Security Services for Internet Flows and Multicasts

Simon S. Lam

Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712
lam@cs.utexas.edu

Basic Cryptography

Symmetric Key System

a shared symmetric key

examples, DES, IDEA, RC4

Asymmetric Key System

a pair of private and public keys

examples, RSA, ElGamal, DSA, Rabin, FFS

Authentication Services

Needham-Shroeder Protocols (*CACM*, 1978)

- Kerberos (MIT, 1988)
- …

Secure Sockets

- SNP (U. Texas at Austin, 1993)
  - published in *Proceedings USENIX*, June 1994
- SSL (Netscape, 1996)

Motivation

Traditional network applications

- message-oriented unicast,
  e.g., email, file transfer, client-server

Emerging network applications

- flow-oriented, e.g., digitized video, stock quotes
- multicast, e.g., teleconference, software distribution

Problem 1: How to share a group key?

Problem 2: How to sign efficiently?

Secure Group Communications Using Key Graphs

by Chung Kei Wong, M. Gouda, and Simon S. Lam

in *Proc. ACM SIGCOMM '98*

available from **www.cs.utexas.edu/users/lam**

Confidential group communications

Examples

- teleconference
- information services
- collaborative work
- virtual private networks

Members share a key to encrypt/ decrypt group communications

Group key management

- Secure rekeying
  - after each join
  - after each leave
  - periodically
- Scalable server and protocols
  - for large groups with frequent joins and leaves

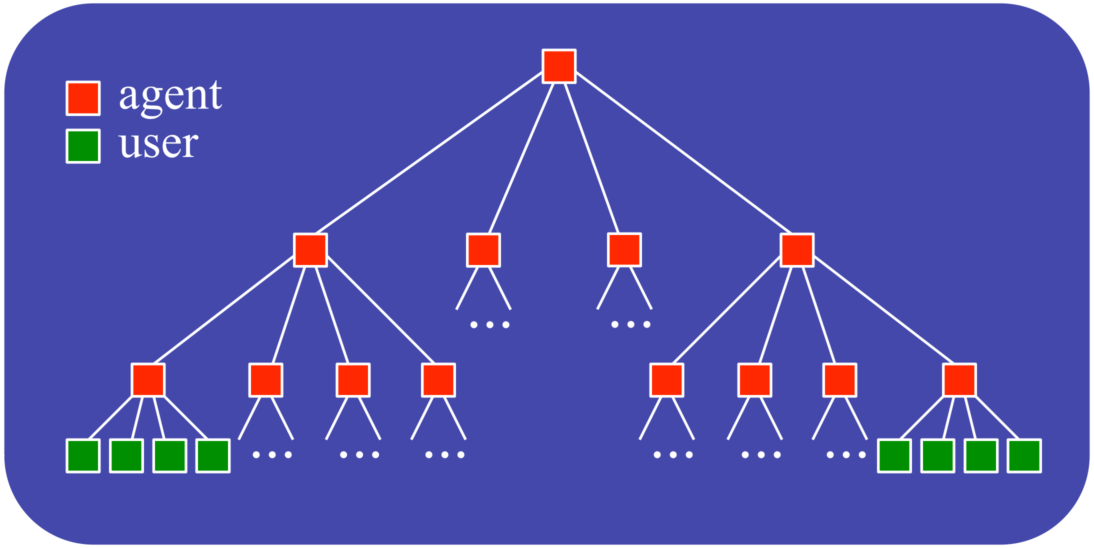
Assumptions

- Key server is trusted and secure
- An authentication service
  - for example, SSL
  - mutual authentication of server and joining user
  - distribution of a key shared by server and joining user (individual key)
- Access control by key server or an authorization service

Secure rekeying

- Non problem after a join
  - new group key encrypted by old group key
  - one encryption/rekey msg for all existing users
- After a leave has occurred
  - new group key encrypted by individual key of each user n-1 encryptions/rekey messages for group size n not scalable

Iolus approach [Mittra 1997]



