu/~gts

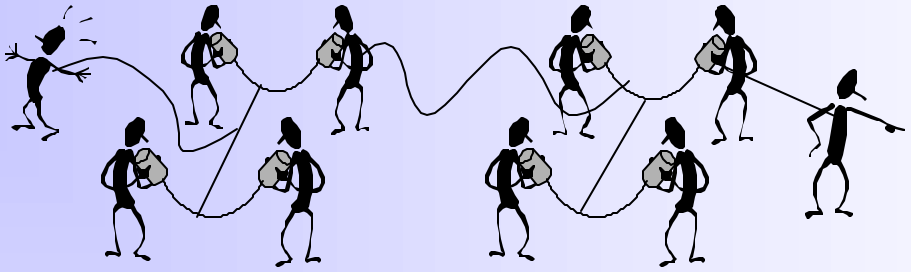
Group Communication

- One-to-many
  - Single-source broadcast: cable/sat. TV, radio, etc.
- Few-to-many
  - Multi-source broadcast (2-tiered groups): televised debates, GPS, time, etc.
- Any-to-any
  - Collaborative applications: conferencing, mailing lists, visualization, instrument control, simulations, replicated servers, etc.

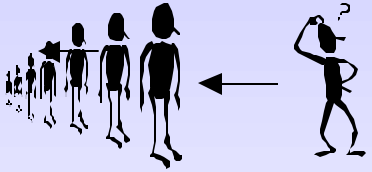
Rich communication semantics, tighter control, more emphasis on **security**

CLIQUES: Security in Dynamic Peer Groups

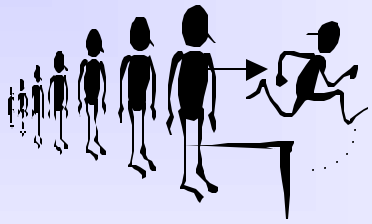
(DARPA/ITO HCN, 07/97-06/00)



Formation



Member add



Member leave

Group merge

Group partition

Background

Targeted environment

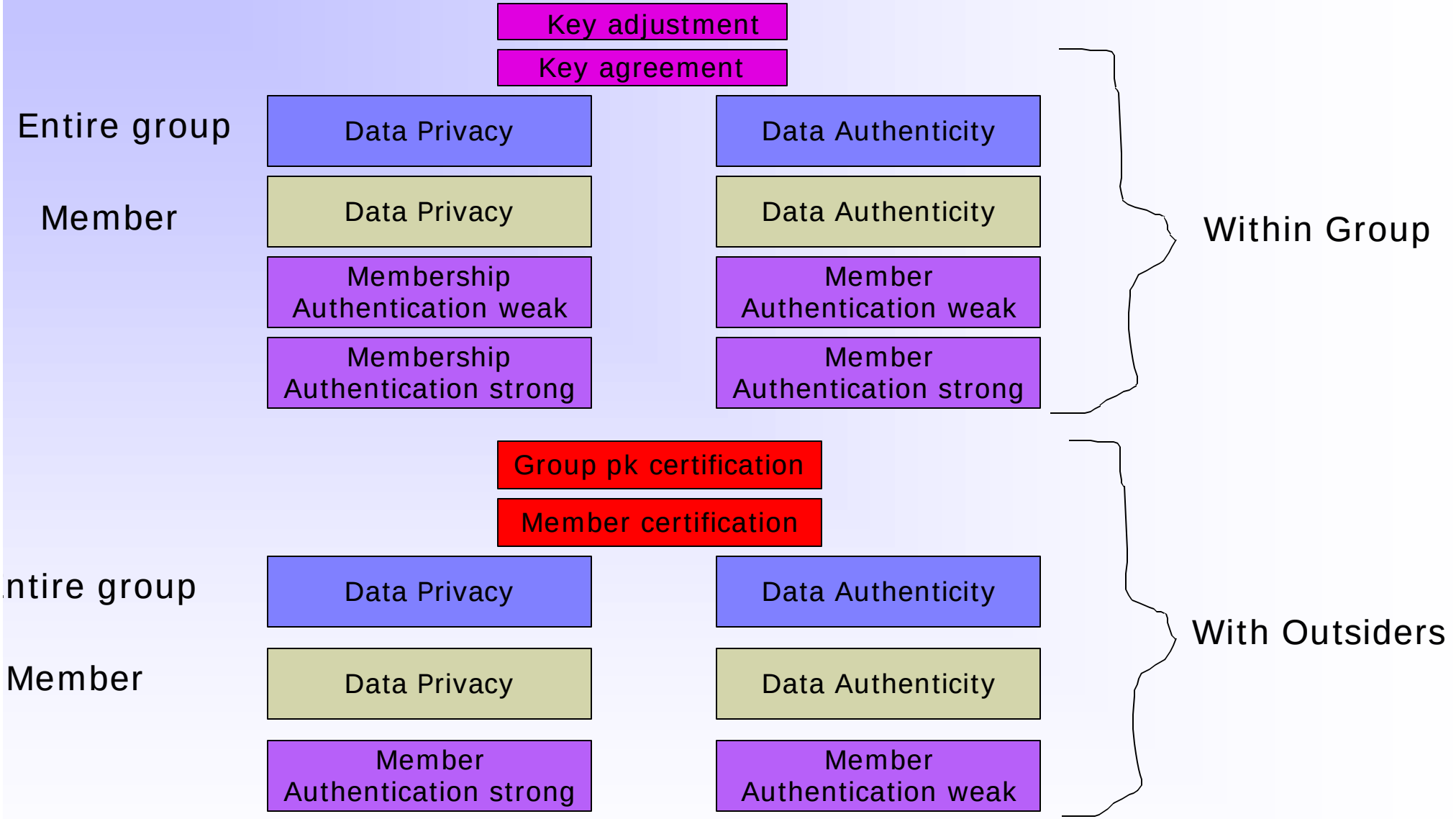
- Relatively small groups
- Dynamic membership
- No hierarchy
- Many-to-Many
- Collaborative applications

