P6 – Scientific Programming

Marcus Mohr
Jens Oeser

Geophysics Section
Department of Earth and Environmental Sciences
Ludwig-Maximilians-Universität München

SoSe 2021

Part #5

The Secure Shell

What is it?

Secure Shell, or SSH, is a cryptographic (encrypted) network protocol operating at layer 7 of the OSI Model to allow remote login and other network services to operate securely over an unsecured network.

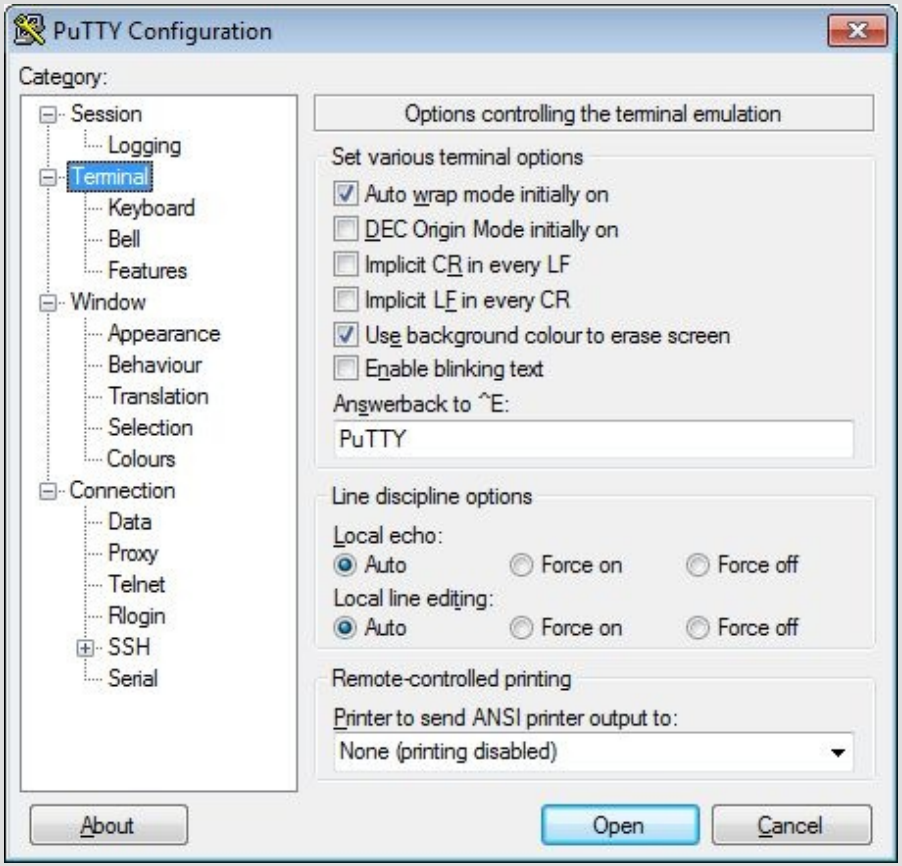
SSH provides a secure channel over an unsecured network in a client-server architecture, connecting an SSH client application with an SSH server. Common applications include remote command-line login and remote command execution, but any network service can be secured with SSH.

source: Wikipedia

SSH Clients

- Linux: OpenSSH
- Windows: PuTTY
- Mac OS: OpenSSH or PuTTY

well-known clients; others exist, too; especially for Mac; for PuTTY download we suggest: http://www.heise.de/download/putty.html



