

# Secure Internet Live Conferencing

---

Christian Horchert <ch@vakuum.net>

Frank Becker <fb@alien8.de>

Chaostreff Dresden <http://www.c3d2.de>

2003-12-28

# What is it about?

Protocol for **authenticated** and **encrypted**  
Live Conferencing

Supports chat and instant messenger like systems.

/join 20C3

# Agenda

- History/Future of SILC
- Protocol
- Using SILC
- Software
- Silc network

# History

- designed by Pekka Riikonen, started '96
- work stopped several times
- 1<sup>st</sup> public release in 2000,
- SILC-client 1.0 Oct. 2003
- currently maturing of protocol
- protocol specs submitted to IETF, currently in a draft phase (will become RFC soon)
- Version 1 of silcd, toolkit to follow soon

# The Protocol



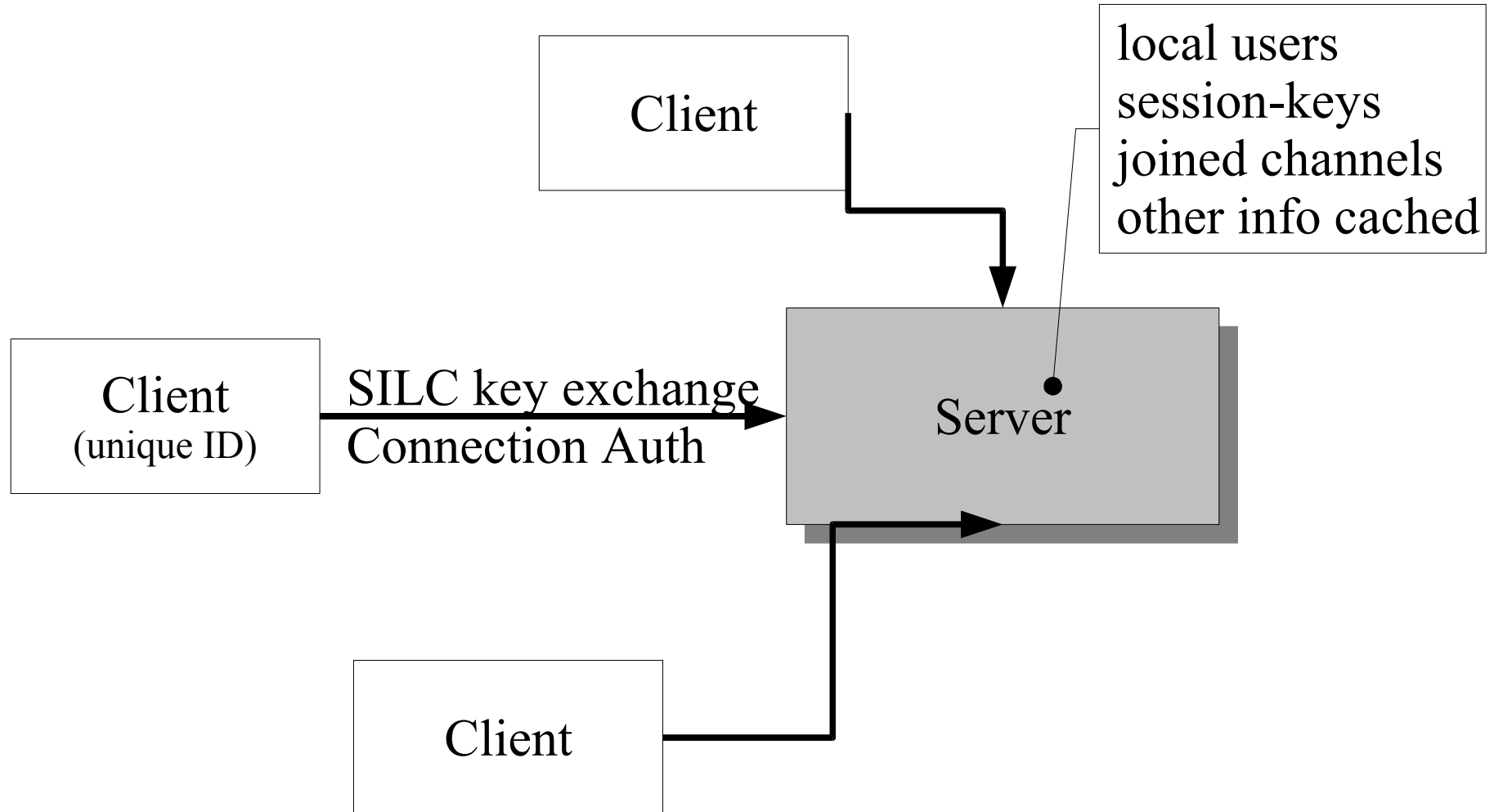
# The Protocol

## Key-Features:

- **all** messages are **encrypted** and **authenticated**
- Keys managed by server or user
- messages are sent through server-net
- File transfer via sftp