Problem:

how to obtain security in peer groups with dynamic membership and decentralized control?

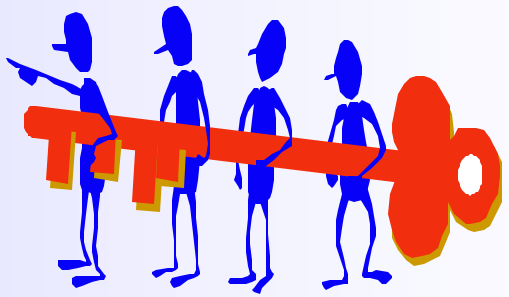
Complexity >> 2- and 3-party security

Security Services



Group Diffie-Hellman

Important features:



- **Formal** proof of security
- Decentralized.
- No ordering, no synchronization
- No topology or network dependencies
- Group controller -- floating or fixed (chore!)
- Everyone contributes to the key.
- Everyone can prove they took part in the generation of the key.
- Two message latency for join of 1 member
- One message latency for leave of N members.
- N+1 message latency for join/merge of N members

Why key agreement?

- Centralized (TTP) schemes untenable
- 2-,3-party extensions unscalable

Diffie-Hellman Primer (DH78)

p – large prime (≥ 512 bits)
$Z_p^* = \{1,..., p-1\}$
g – base (generator)

Alice

Bob

$a \in_R Z_p^*$

$A = g^a \bmod p$

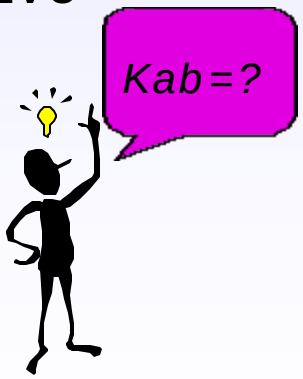
$B = g^b \bmod p$

$b \in_R Z_p^*$

$K_{ba} = A^b \bmod p$

$K_{ab} = B^a \bmod p$

Eve



DH Primer (contd)

p – large prime, g – generator in Z_p

Discrete Log Problem :

– – – – – – – – – – – – – –

Given : $A = g^a \bmod p$

FIND : a

Diffie – Hellman Problem :