Remote Access

```
ssh [OPTIONS] HOSTNAME [COMMAND]
```

- without `[COMMAND]` used to log onto a remote machine
- `[OPTIONS]`
  - `-x` or `-X`
    - disable or enable X11 forwarding (high-speed connection advisable)
  - `-l USERNAME`
    - attempt to log onto remote machine using this `USERNAME`
    - alternative: `USERNAME@HOSTNAME`
- `HOSTNAME`
  - connect to specified remote machine (`FQDN`)
- `[COMMAND]`
  - instead of a login shell `[COMMAND]` is executed on remote host

```
gkd22aa@cip50:~$ ssh -X cip76
Last login: Fri Dec 11 14:19:43 2009 from cip50.cipmath.loc
gkd22aa@cip76:~$ xclock
[STRG-C]
gkd22aa@cip76:~$ ssh -x cip52 xeyes
Error: Can't open display:
gkd22aa@cip76:~$ ssh -l $USER cip30 'ps -f -u $LOGNAME'
UID        PID PPID C STIME TTY          TIME CMD
gkd22aa 5390 5387 0 15:37 ?    00:00:00 sshd: gkd22aa@notty
gkd22aa 5393 5390 0 15:37 ?    00:00:00 ps -f -u gkd22aa
gkd22aa@cip76:~$ ssh cip52
gkd22aa@cip52:~$ echo $HOSTNAME
cip52
gkd22aa@cip52:~$ exit
logout
Connection to cip52 closed.
```

File Transfer

```
sftp [USERNAME@]HOSTNAME[:DIRECTORY]
scp [USERNAME@][HOSTNAME:]SOURCE] \
    [USERNAME@][HOSTNAME:]DESTINATION
```

- secure copy (remote file copy program)
- `[USERNAME]`
  - specifies the user to log in as on the remote machine `USERNAME`
- `HOSTNAME`
  - connect to specified remote machine (`FQDN`)
- `[DIRECTORY]`
  - replaced with directory on the remote machine
- `[SOURCE] [DESTINATION]`
  - replaced with directory or filenames on the local or remote machine

```
gkd22aa@cip50:~$ sftp cip34
Connecting to cip34...
sftp> put ex-1.xyz ex1.dat
Uploading ex-1.xyz to /home/cip/gkd22aa/ex1.dat
ex-1.xyz                      100% 565 0.6KB/s 00:00
sftp> get ex-2.xyz ex2.dat
Fetching /home/cip/gkd22aa/ex-2.xyz to ex2.dat
/home/cip/gkd22aa/ex-2.xyz    100% 797 0.8KB/s 00:00
sftp> ls
Desktop ex-1.xyz ex-2.xyz ex1.dat ex2.dat copy.txt maildir ort_e
sftp> exit
gkd22aa@cip50:~$ scp $USER@cip76:~/ex-1.xyz Desktop/
ex-1.xyz  100% 565 0.6KB/s 00:00
gkd22aa@cip50:~$ scp copy.txt $USER@cip76:~/neu.dat
copy.txt 100%  90 0.1KB/s 00:00
```

Authentication Methods

- Access to a remote host requires that you **authenticate yourself** as the specified user on that host.
- The two most commonly used methods for this are:
  - **Password Authentication**
    - advantage of SSH: your password is send over an encrypted communication channel!
  - **Public Key Authentication**
    - also knows as Challenge Resonse Authentication
    - or RSA Challenge
    - requires a **key pair**

Key Pairs (1/5)

- a key pair consists of a public key and a private key
- is used for asymmetric encryption
  - encrypt with public key $\longrightarrow$ decrypt with private key
  - encrypt with private key $\longrightarrow$ decrypt with public key
- public key can be known to everybody (no harm done)
- private key must be kept secret!

Key Pairs (2/5)

- On Linux (OpenSSH) key pairs are generated with the `ssh-keygen` command.
- Keys will be placed in `$HOME/.ssh` sub-directory.
- Default names are:
  - `id_rsa` for the private key
  - `id_rsa.pub` for the public key
- For enhanced security your private key is stored encrypted $\longleftrightarrow$ passphrase.
- Your public key is stored as plain text.

```
gks2aa@cip50:~$ ssh-keygen -b 4096 -C mohr@cippool
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cip/gks2aa/.ssh/id_rsa): <return>
Created directory '/home/cip/gks2aa/.ssh'.
Enter passphrase (empty for no passphrase): <type passphrase>
Enter same passphrase again:  <repeat passphrase>
Your identification has been saved in /home/cip/gks2aa/.ssh/id_rsa.
Your public key has been saved in /home/cip/gks2aa/.ssh/id_rsa.pub.
The key fingerprint is:
3b:19:d6:41:64:47:30:77:ca:68:04:0e:2a:ec:e5:20 mohr@cipool
The key's randomart image is:
+--[ RSA 4096]----+
|      . .oBo+ .  |
| .   . o + * o   |
|E + o   . + o    |
| o =     o .     |
|  . .   S .      |
|       . +       |
|        +        |
|         .       |
|                 |
+-----------------+
```

