# SSH (Secure Shell)
# Secure Remote Login

Tatu Ylönen

ylo@cs.hut.fi

# What is SSH?

- Secure login connections over an insecure network
- Does not trust the network or its services
- File transfers
- X11 and TCP/IP forwarding
- Strong encryption and authentication
- Replacement for rlogin, rsh, rcp, rdist, telnet

# Threat Model

Basic assumption: Attacker has control over the network

# Interception of All Transmitted Data

- Passwords
- Authorization cookies
- Trade secrets
- Personal information

# Connection Hijacking

- Can be done without either party noticing anything; allows modification of data in both directions
- One-time passwords and any other methods that provide added security during the authentication phase only are not secure
- Firewalls can sometimes be attacked by hijacking the connection after authentication
- Software for hijacking commercially available now

# Other Active Attacks

- Pretending to be another host
- DNS spoofing
- Routing spoofing
- NFS or Windows file sharing spoofing
- NIS spoofing

# Denial of Service

- Difficult to prevent
- Usually no leak of sensitive information or data loss

# Summary of Internet Security

- Current IP protocol cannot be trusted
- Higher level protocols (e.g., TCP and UDP) often assume that IP can be trusted
- Information security must be provided by application level mechanisms

Strong cryptography is the only visible solution

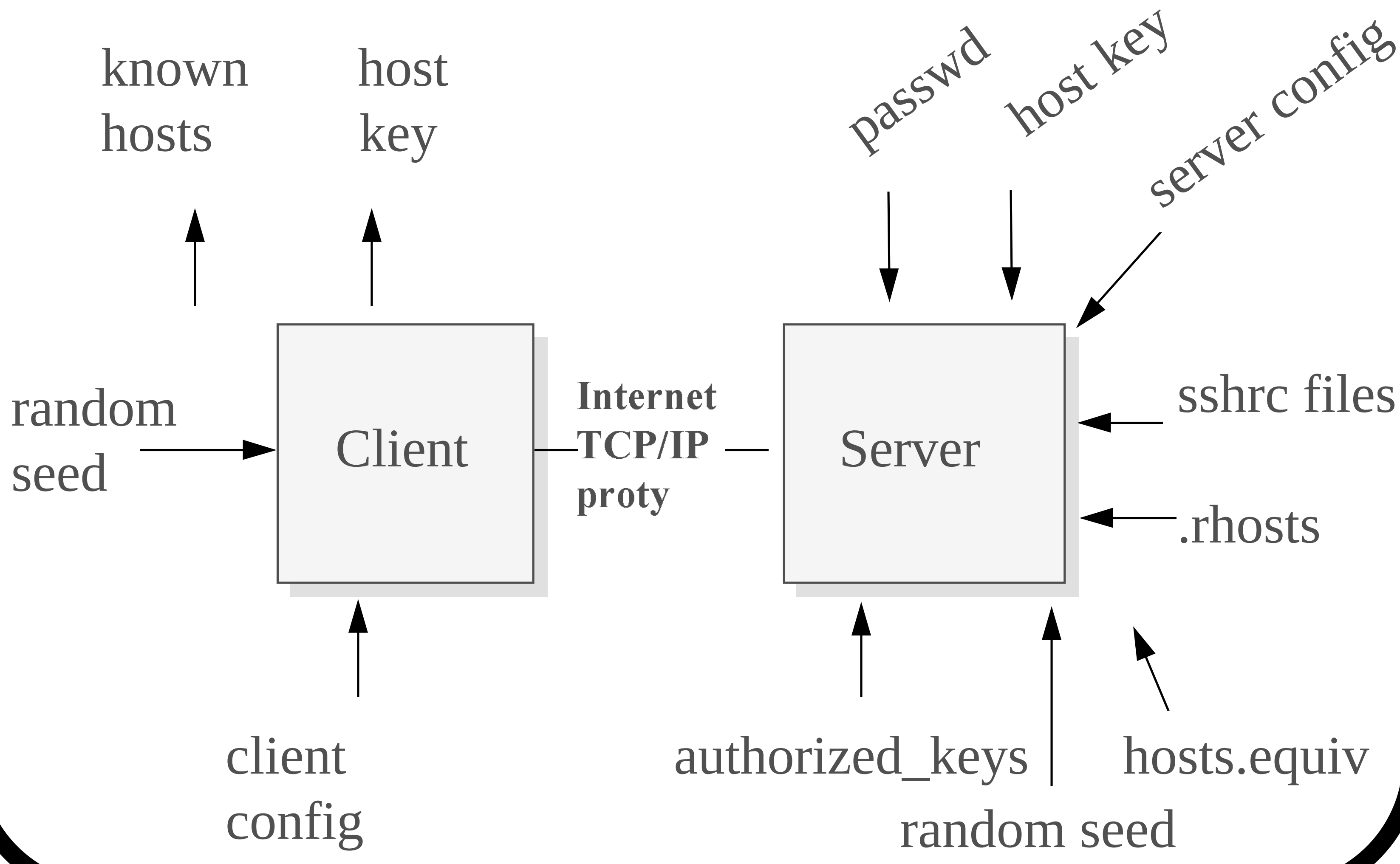
# SSH Secure Remote Login

- Strong authentication
  - » Does not trust the network
- Server machine authenticated with RSA
  - » Prevents “Trojan horse” attacks
- Fully distributed key management
- All traffic is automatically encrypted
- Protection of data integrity
- X11 forwarding
- TCP/IP forwarding
- Authentication agent (in future possibly on smart card)

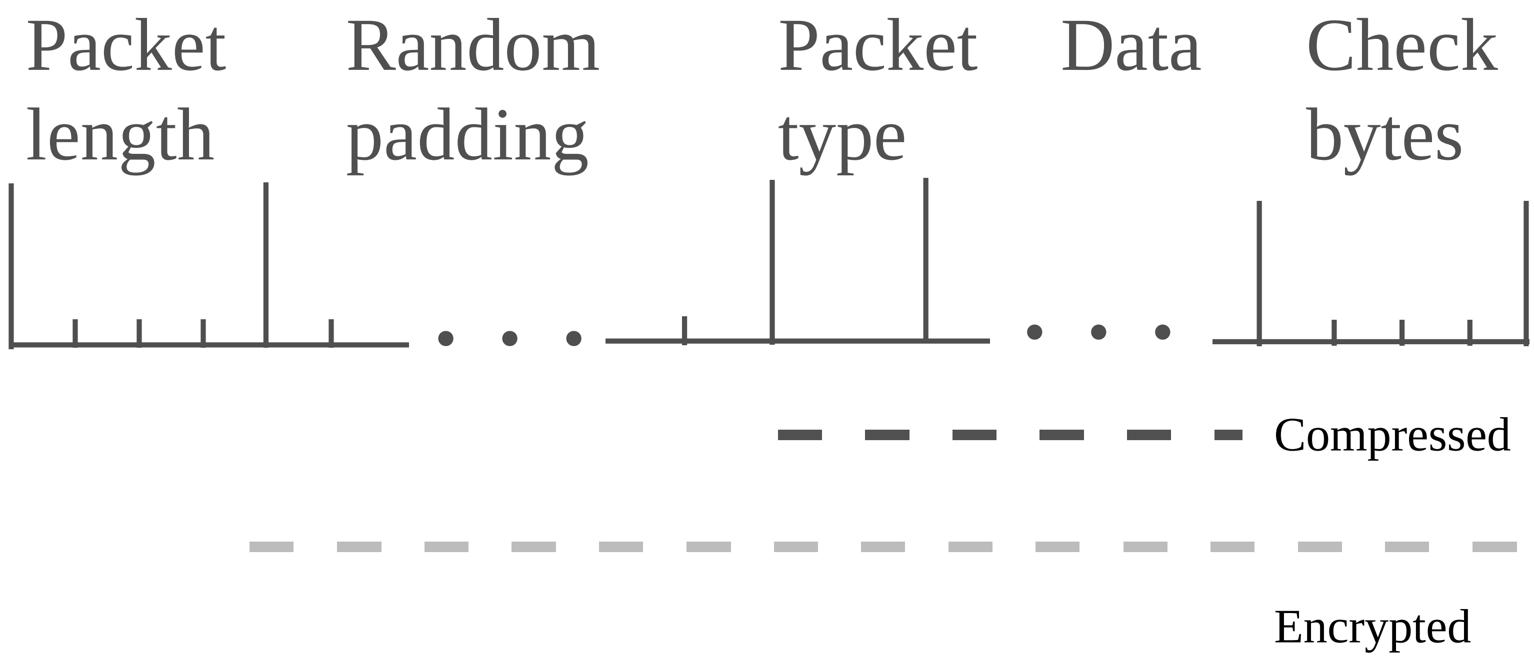
# SSH Features (cont´d)