– – – – – – – – – – – – – – – –

Given : $A = g^a \bmod p$ *and* $B = g^b \bmod p$

FIND : $g^{ab} \bmod p$

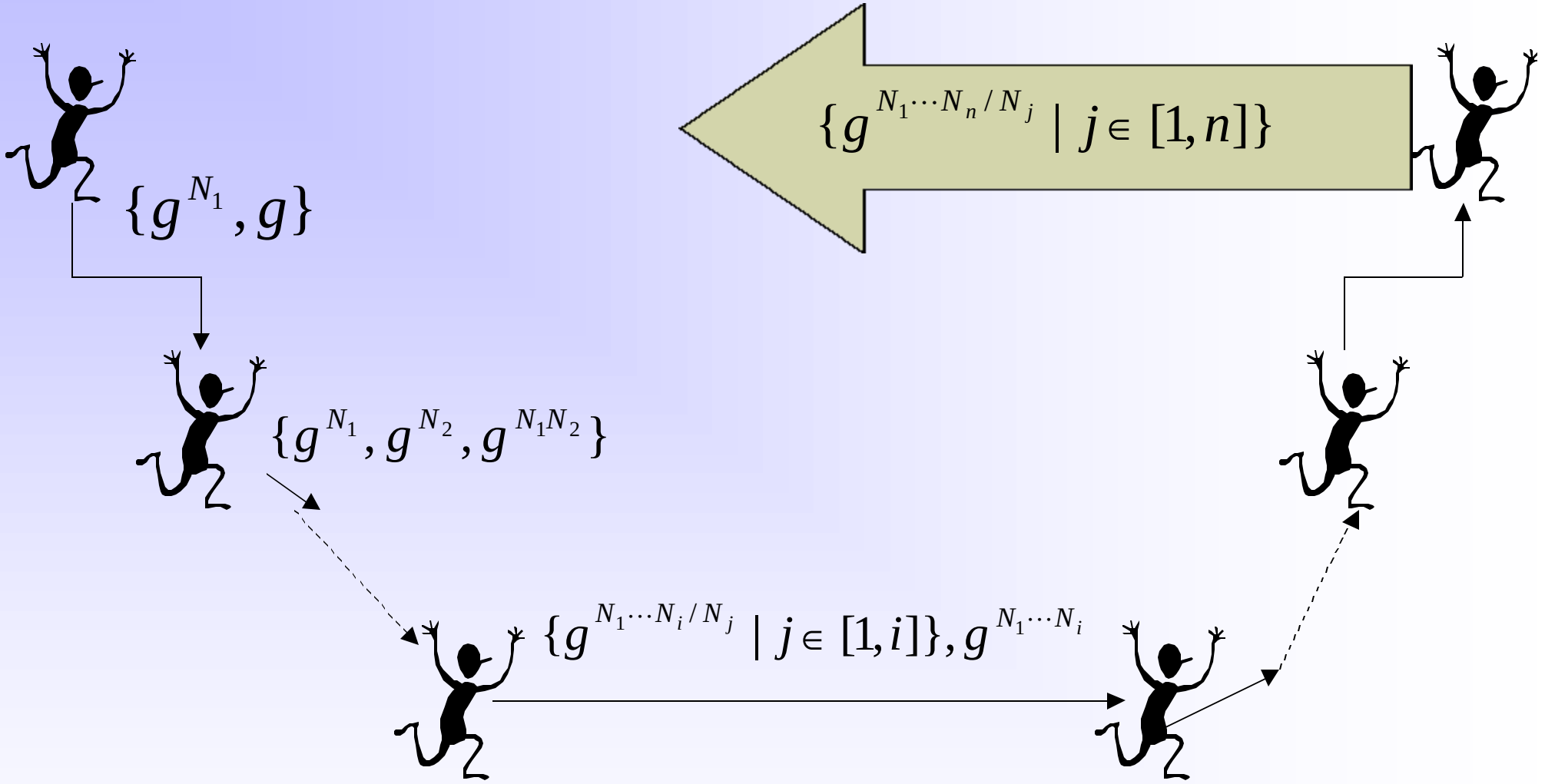
Decision DH Problem :

– – – – – – – – – – – – – –

Given : $y_a = a^{x_a} \bmod p$, $y_b = a^{x_b} \bmod p$

Distinguish : $K_{ab} = a^{x_a x_b} \bmod p$
from a random number!