Key Pairs (4/5)

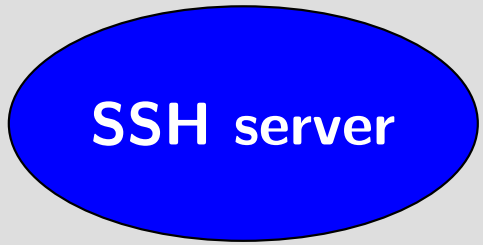
```
gks2aa@cip50:~$ ls -lF .ssh
insgesamt 8
-rw------- 1 gks2aa gks2 1766 Apr 20 10:48 id_rsa
-rw------- 1 gks2aa gks2  394 Apr 20 10:48 id_rsa.pub

gks2aa@cip50:~$ file .ssh/id_rsa
.ssh/id_rsa: PEM RSA private key

gks2aa@cip50:~$ file .ssh/id_rsa.pub
.ssh/id_rsa.pub: OpenSSH RSA public key
```

Key Pairs (5/5)

- Examining the public key we see three components:
  - key type
  - key itself (base64 encoded)
  - comment / symbolic key name

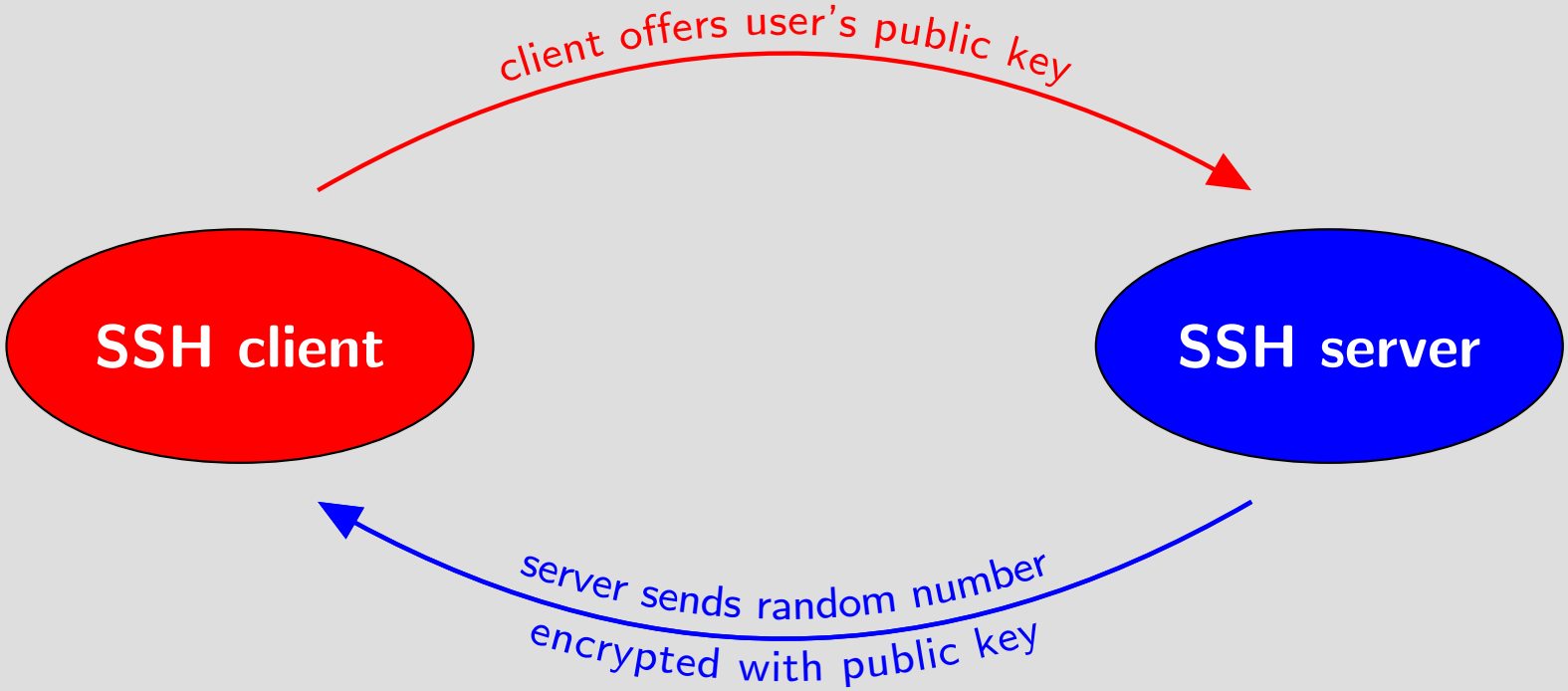
```
gks2aa@cip50:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDx33CDE1fcp2b+X8R+KgDE0pSXjsTpj4LhRRA