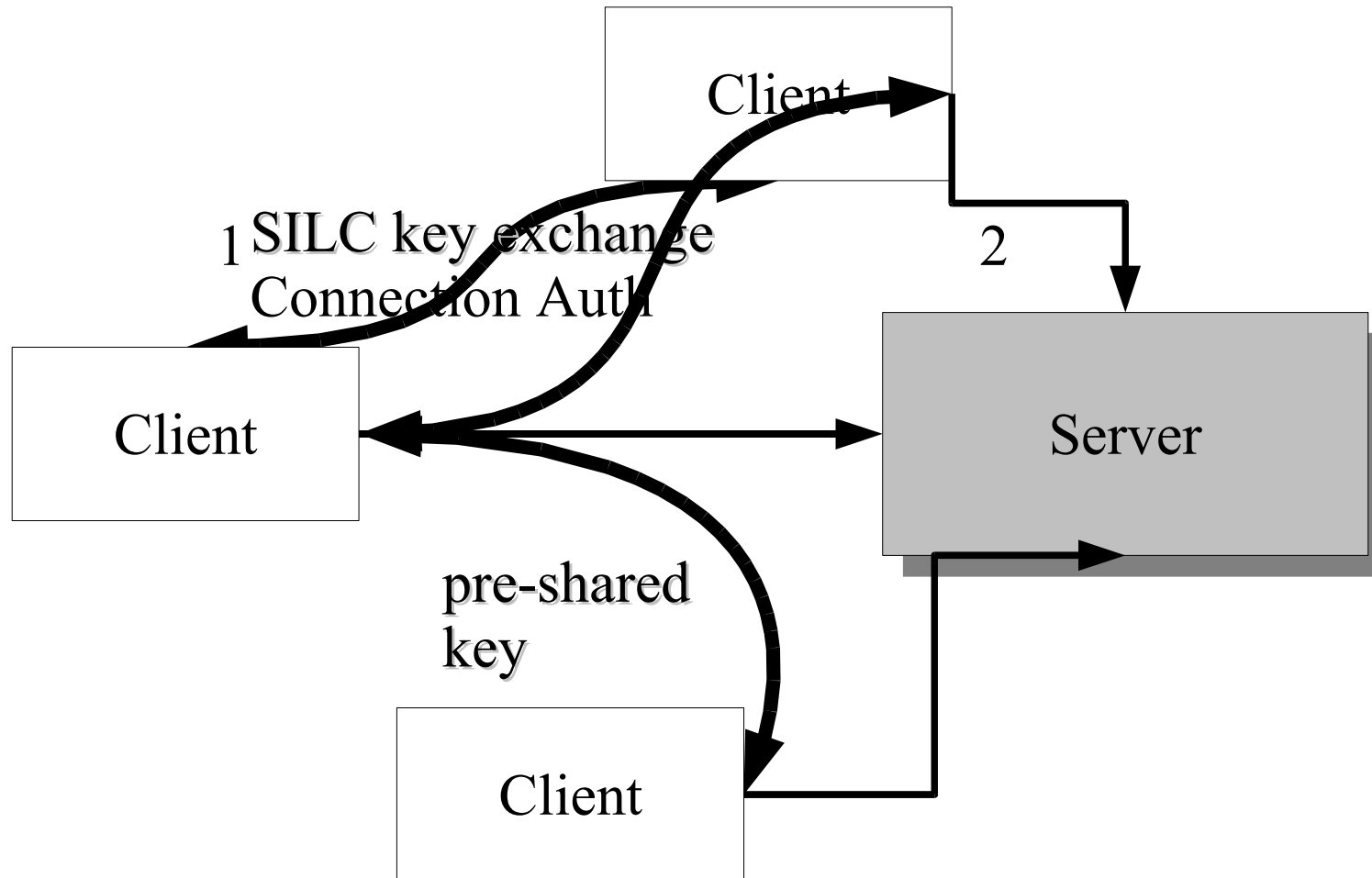
# Network Topology

normal client traffic



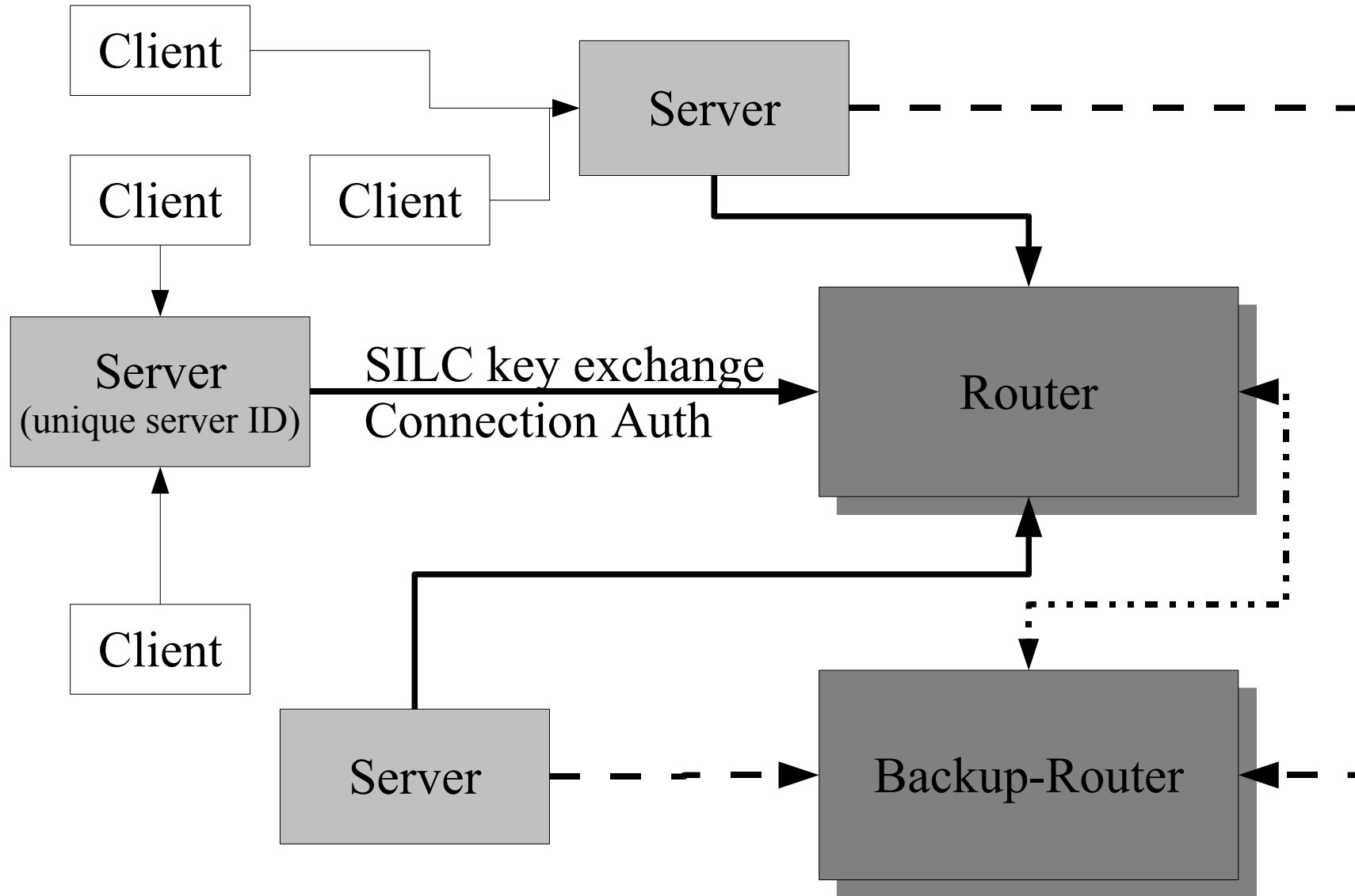
# Network Topology

client-client traffic



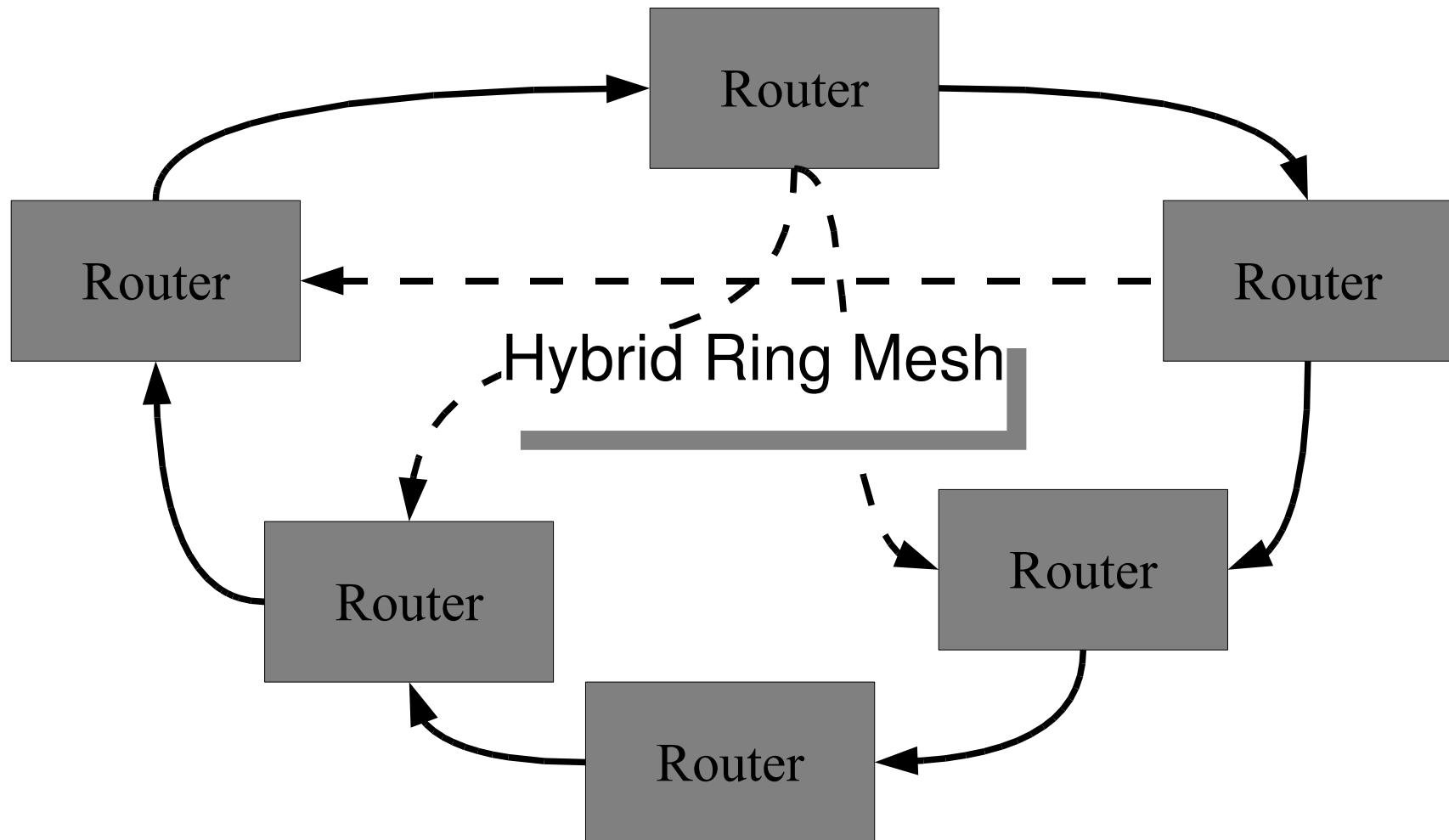