A hierarchy of security agents

No globally shared group key

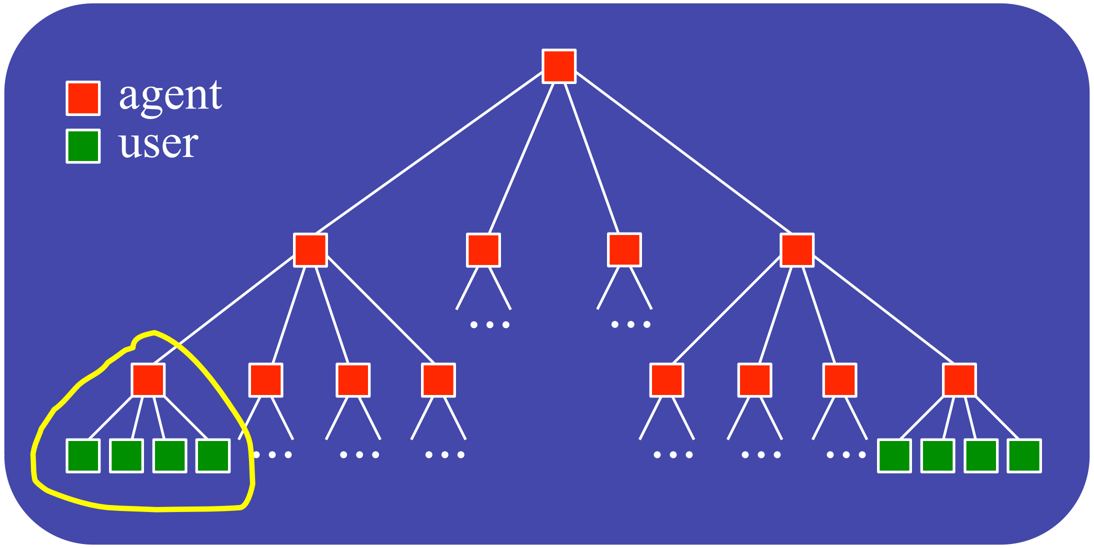
- join/leave affects local subgroup only

Agents forward message key

- decrypting and re-encrypting with subgroup keys

Requirement: many trusted agents

Iolus approach [Mittra 1997]



A hierarchy of security agents

No globally shared group key

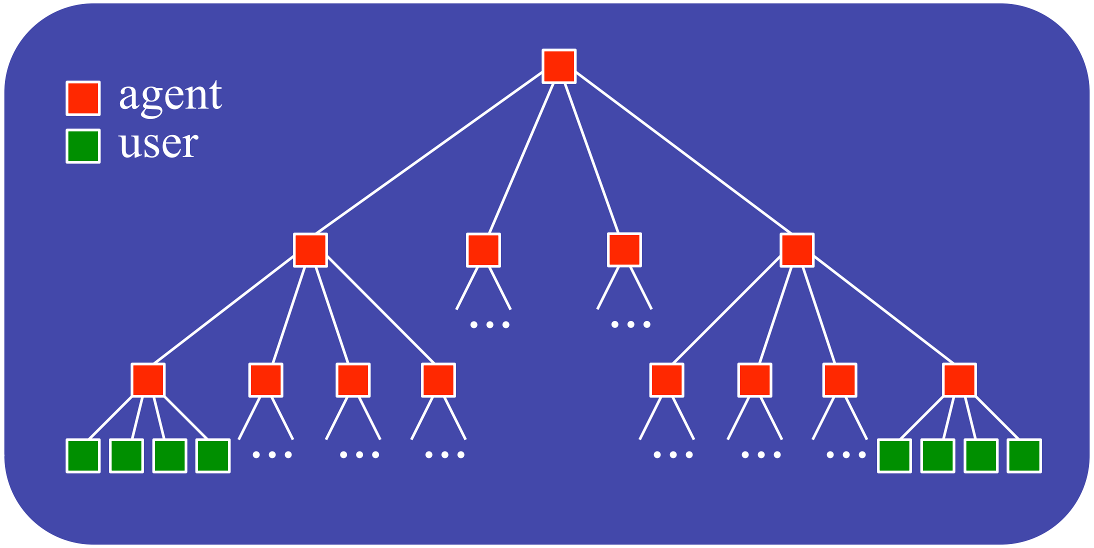
- join/leave affects local subgroup only

Agents forward message key

- decrypting and re-encrypting with subgroup keys

Requirement: many trusted agents

Iolus approach [Mittra 1997]