t1qJgw7oQX9rC3l7waBotWOzDBtuzE90pO63C8nDGySYW3/07lsbKHwxeh5kT2I1gIJpJ3olWBl
Ez6owRtaszv8R7hPYFm5xC609n6OyCcgFLN/M0qWyjTtTUDbpbYpzgjqOsdde9l2SQ5iMMY7JGb
rYcc5uQMQqrGa5X0Q9yNlb+9nbW/0Lwvfe+ZYVJwlT4r1Sd1qGtriswwYIxb6wrhG2xwFUbfLzA
28p9y7+jvy1MqVjyM8V3l+UyZuEGTi3eXuYVAQdwJWGNkbTBVlXOV6KPxKYyAAHe9UeI/bZOZ/n
yWAeoEuhGWBIwL1ZLEWNjLWYsaLAFCAOBoxxOJXoy3BwLMpVsWl+vqBvn+qMxXVqUTOBaR7Guxk
soISiOEy+dY25Lf7389LaenuOEP+QpjLMp9nYmEIpfqjcfF7O6en2i99MchVAJtunz5h40OUyFR
HRJGfy4DA/hVwwuqKTBhi6vlkyBxylUOd89uvgf3u4t0nnbQQvhqgxqtkyWhM8dYi2rDHRLdIAL
eYOZegPuBgjDgIOPFQlOhW3axj7O8adSdWpZP31lEZAvG83oFCnGp9GqynL8us+3m3VTOGgv/H3
uF16RkDruUA3jRn/Za7JKIos6l3k41pTJJfTLXS5zZkfQEw== mohr@cipool
```

Public Key Authentication (1/2)