GDH Key Agreement (contd)

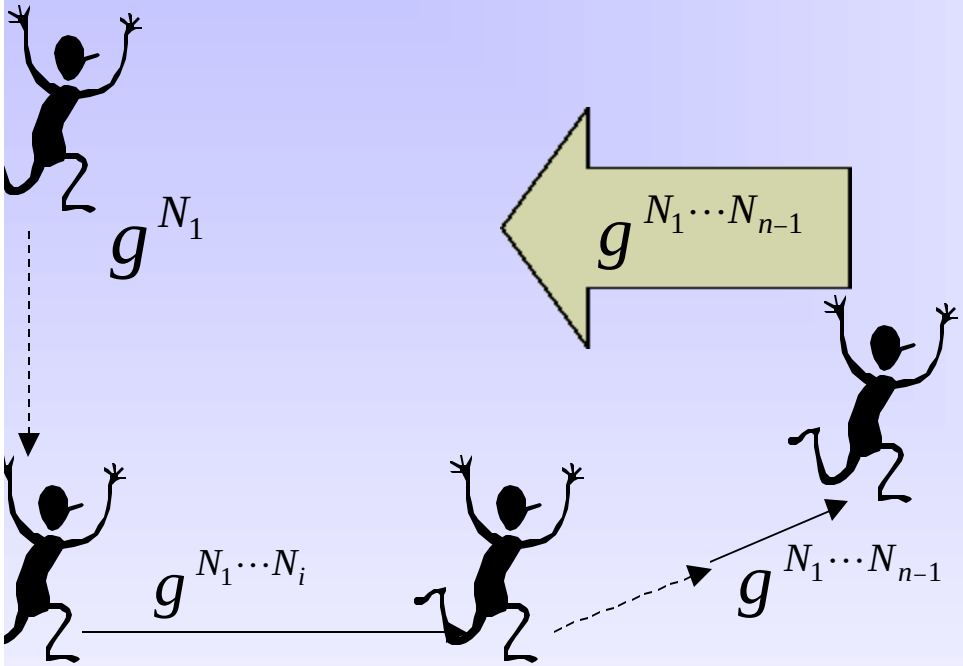


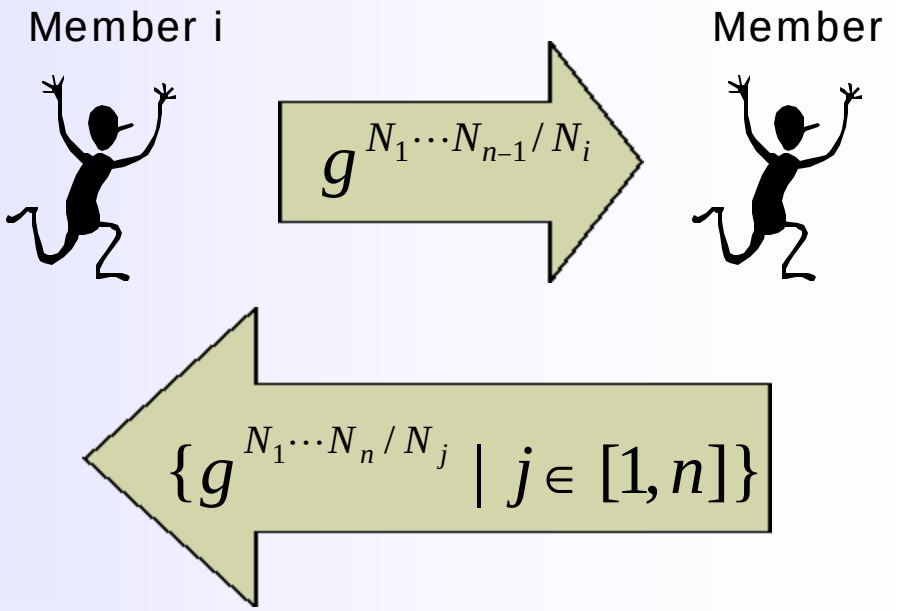
$_n = g^{N_1 \cdots N_n} \bmod p$

olynomially indistinguishable
om random number!