A hierarchy of security agents

No globally shared group key

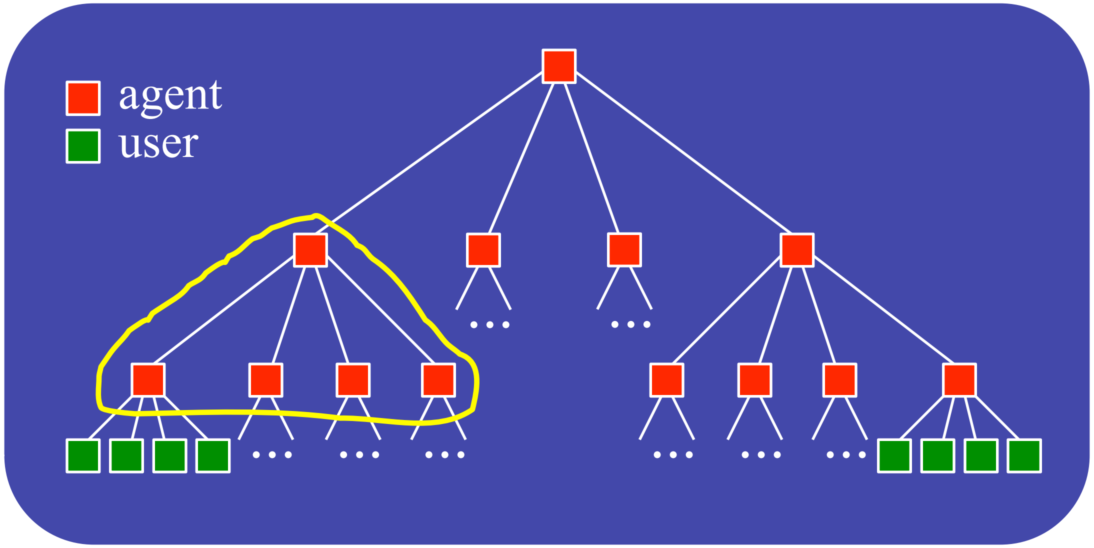
- join/leave affects local subgroup only

Agents forward message key

- decrypting and re-encrypting with subgroup keys

Requirement: many trusted agents

Iolus approach [Mittra 1997]



A hierarchy of security agents

No globally shared group key

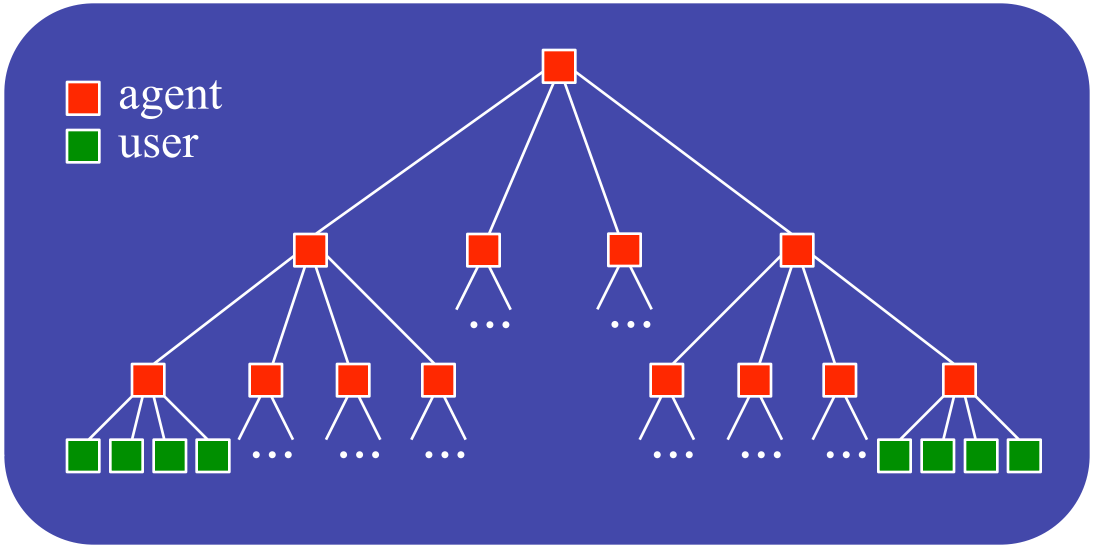
- join/leave affects local subgroup only

Agents forward message key

- decrypting and re-encrypting with subgroup keys

Requirement: many trusted agents

Iolus approach [Mittra 1997]