# Network Topology

server-router traffic, Cell

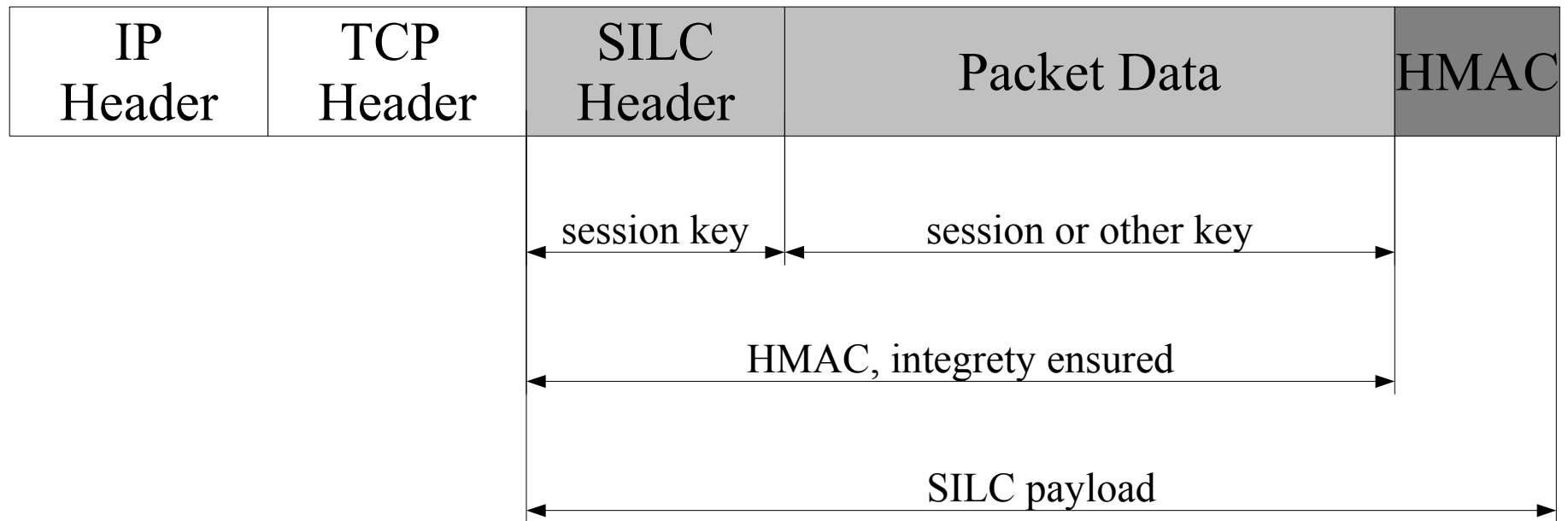


# Network Topology

router-router traffic



# Packet format overview



# SILC Key Exchange (SKE)

- (1) initiator sends properties (cipher, hash function, HMAC function, public key algorithm)
- (2) responder selects properties
- (3) Diffie-Hellman key exchange exchange public keys too
- (4) mutual authentication mode