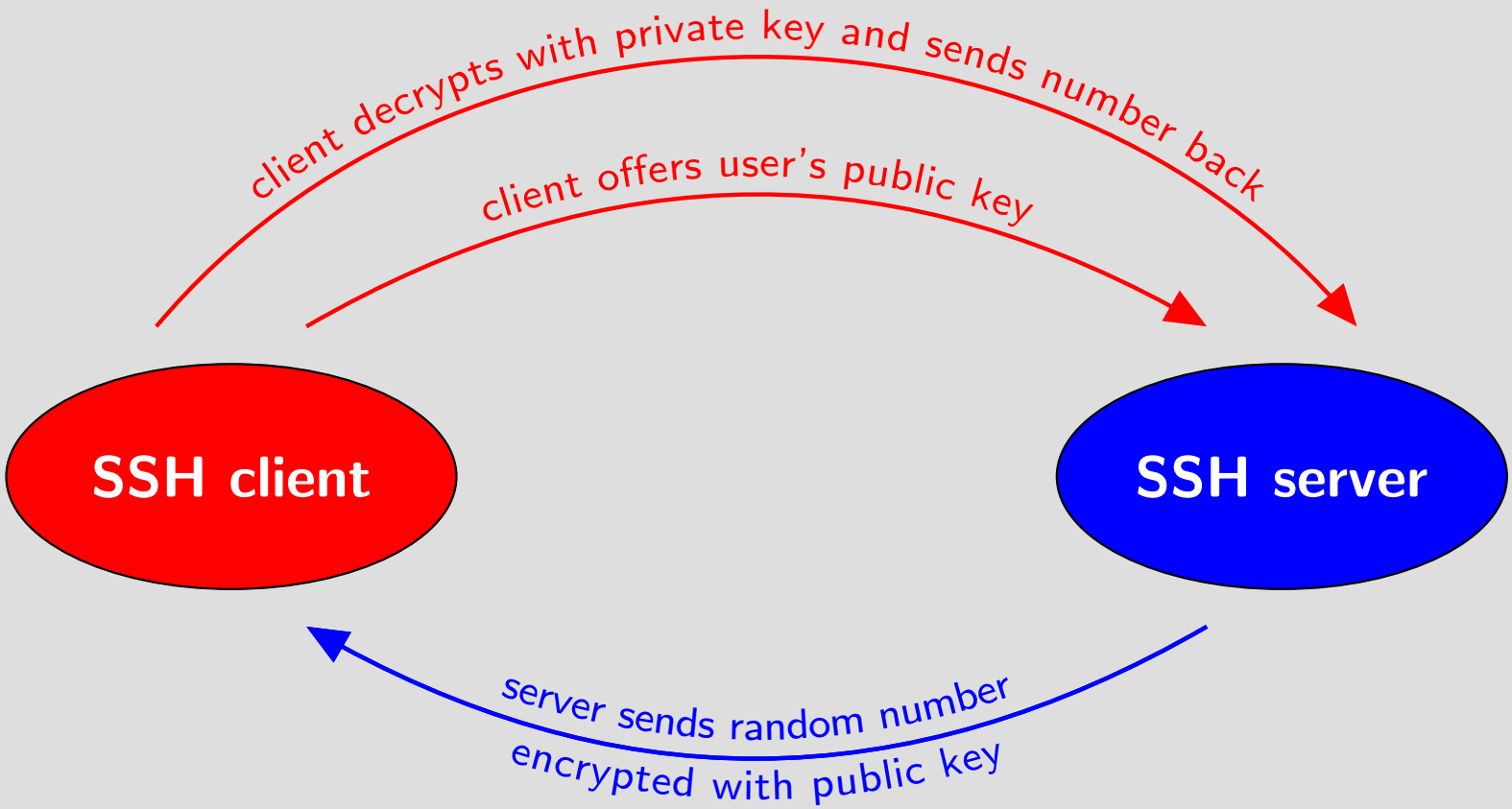


Public Key Authentication (1/2)