A hierarchy of security agents

No globally shared group key

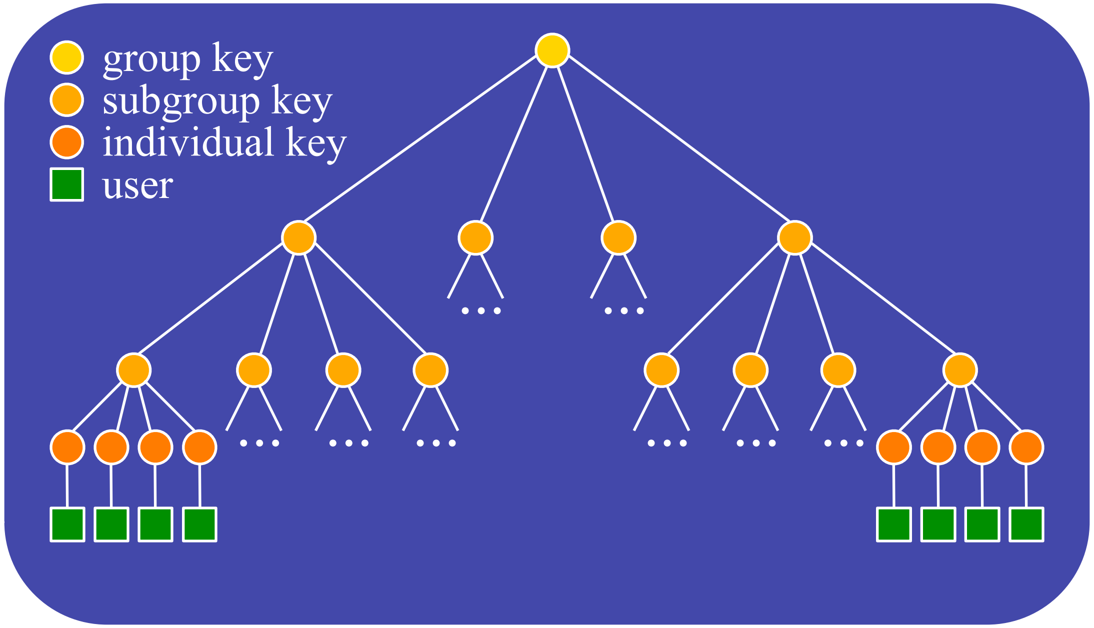
- join/leave affects local subgroup only

Agents forward message key

- decrypting and re-encrypting with subgroup keys

Requirement: many trusted agents

Our approach



A hierarchy of keys

Multiple keys for each user

- user has every key along path to root

A single trusted key server is sufficient (may be replicated for reliability)