Key adjustments/refresh protocols
easily derived and shown secure

Another GDH Key Agreement

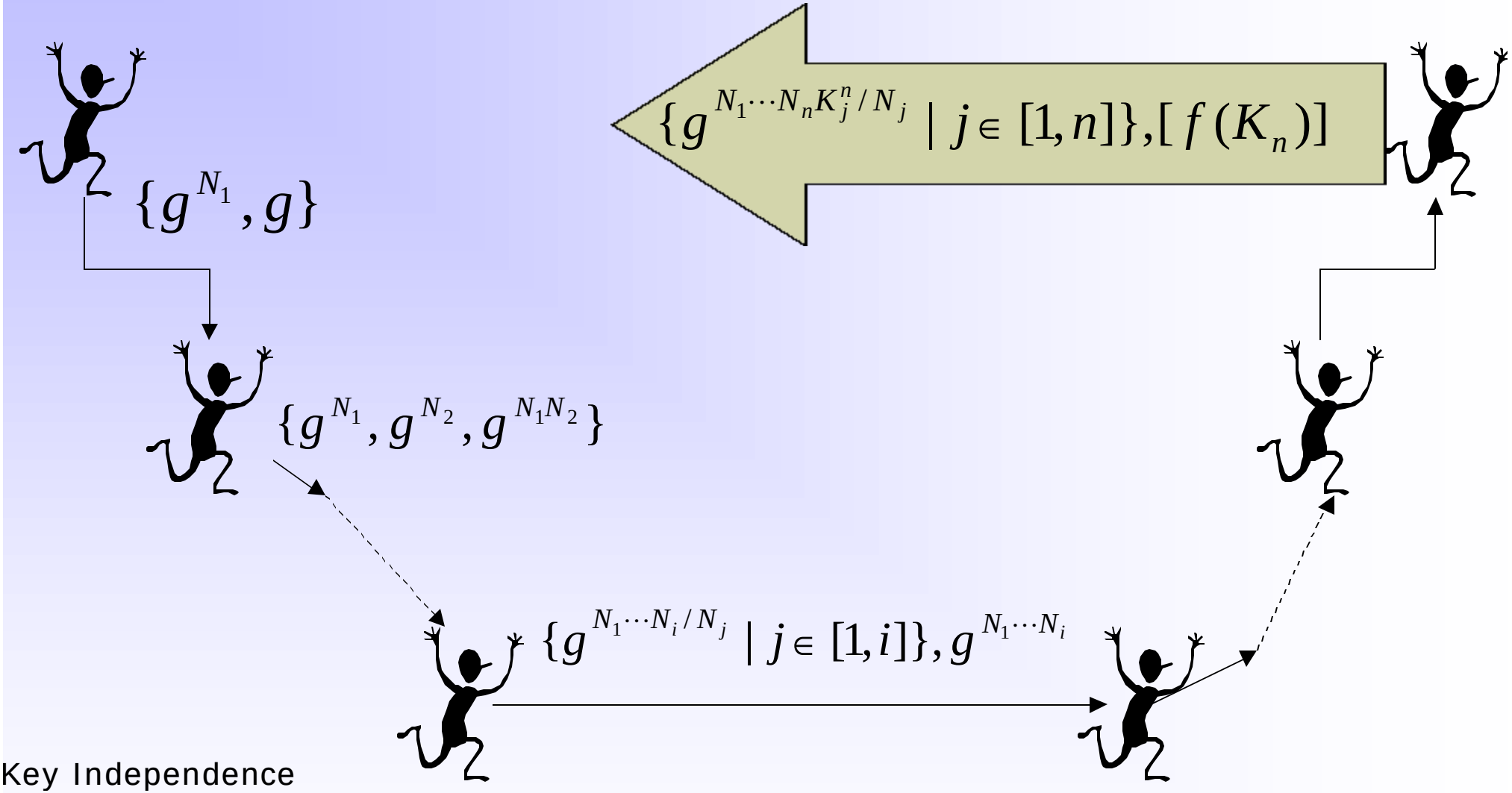




$$K_n = g^{N_1 \cdots N_n} \bmod p$$

- 2 exponentiations per member
- Impractical!

Authenticated GDH Key Agreement



Key Independence
Perfect Forward Secrecy
KKA Resistance
Key Confirmation
Key Authentication

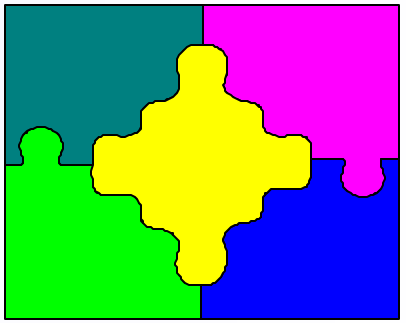
Stronger version:
- Membership Integrity
- Partial entity authentication