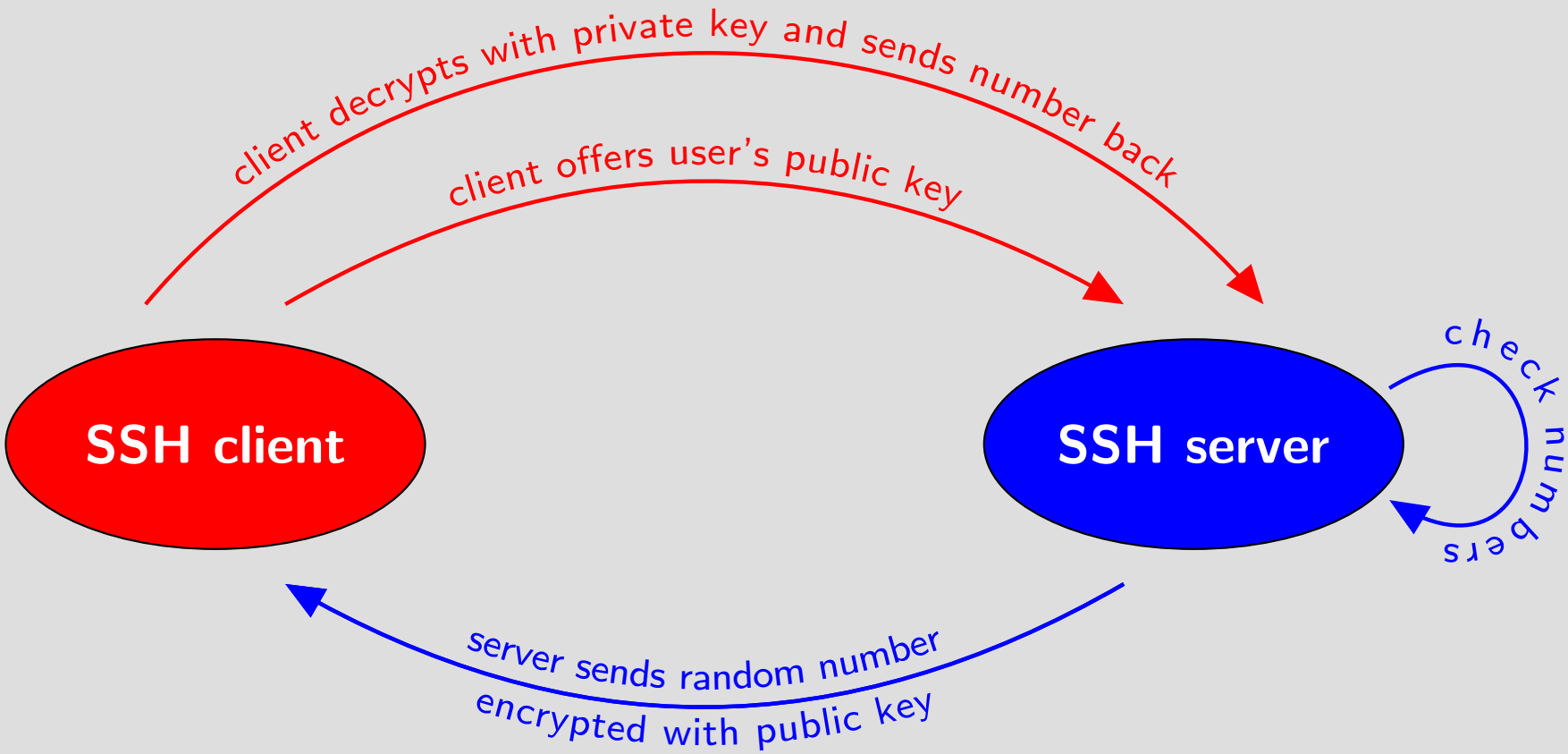
Public Key Authentication (1/2)