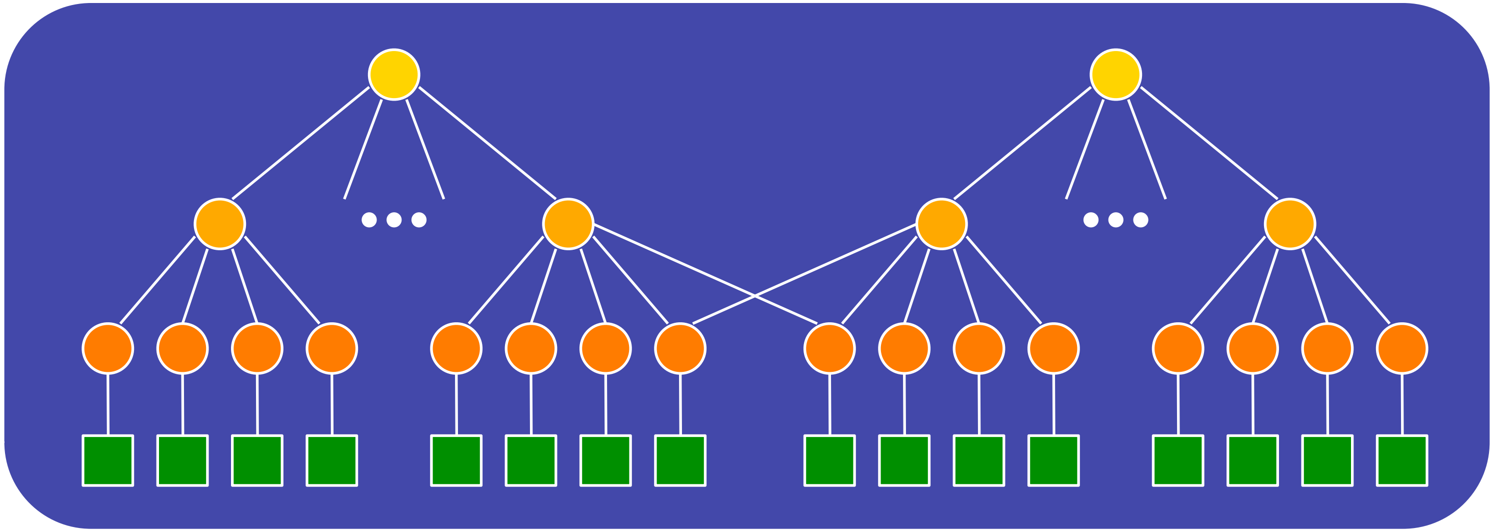
Key graph

For a single secure group

- key tree sufficient for scalability

Multiple secure groups

- merging multiple trees into a graph



Rekeying strategies

User-oriented

Key-oriented

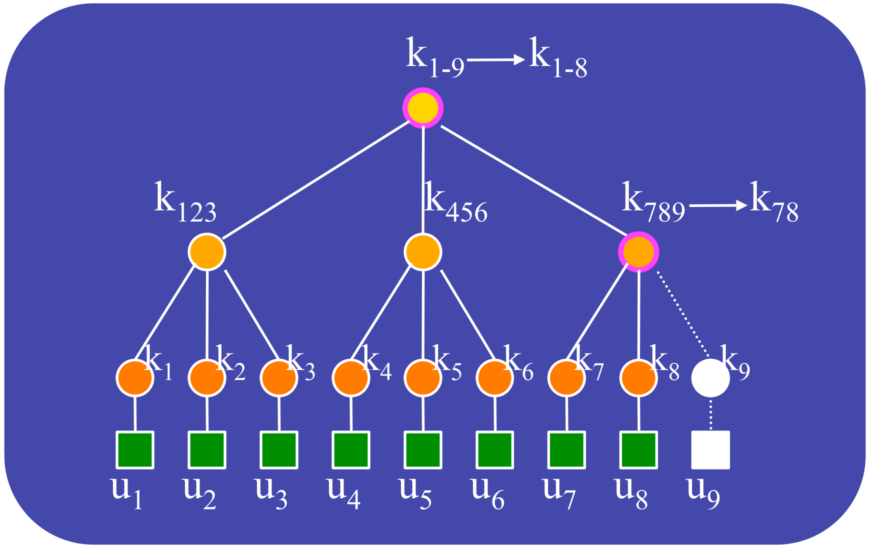
Group-oriented

User-oriented rekeying

Select new keys needed by a user, form a rekey message and encrypt it

Multiple rekey messages

Most work on server, least work on user



Leaving

s ® $\{u_1, u_2, u_3\}$: $\{k_{1-8}\}_{k_{123}}$

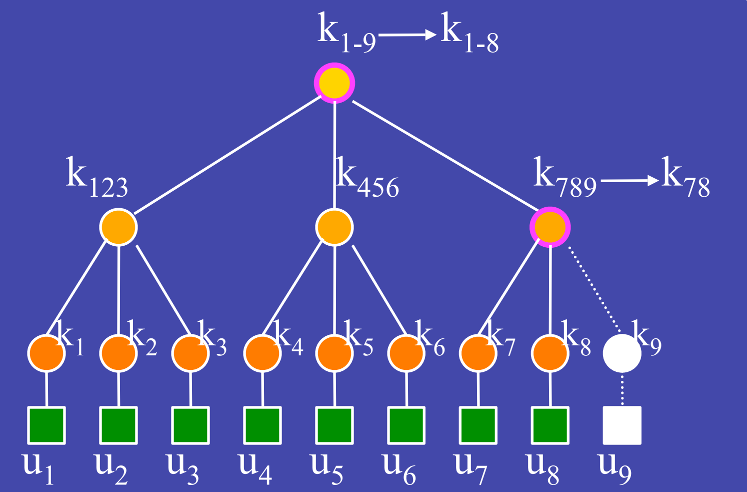
s ® $\{u_4, u_5, u_6\}$: $\{k_{1-8}\}_{k_{456}}$

s ® u_7 : $\{k_{1-8}, k_{78}\}_{k_7}$

s ® u_8 : $\{k_{1-8}, k_{78}\}_{k_8}$

Key-oriented rekeying

- Encrypt each new key, then compose rekey messages
- Multiple rekey messages
- Less work on server than user-oriented