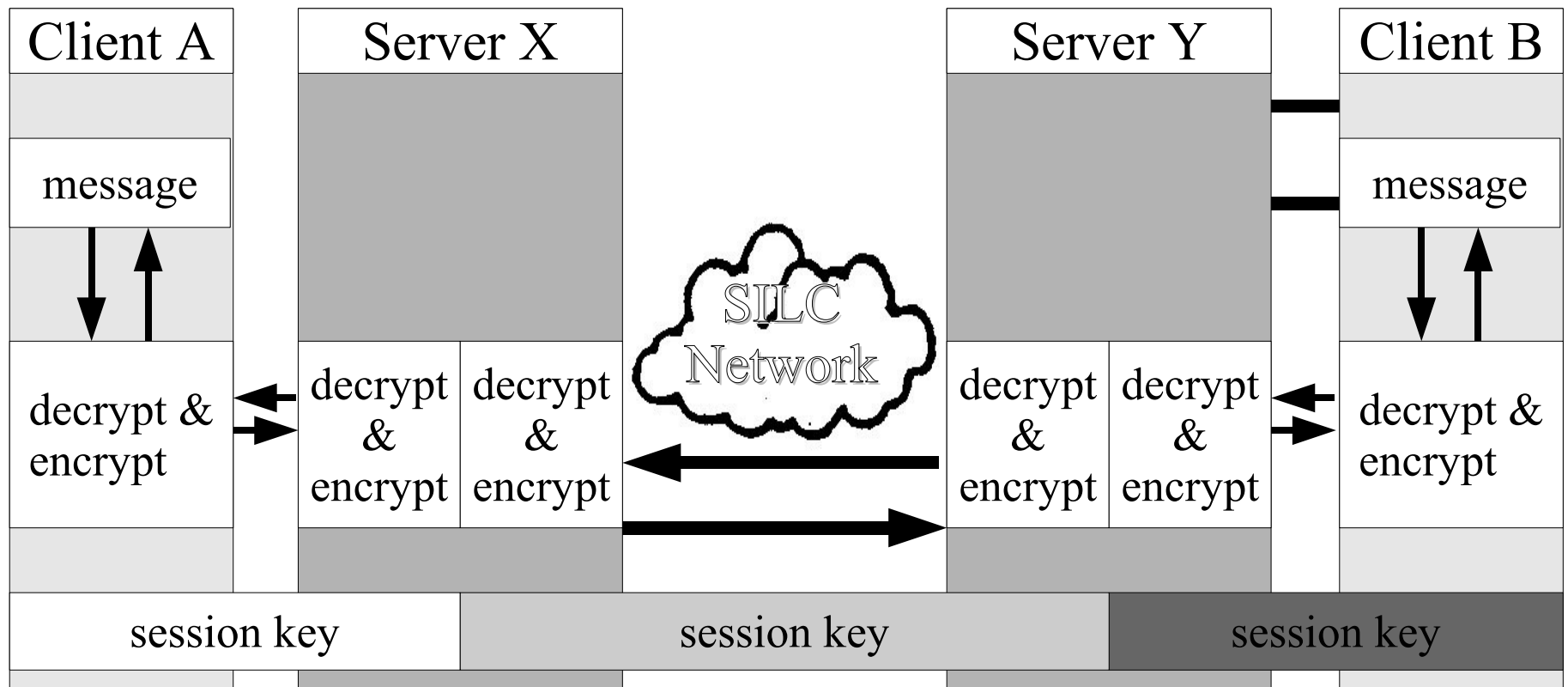
Result: Session key

# SILC Connection Authentication

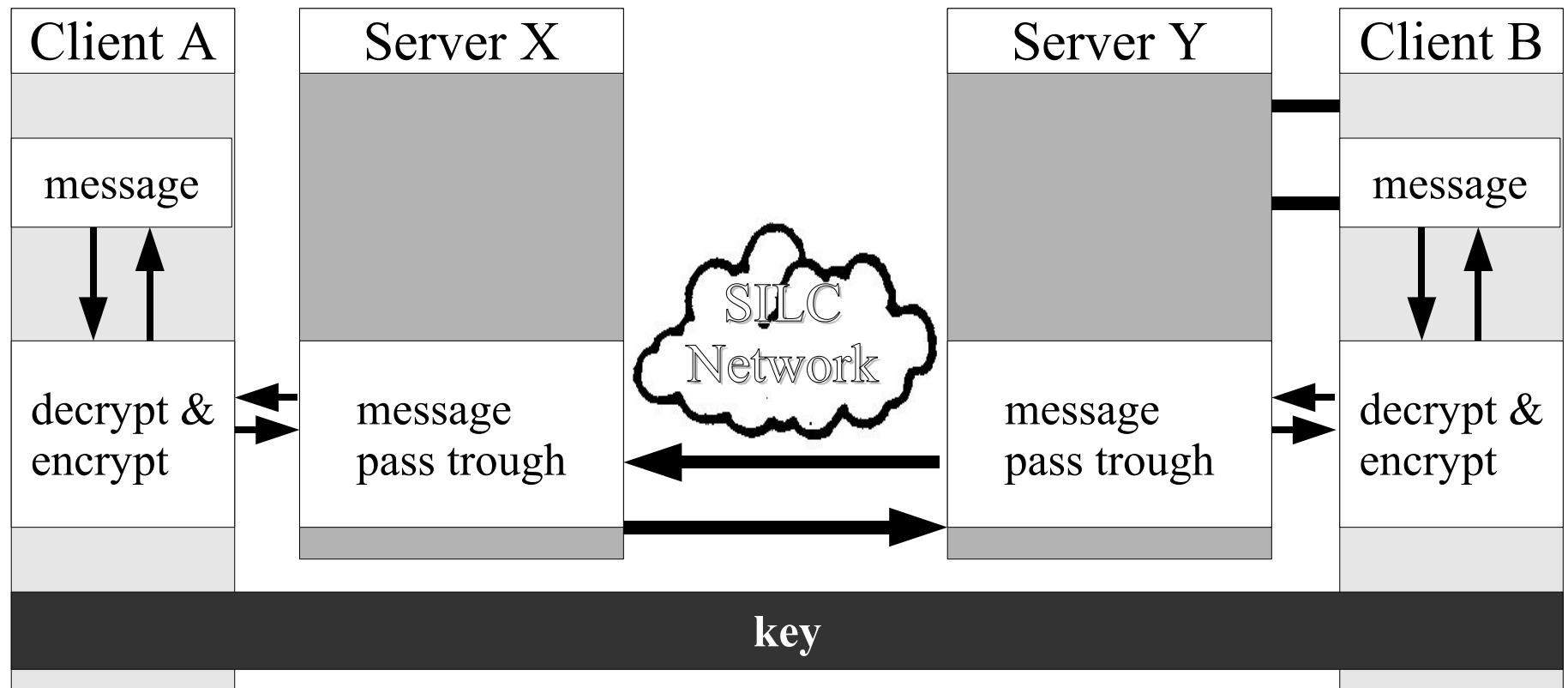
- Done right after SKE
- Authenticate connecting parties (e.g. client to server)
- based on  
    passphrase (packet encrypted) or  
    public-key (challenge sent to client)