- Optional compression of all tranmitted data
- Easy to use
  - » X11 (DISPLAY and xauth data set automatically)
  - » Automatically 8-bit clean
  - » Exit status is returned
  - » Can run arbitrary commands on a pty

# Operating Environment



# Packet Format



# Protocol

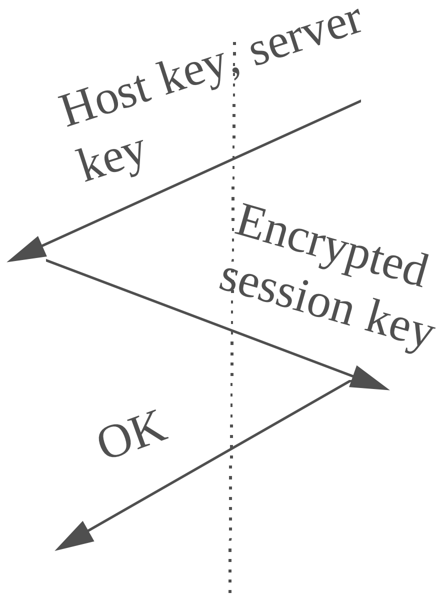
- Connection establishment
- Key exchange and server authentication
- User authentication
- Preparatory operations
- Interactive session

# Key Exchange and Server Authentication

## Client

Client
verifics
host key

Properly
encrypted
confirmation
authenticates server




## Server

Host key: 1024 bit RSA
Server key: 768 bit RSA
- regenerated hourly

Session key 256 bits
double-encrypted with RSA

Server decrypts session key
and enables encryption

# Key Exchange (cont´d)

- Server is RSA-authenticated
- Client chooses session key and algorithm
- Server key is discarded and regenerated every hour to make decrypting intercepted historical sessions imposible even if host key is later compromised

# User Authentication

- Client-driven
- Server responds OK, fail, or challenge
  - Password
  - RhostsRSA
  - RSA
- RSA authentication is challenge-based:
  - » Client sends public key
  - » Server checks if key is acceptable, generates 256 bit random challenge, RSA-encrypts
  - » Client RSA-decrypts and computes MD5 of challenge (+ other data to bind the response to a particular connection)

# Preparatory Phase

- PTY allocation
- X11 forwarding
- TCP/IP forwarding
- compression
- agent forwarding