Leaving

s ® $\{u_1, u_2, u_3\}$: $\{k_{1-8}\}_{k_{123}}$

s ® $\{u_4, u_5, u_6\}$: $\{k_{1-8}\}_{k_{456}}$

s ® u_7 : $\{k_{1-8}\}_{k_{78}}, \{k_{78}\}_{k_7}$

s ® u_8 : $\{k_{1-8}\}_{k_{78}}, \{k_{78}\}_{k_8}$

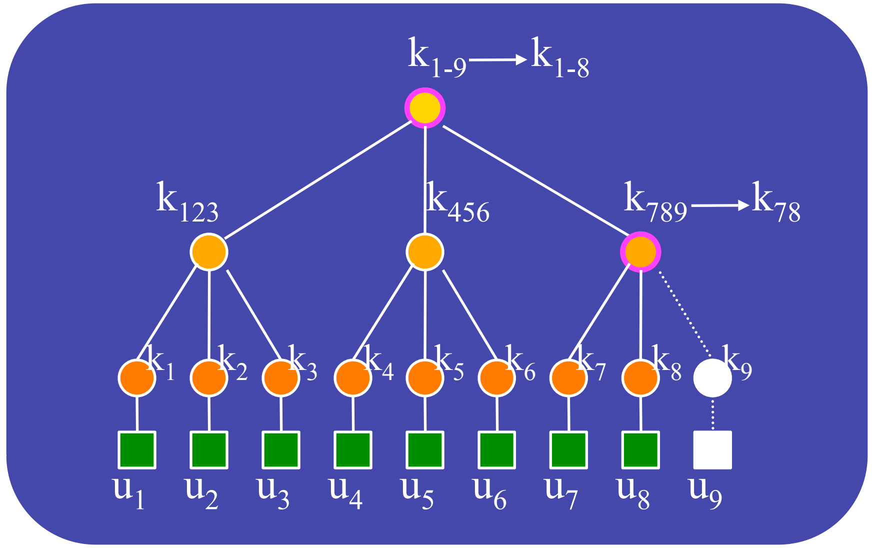
Group-oriented rekeying

One rekey message containing all encrypted new keys

Message size $O(\log n)$

Each user decrypts what it needs

Least work on server, more work on user



Leaving

s ® $\{u_1,\ldots,u_8\}$:

$$\{k_{78}\}_{k_7}, \{k_{78}\}_{k_8},$$
$$\{k_{1-8}\}_{k_{123}}, \{k_{1-8}\}_{k_{456}},$$
$$\{k_{1-8}\}_{k_{78}}$$