Result: authenticated client

# Private Message w/ Session Keys



# Private Message w/ Private Key



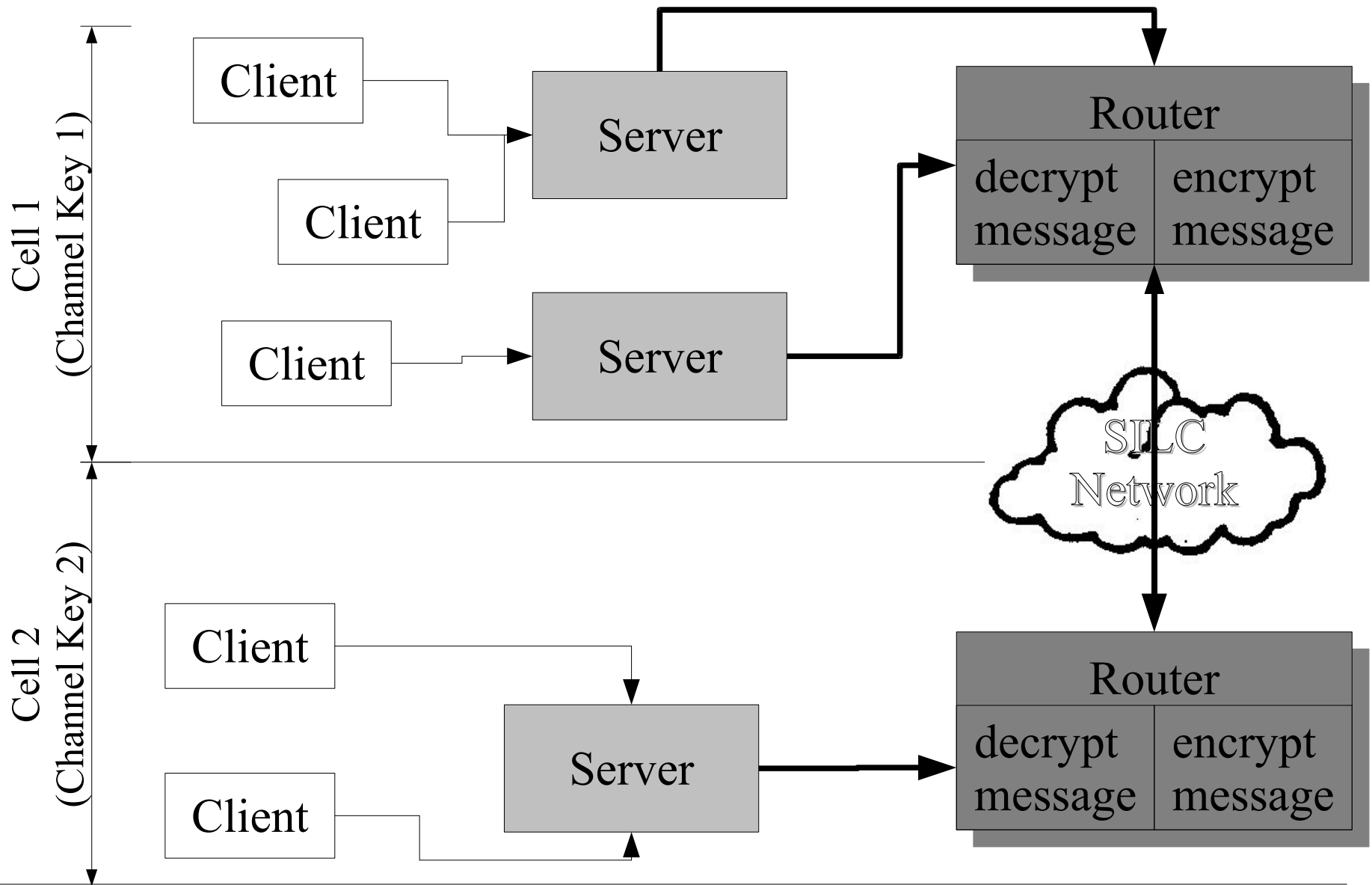
# Channels

- Named group, all joined clients receive messages
- Channel names are unique, max. 256 characters

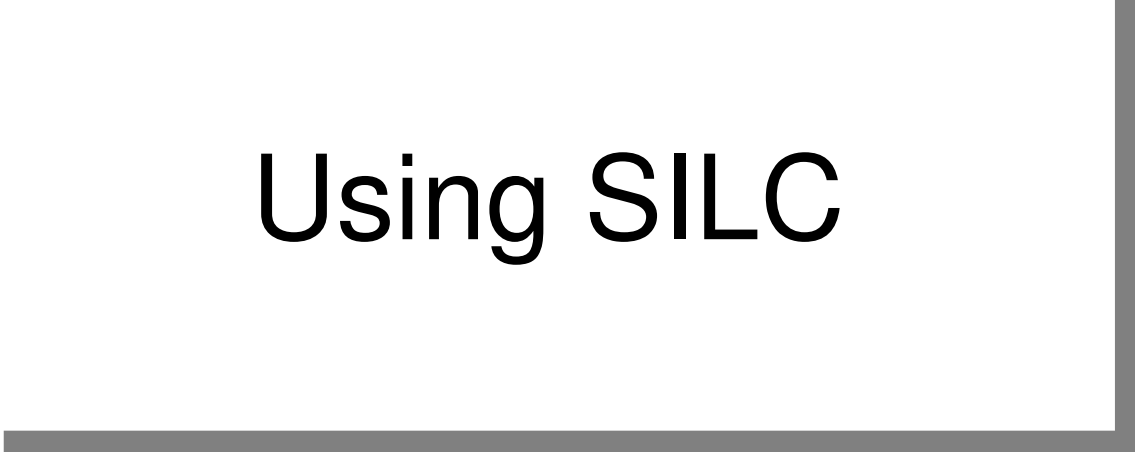
# Channel Message Delivery

- All messages in a **cell** are encrypted and authenticated by one channel key
- Messages between cells are encrypted by session key
- regular re-keying
- in addition: key is generated if channel is founded or client joins or leaves
- Private Keys are possible (passphrase, publickey)

# Channel Message Delivery illustrated



# Using SILC



# Using SILC: Nicks

- no unique nick names
- no nick services (avoid nick wars)
- authenticated and identified by public key
- attributes and present modes
- other useful modes
  - blocking non op msgs
  - blocking private msgs
  - marking / blocking bot msgs
  - reject watching

# Using SILC: Channels

- joining and founding a channel (save your key!!!)  
  /cmode +f channel
- no channel services needed (takeovers are difficult)
- can be private, secret, moderated, invite only, limited
- setting channel keys

# Using SILC: Channel w/ secret

- get key by mail, phone, ...
  - /JOIN channel
  - /CMODE +k
  - /KEY CHANNEL channelname set secret
- must be done by each client
- => UserD (and server admin) cannot see msgs

# Using SILC: Channel w/ public channel keys

- get key by mail, /getkey, ...  
  /JOIN channel  
  
  /CMODE channel +C +pubkeyUserA \  
    +pubkeyUserB +pubkeyUserC
- UserB and UserC join w/ /JOIN channel -auth
- => UserD (and server admin) cannot see messages

# Using SILC: Messaging

- **msgs can be signed (use /SMSG)**  
[?] fukami: signed msg, you don't have the key  
[S] fukami: signed msg, you've got the key
- **MIME msgs**  
/SCRIPT LOAD silc-mime.pl  
/MMSG -CHANNEL channelname path/to/file
- **private msg protection with keys**

# Using SILC: Securing priv msgs

- shared secret by phone, mail, letter ...

```
/KEY MSG UserA set secret
```

```
/KEY MSG UserB set secret -responder
```