Security Services Provided

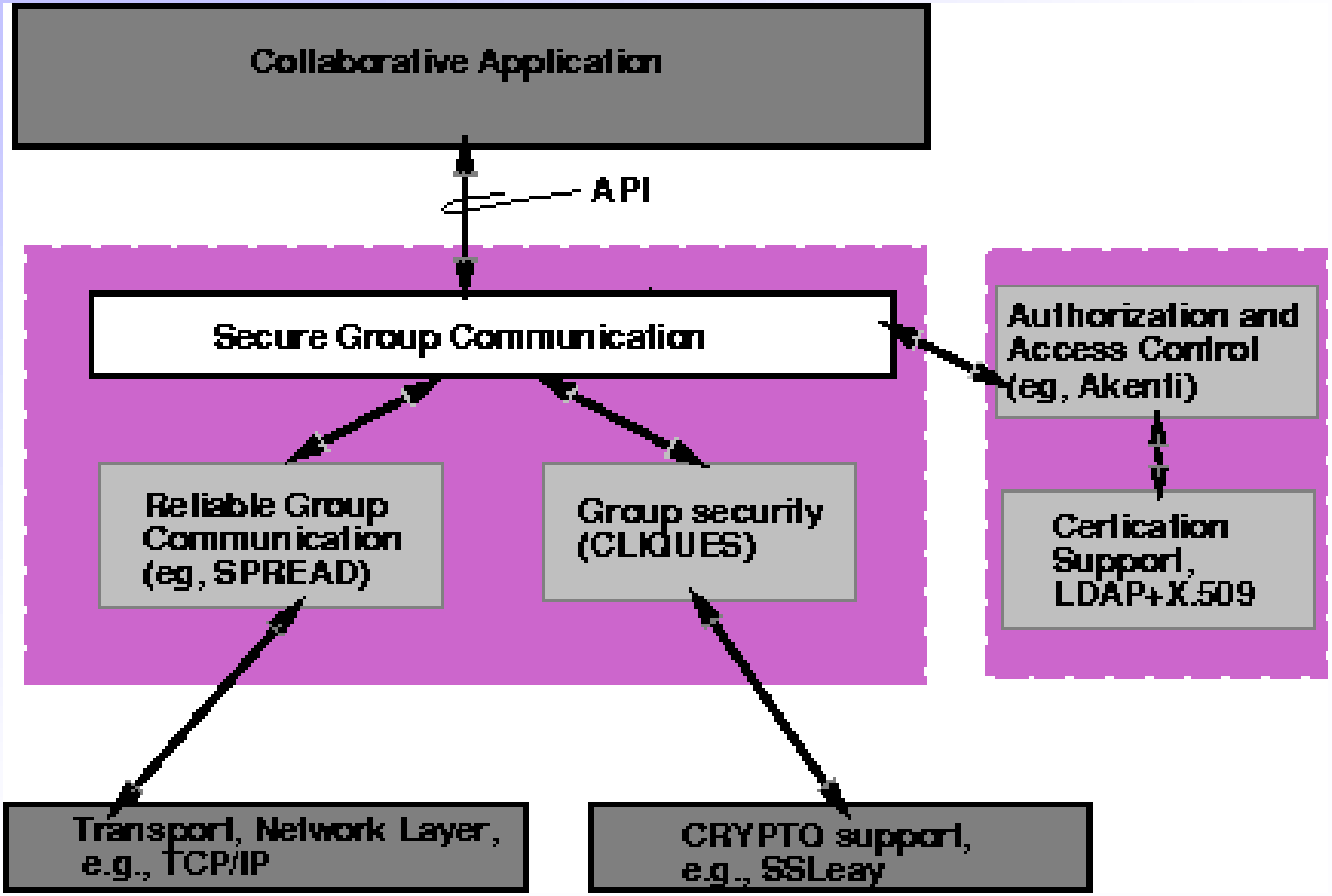
- Decentralized authenticated group key agreement with provable security based on group Diffie-Helman: each member contributes equally to group key
- Membership changes: single member, many members and sub-groups
- Membership authentication and non-repudiation: based on knowledge of key-share
- Authenticated join/leave: requires long-term DH credentials

Other pieces of the puzzle

- Certification infrastructure
- Reliable group communication subsystem
- Membership Authorization / Access control