Experiments

Two SGI machines connected by 100 Mbps Ethernet

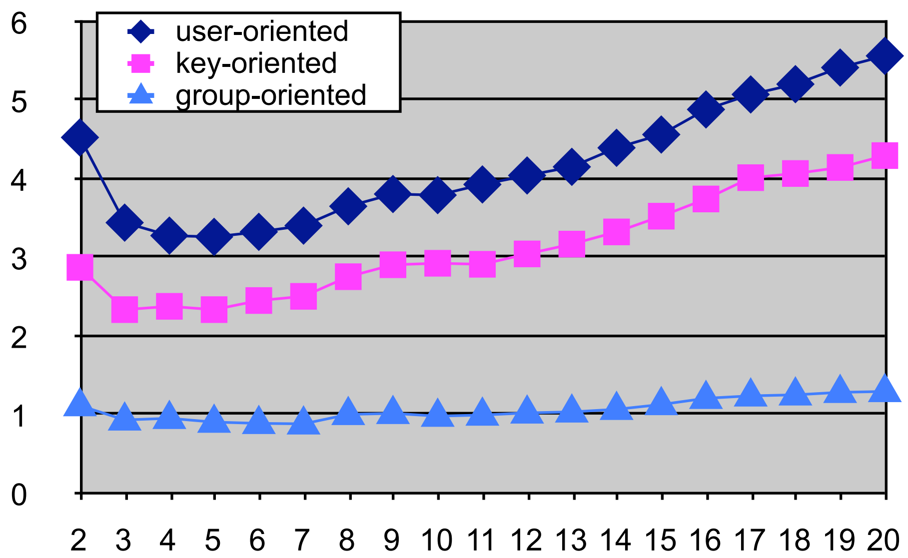
- server on one, users on the other

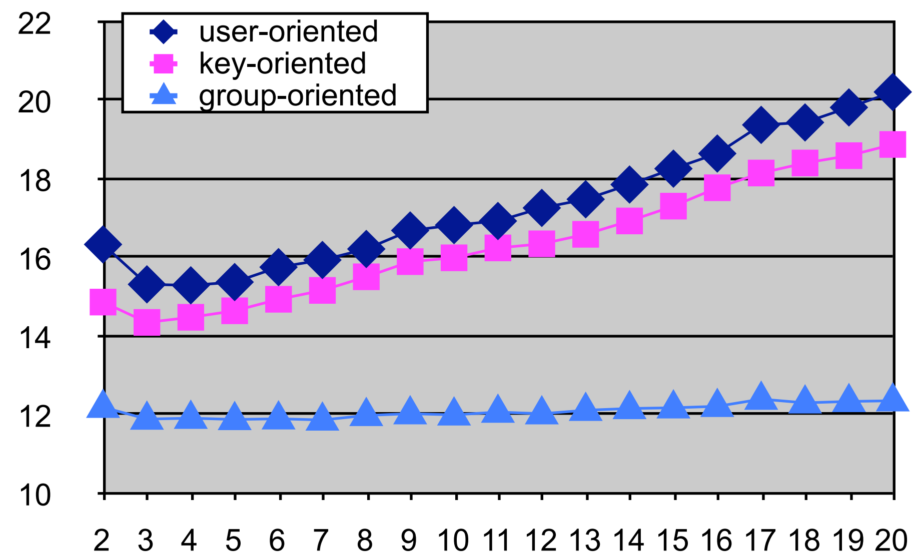
Rekey messages sent as UDP packets

DES, MD5, RSA from CryptoLib

- *n* joins, then 1000 randomly generated join/leave requests

Server processing time versus key tree degree

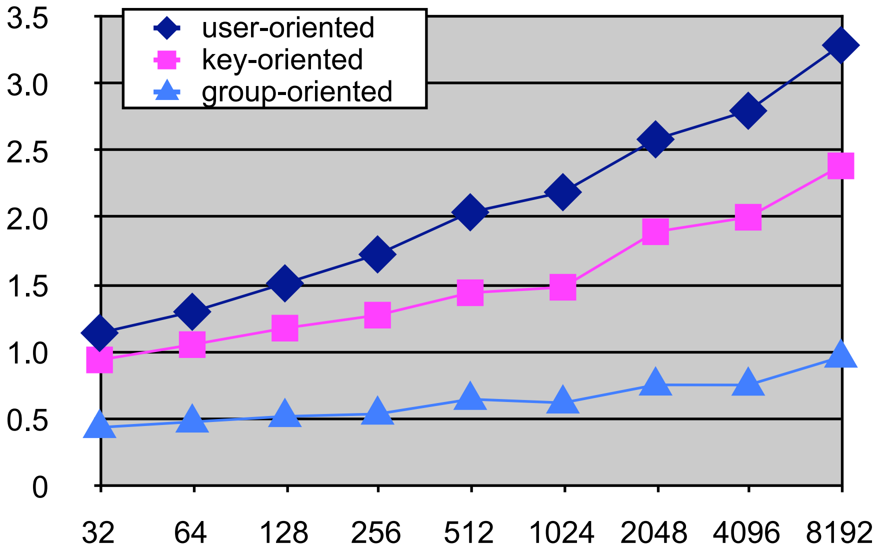


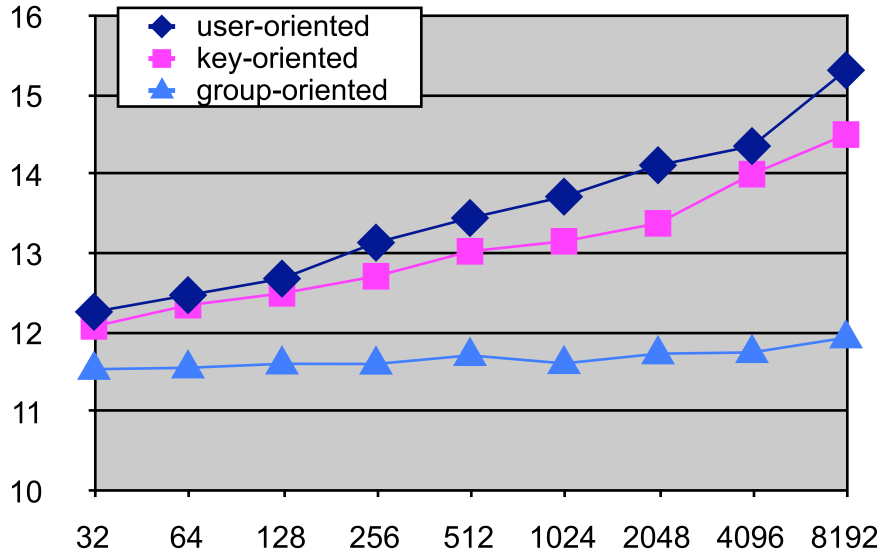


Initial group size 8192

4 is optimal degree (analytic result)