- => secured communication

# Using SILC: File Transfer

- send file p2p using sftp

```
/FILE SEND path/to/file UserB
```

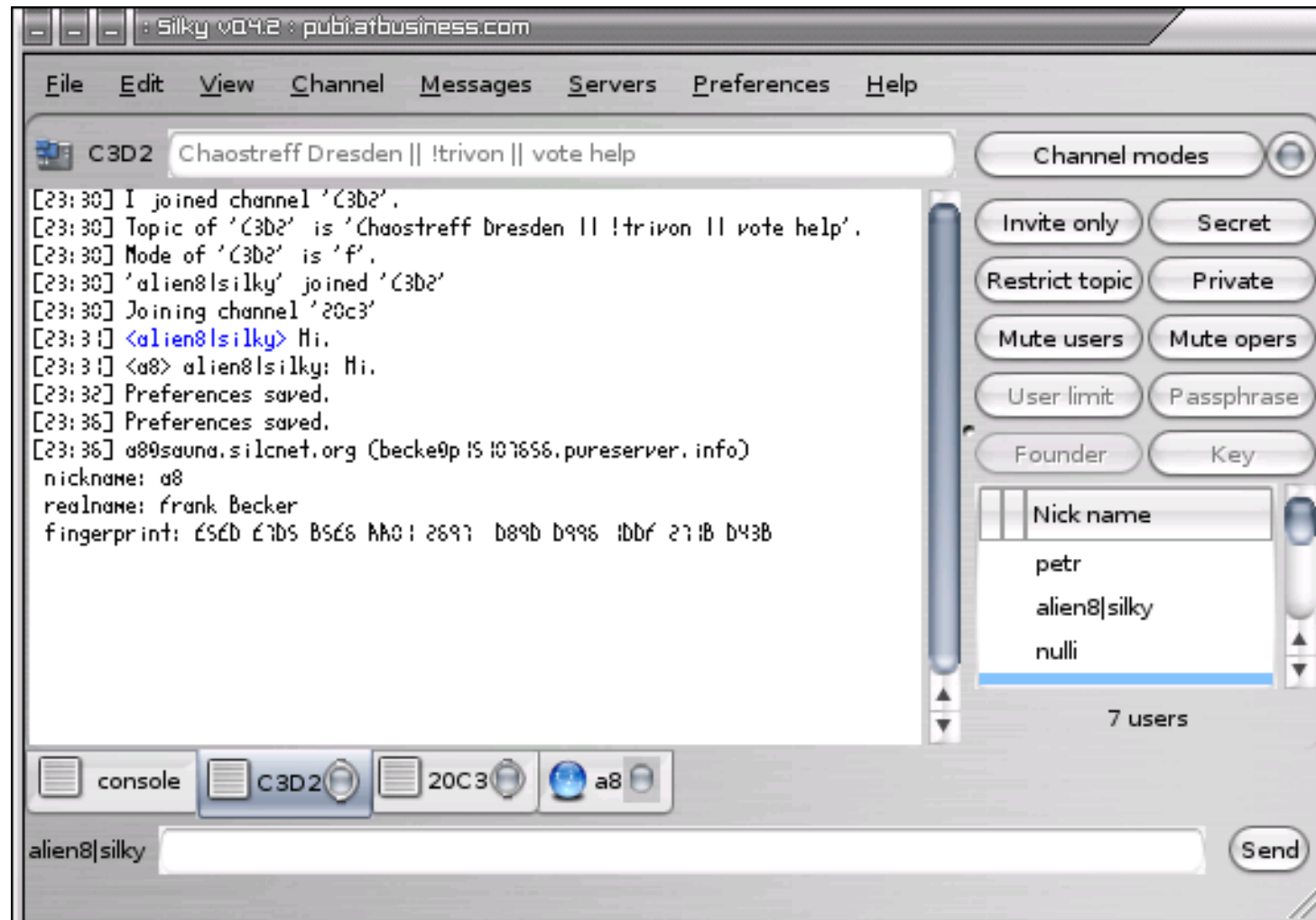
```
/FILE ACCEPT UserB
```

```
/FILE CLOSE (to close session immediately)
```

**use** `-no-listener` if behind NAT

# Programmes and Frameworks

- silcd
- silc-client
- silc-toolkit
- Silky
- jsilc
- samadhi



# SILC Network

- Connect to a server next to you
- `silc.silcnet.net` is round-robin DNS to all the SILCNet servers
- multiple server connections are possible



# few Credits

- Pekka Riikonen (Main developer, SILCNet coordinator/administrator)
- Timo "cras" Sirainen (Irssi/SILC client)
- Jochen "c0ffee" Eisinger (SILC plugin)
- Toni Willberg (Silky)
- Giovanni Giacobbi (silconfig, silclog, silcd bugfixes, RPM packages)
- Lubomir "salon" Sedlacik (NetBSD package, project server administrator)
- Tamas Szerb (Debian packages)
- Mika "Bostik" Boström (Man-pages, Bughunting)
- Juha Räsänen (ElGamal implementation)
- Ville Räsänen (Client side of STATS-command, Some ROBXOdoc formatting, Bughunting)
- Patrik Weiskircher (whois attributes, bugfixing)

# Links

- Home of SILC: manuals, FAQs, white papers, pre-compiled packages of silcd, silc-client (rpm, deb) and much more: <http://silcnet.org>
- Irssi-plugin: <http://penguin-breeder.org/silc/>
- Silky: <http://silky.sf.net>
- Ports for NetBSD, FreeBSD, OpenBSD, Darwin