Proposed Architecture



STATUS

Initial Key Agreement

Auxiliary Key Agreement (membership changes)

Authenticated Key Agreement

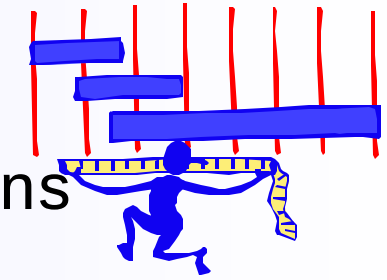
JAVA implementation (rel. 0.0)

C implementation (rel. 0.1) coupled with JHU's SPREAD

CLQ_API: completed (rel. 1.0) mid-Feb

Testing and integrating with SPREAD

Current performance results: O(n) exponentiations



Integration with TOTEM on-going (LBNL)

Integration with AKENTI: near future

Publications

M. Steiner, G. Tsudik and M. Waidner
Diffie-Hellman Key Distribution Extended to Groups
1996 ACM Conference on Computer and Communications Security (CCCS'96)

M. Steiner, G. Tsudik and M. Waidner
CLIQUES: A New Approach to Group Key Agreement
1998 IEEE International Conference on Distributed Computing Systems (ICDCS'98)

G. Ateniese, M. Steiner and G. Tsudik
Authenticated Group Key Agreement and Friends
1998 ACM Conference on Computer and Communications Security (CCCS'98)

M. Steiner, G. Tsudik and M. Waidner
Key Agreement in Dynamic Peer Groups
IEEE Transaction on Parallel and Distributed Systems (TPDS), submitted.

G. Ateniese, M. Steiner and G. Tsudik
Authenticated Group Key Agreement and Friends
IEEE Journal on Selected Areas in Communication (JSAC), to appear.

CLQ_API prerequisites

Underlying group communication subsystem must provide reliable synchronized event notification for:

- group joins
- group leaves
- partitions
- node failures or disconnects
- merges (partition heals)

CLQ_API

called by a new group member who received a NEW_MEMBER message from the current controller.

clq_join (ctx, member_name, group_name, input, output);

called by the current controller to hand over group context to a new member (who becomes next controller).

clq_pass_ctx (ctx, member_name, output);

called by every member upon reception of a KEY_UPDATE_MESSAGE from the current group controller

clq_update_ctx (ctx, input);

CLQ_API (contd)