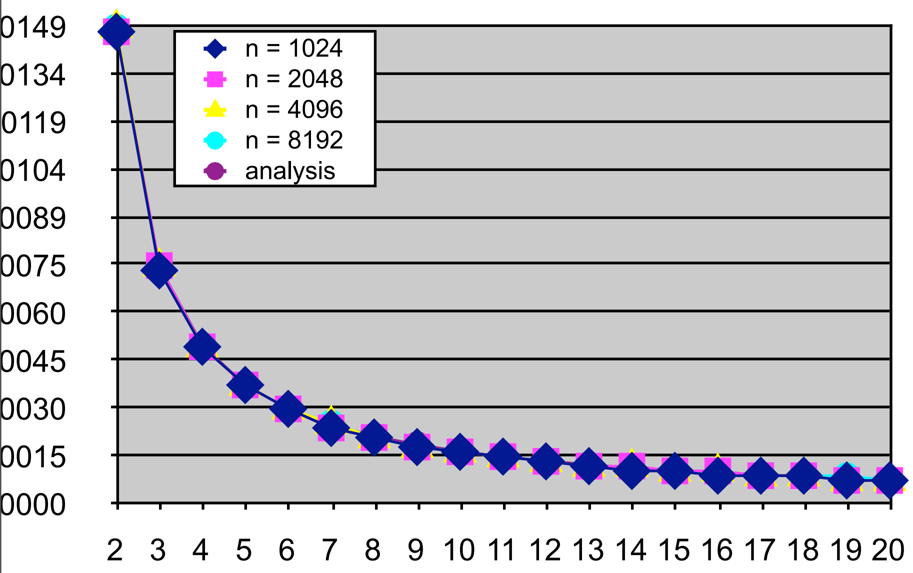
Server processing time versus group size

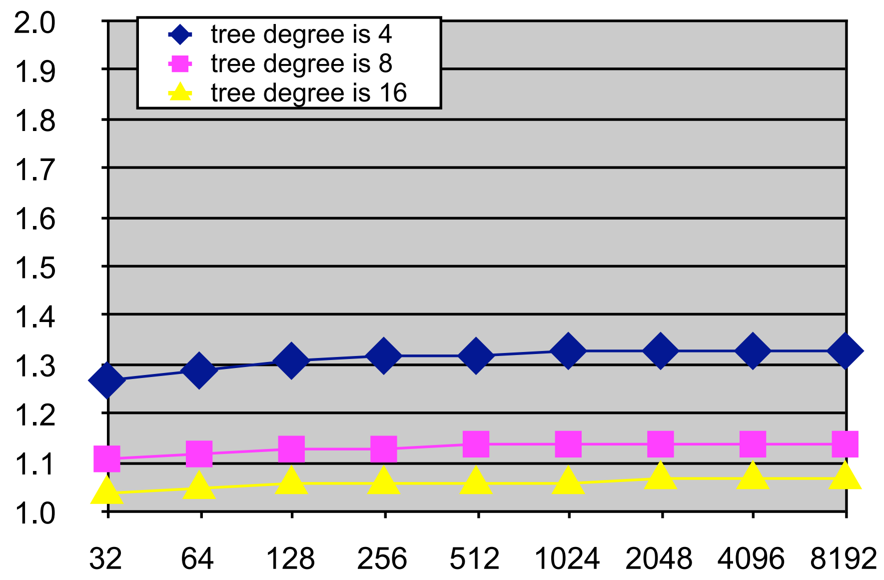




Increases linearly with logarithm of group size

Number of key changes by a user (per request)





Very close to analytic result, $d \,/\, (d-1)$

Rekey messages sent by server

With encryption and signature
(initial group size 8192, key tree degree 4)

Rekey messages received by user

With encryption and signature
(initial group size 8192, key tree degree 4)

Conclusions

Scalable performance demonstrated experimentally and analytically

Group-oriented rekeying requires smallest processing time and transmission bandwidth of server

Hybrid approach with use of user- or key-oriented rekeying for users with limited capabilities

Hybrid approach with use of some Iolus agents at strategic locations (C. Partridge)

Multiple secure groups (work in progress)

Security issues for flows and multicasts

Confidentiality of group communications

- this paper

Authenticity, integrity, non-repudiation

- Digital Signatures for Flows and Multicasts
  IEEE ICNP '98, Austin, October 1998