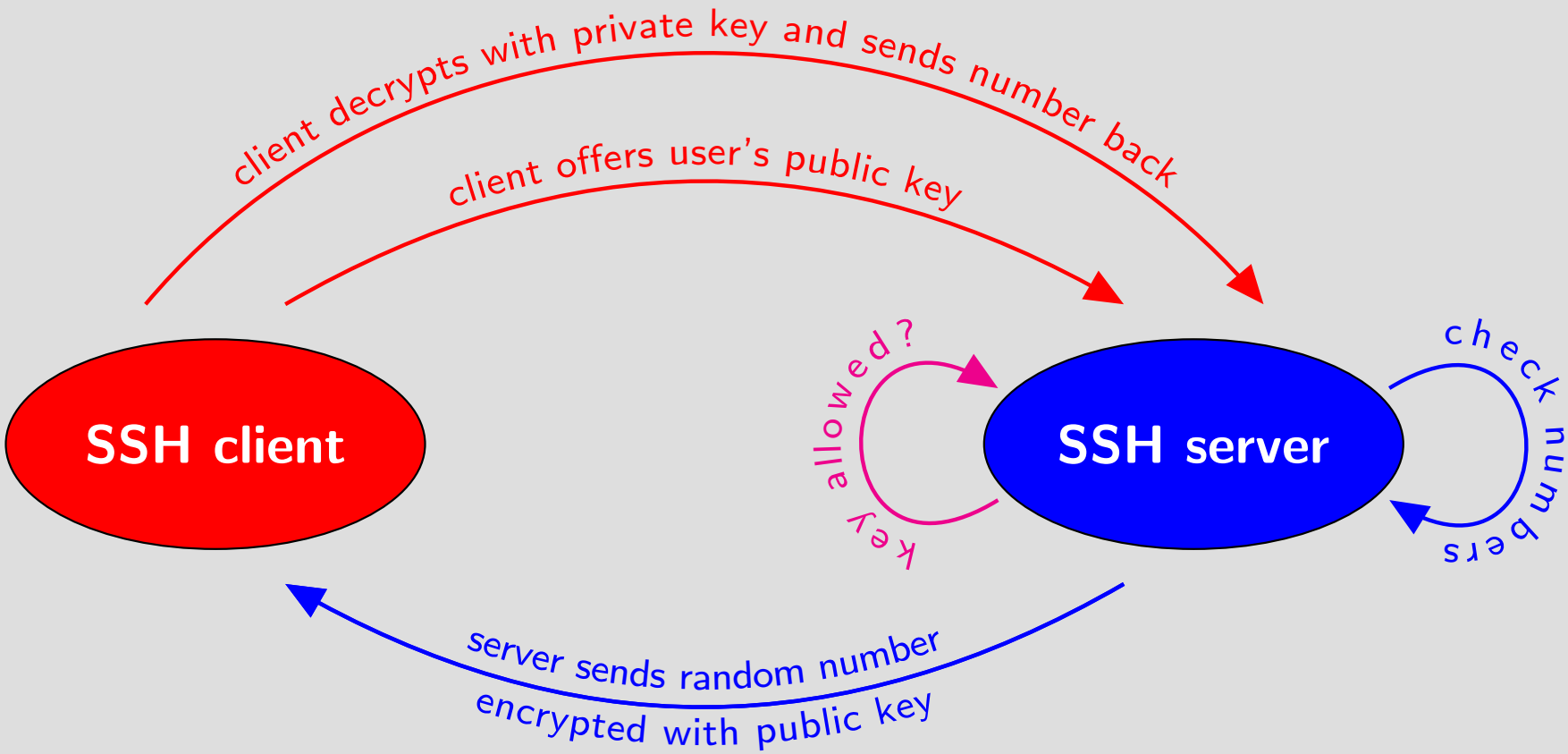
Public Key Authentication (1/2)



Public Key Authentication (1/2)



Public Key Authentication (1/2)



Public Key Authentication (2/2)

Server will only allow public keys for the RSA challenge that are listed in the user's `authorized_keys` file!

- Put your public key to be used by client into `authorized_keys` file on server.
- Config below allows user mohr from Geophysics to log onto cip50 as user `gks2aa`.

```
gks2aa@cip50:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAxZNhXsWTlPlxVAkSLedpInqVC9dMjhUY2PrVg
T/EN6IRU2zPRbcGeD/hcwpg84ef2LPEpZ5kOUY4+pA/3Kvngvm6OWoyi/FlOYAFftr4mu0YY5
oS8kbzBhHAe7Y3J50F7ZY+uAKNXIW4qeG/2sJYlyviqCff/vKVdMUZ8Ke3+BM= mohr@one
```

Known Hosts

- client and server also use key-pairs
  - for encrypting communication
  - and to establish their identities
- if you contact a server whose public key is not known (not in system-wide config file)
  - you will be asked whether you trust that server
  - public key will be added to your personal `.ssh/known_hosts` file