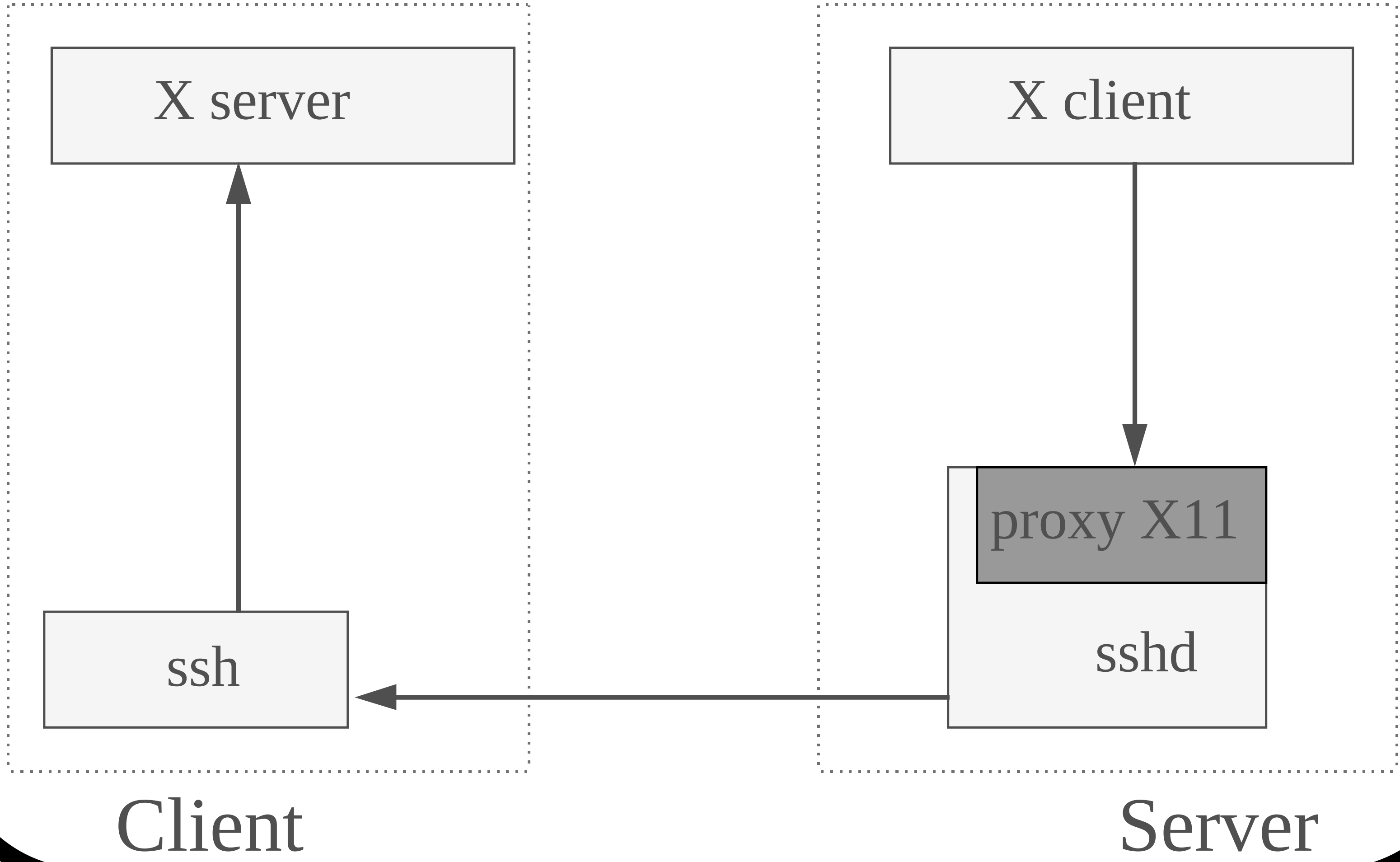
- request to execute shell or command

# Interactive Phase

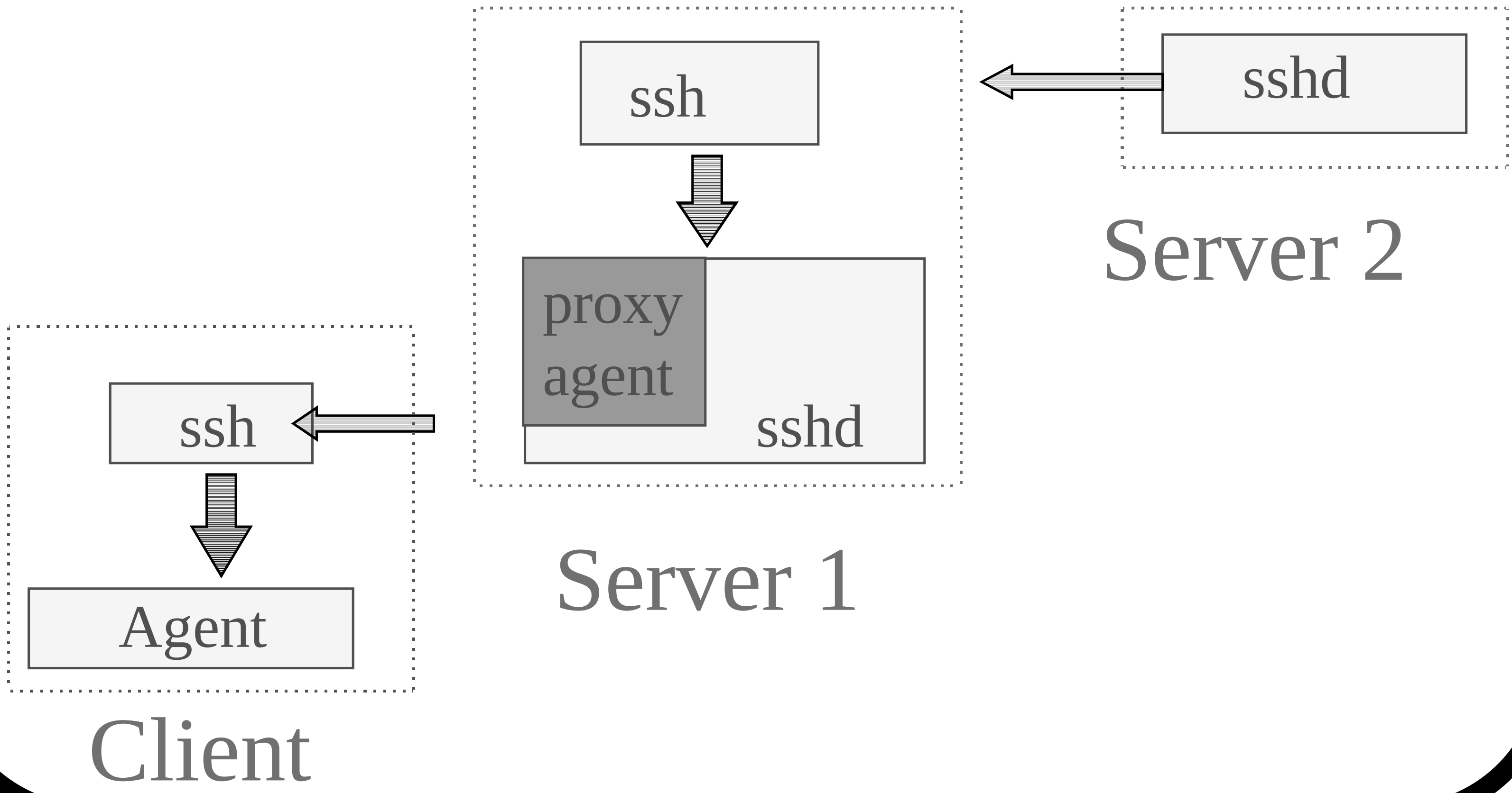
- packet-based: each site may send at any time