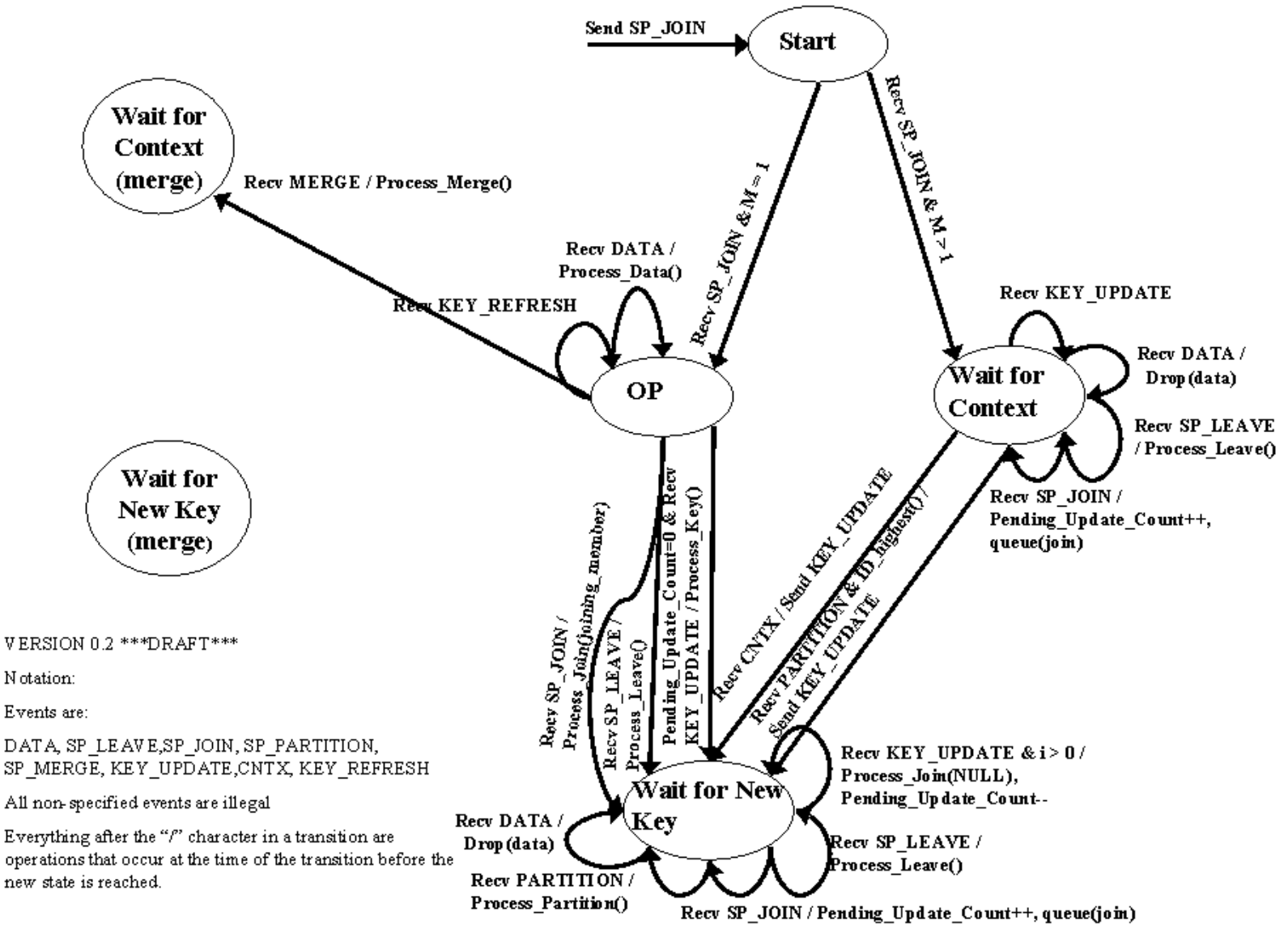
called by every group member after member leaves or partition occurs; removes all valid members in member_list from the group_member_list. Only controller gets output token

clq_leave (ctx, member_list[], output);

called only by controller when group_secret needs to be updated.

clq_refresh_key (ctx, output);

Secure Spread SD



Lessons learned

- Paper protocols <> real protocols
- Incremental formation of groups
- Security, group comm not a simple composition
- Difficulty of handling many merging sub-groups
- Group size limits (100?)
  - other DH-like keys
  - elliptic curve duals
- Provable security matters!

Challenges and directions

- Two-tiered groups (few-to-many)
- Group membership policy (Auth + AC)
  - How to specify?
  - Enforce?
- Group certification: group key, membership, etc.
  - Dynamic membership?
  - Individual vs. opaque certificates?
- How to tolerate Byzantine behavior by member(s)?
  - Cannot prevent key release or denial-of-service...
  - Member proves correct protocol execution
- Group Barter
- Group Signatures

Related Work

- Cornell: Birman et al. (ISIS, Horus, Ensemble)
- UCSB: Melliar-Smith et al. (Totem)
- HUJ: Dolev et al. (Transis)
- JHU: Amir et al. (Spread)

- TIS: Balenson et al. (OFT)
- UTA: Lam, Gouda
- NSA Wallner et al. (LKH)
- IBM: Canetti et al.
- ETHZ: Carroni, Sun (???)

- Ingemarsson, Tang, Wong (ToIT 81)
- Burmester/Desmedt 94 (Eurocrypt 94)
- Steer/Diffie (Crypto'89)
- DeSantis/Vaccaro/Yung

Summary

Impact:
- IBM
- JHU
- LBNL (DoE)
- IRTF

CLIQUES web page: http://www.isi.edu/div7/cliques
- API documentation
- Publications
- Presentations

API code by request from gts@isi.edu

Collaborators:
- IBM Research, Johns Hopkins, LBNL, Nortel

People:
- G. Ateniese, O. Chevassut, D. Hasse, Y. Kim, G. Tsudik

Other current research

Integrated Multicast in Ad Hoc Networks (NSF)

Secure IP multicast (Nortel)

Survivability Using Controlled Security Services (DARPA)

Server-Assisted Digital Signatures (NSA)

Access Control in Collaborative Applications (DoE in subm.)

Low Latency Massive Data Transfers (Skunkworks)