| Client sends | Server sends |
|---|---|
| stdin data | stdout data |
| eof | stderr data |
| | X11 open |
| TCP/IP open | agent open |
| channel data | TCP/IP open |
| channel close | channel data |
| window change | channel close |
| | exit status |

# X11 Forwarding



# Agent Forwarding



# Cryptography

- Host key 1024 bits RSA
  - » Host authentication, key exchange
- Server key 768 bits RSA
  - » Key exchange: provides forward secrecy after one hour
- Session key 256 bits (only part of it is currently used)

| | |
|---|---|
| IDEA-CFB | 128 bits |
| 3DES-CBC | 168 bits |
| RC4 | 128 bits (seperate keys for each direction) |
| DES-CBC | 56 bits (weak!) |

# Cryptography (cont´d)

- Data integrity
  - » Currently crc32 inside encryption
  - » In future will be changed to a keyed MAC outside encryption
- Strong pseudo-random number generator

# Performance

- Opening a connection takes about one second
- Typical transfer rate for i486/Pentium machines

| | |
|---|---|
| IDEA | 1-2 Mbits/sec |
| DES | 3-4 Mbits/sec |
| RC4 | 5 Mbits/sec |

- People have reported up to 40 Mbits/second software encryption speeds on faster machines
- Transfer rate is limited by ethernet speeds
- One does not need to justify why to encrypt - it costs very little. Instead, one must justify why not to encrypt, as that can have severe consequences

# Availability

- Unix
  - » Almost all variants, including: AIX, BSD, HPUX, IRIX, Linus, OSF/1, Solaris, SunOS, SysV, Unicos
  - » Distributed in source code
- Windows
  - » An “unofficial client” available from Kroatia
  - » Full Windows client: pre-alpha versions available now, beta around April -96
- OS/2: client available (from Switzerland)
- Macintosh and VMS versions are being prepared (in Sweden and in Slovenia, respectively)

# Installation on Unix

./configure
make
make install

Supports SOCKS, FWTK, tcp-wrappers

# Typical Usage (Unix)

ssh host
ssh host.domain.foo -l user
ssh host command
ssh -f hosts xterm

# How Widely Is It Currently Used?

- In at least 40 countries
- At several thousand sites
- Mailing list has 600 addresses from about 40 countries
- WWW pages accessed 1000-2000 times per day
- During about ten days, WWW pages accessed from 6000 different machines from 55 top-level domains (many of them WWW cache servers)

# SSH vs. IPSEC

- SSH is available now
- It will be years before IPSEC is widely deployed
- IPSEC with current transports may not be secure enough
- User-level authentication

# More Information

http://www.cs.hut.fi/ssh

info@ssh.fi

draft-ylonen-ssh-protocol-00.txt