SSH Configuration - Interm. Level

SSH Configuration
Intermediate Level

#MCH2022

@LEYRER
https://martin.leyrer.priv.at

Command Line Syntax

If a <newline> follows the <backslash>, the shell shall interpret this as line continuation.

The Open Group Base Specifications Issue 7, 2018 edition; IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008)

\ Example

Regular

```
$ ls -a -l -t /tmp
```

With Backslash

```
$ ls \
     -a \
     -l \
     -t \
     /tmp
```

SSH Configuration - Interm. Level

ssh leyrer@ssh.example.com

ssh leyrer@hziulquoigmnzhah.com

ssh_config(5)

1. command-line options
2. user's configuration file (~/.ssh/config)
3. system-wide configuration file (/etc/ssh/ssh_config)

~/.ssh/config

Hostname Aliases

```
Host demo
    HostName ssh-server.example.com
```

Hostname Aliases

```
Host demo d1 popocatepetl
    HostName ssh-server.example.com
```

Hostname Aliases

```
Host smtp imap www
    HostName %h.example.com
```

No Usernames

```
Host demo bastion
    HostName ssh-server.example.com
    User leyrer
```

ssh leyrer@ssh.example.com

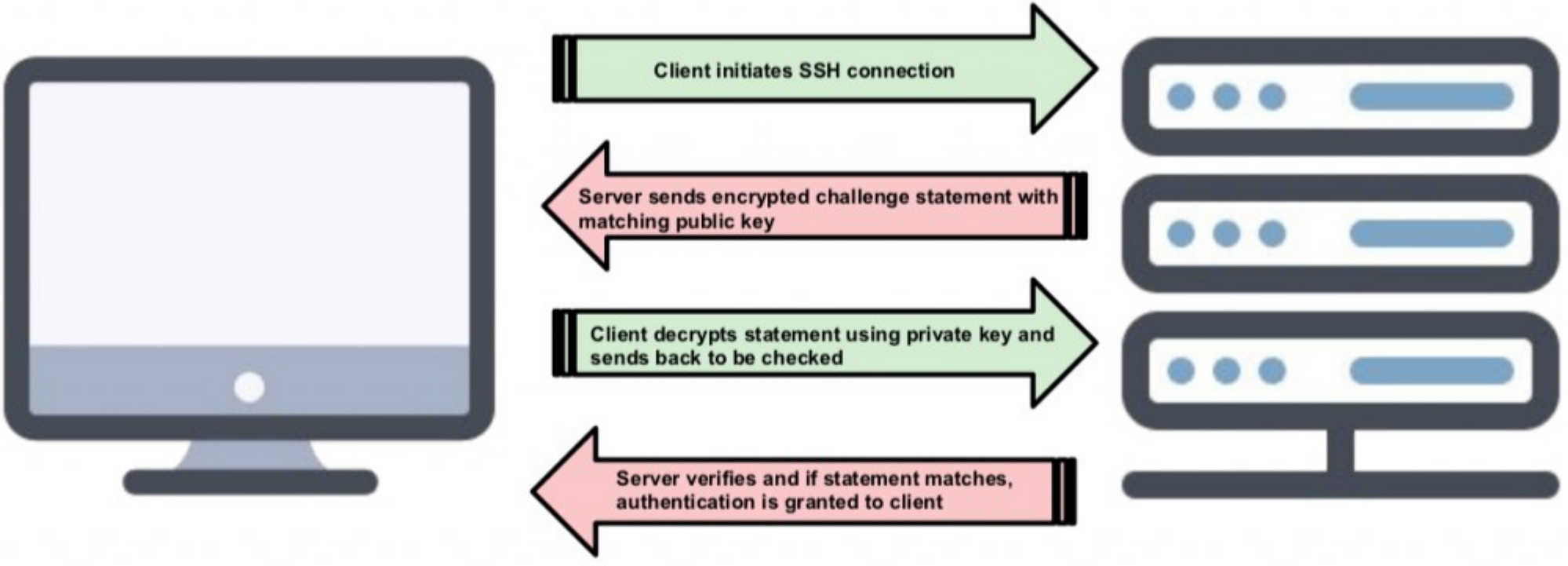
ssh demo

No Passwords

SSH Public Key authentication

Asymmetric Cryptography

- **public** key
  - on the SSH server(s)
- **private** key
  - on the PC/Laptop


Client initiates SSH connection
Server sends encrypted challenge statement with matching public key
Client decrypts statement using private key and sends back to be checked
Server verifies and if statement matches, authentication is granted to client

ssh-keygen

```
$ ssh-keygen -t ed25519 \
    -a 420 \
    -f ~/.ssh/demo.ed25519 \
    -C "leyrer’s key (mch2022)"
```

ssh-keygen output

```
Generating public/private ed25519 key pair.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/leyrer/.ssh/demo.ed25519

Your public key has been saved in /home/leyrer/.ssh/demo.ed25519.pub

The key fingerprint is:

SHA256:x9IvXWUWA6fNcyY+AqkDwpdCf2LPOHzLEzakFlOYQlY leyrer’s key
(mch2022)
```

ssh-keygen output

```
Generating public/private ed25519 key pair.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:
```

ssh-keygen result

```
/home/leyrer/.ssh/demo.ed25519

/home/leyrer/.ssh/demo.ed25519.pub
```

demo.ed25519.pub

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKoBO3NrfLS8Z
qWXWv9Uwf12B7QiaiMEs6N2F9f7PNln Dem
leyrer sein key (gpn20)
```

demo.ed25519

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1r                                                    AAABCErro6Gn
+4QcKJBPVP1m                                                    O3NrfLS8ZqWX
Wv9Uwf12B7Qi                                                   eq7pLoT4psCZ
BwBvXSR/KA3w                                                    c97ONBmMTjCA
1+Z6Ff8xKMb/                                                   oAiY5l9qEJAx
YLZnPcINuRjO
-----END OPENSSH PRIVATE KEY-----
```

Asymmetric Cryptography

- **public** key
  - on the SSH server(s)
- **private** key
  - on the PC/Laptop

Public/Private Key

- They are free !!!
- Create one for each server, customer or service you connect to.

ssh-copy-id

```
ssh-copy-id \
    -i ~/.ssh/demo.ed25519.pub
```

Manual Copy

- copy content of local `~/.ssh/demo.ed25519.pub`
- paste it into `~/.ssh/authorized_keys` on servers

Connect

```
ssh \
    -i ~/.ssh/demo.ed25519 \
    demo
```

Private Key - Less Typing

```
Host demo bastion
    PreferredAuthentications publickey
    IdentityFile ~/.ssh/demo.ed25519
```

~/.ssh/config

```
Host demo bastion
    HostName ssh.example.com
    User leyrer
    PreferredAuthentications publickey
    IdentityFile ~/.ssh/demo.ed25519
```

Connect

`ssh demo`

Connect

```
ssh demo

Enter passphrase for key
'/home/leyrer/.ssh/demo.ed25519':
```

ssh-agent(1)

- stores unencrypted keys in memory
- communicates via sockets
- GNOME Keyring, Kwallet, ... include ssh-agent functionality

ssh-add(1)

- adds private key identities to ssh-agent
- “-c” ask for permission before use
- “-d” removes key from ssh-agent

Connect without passphrase

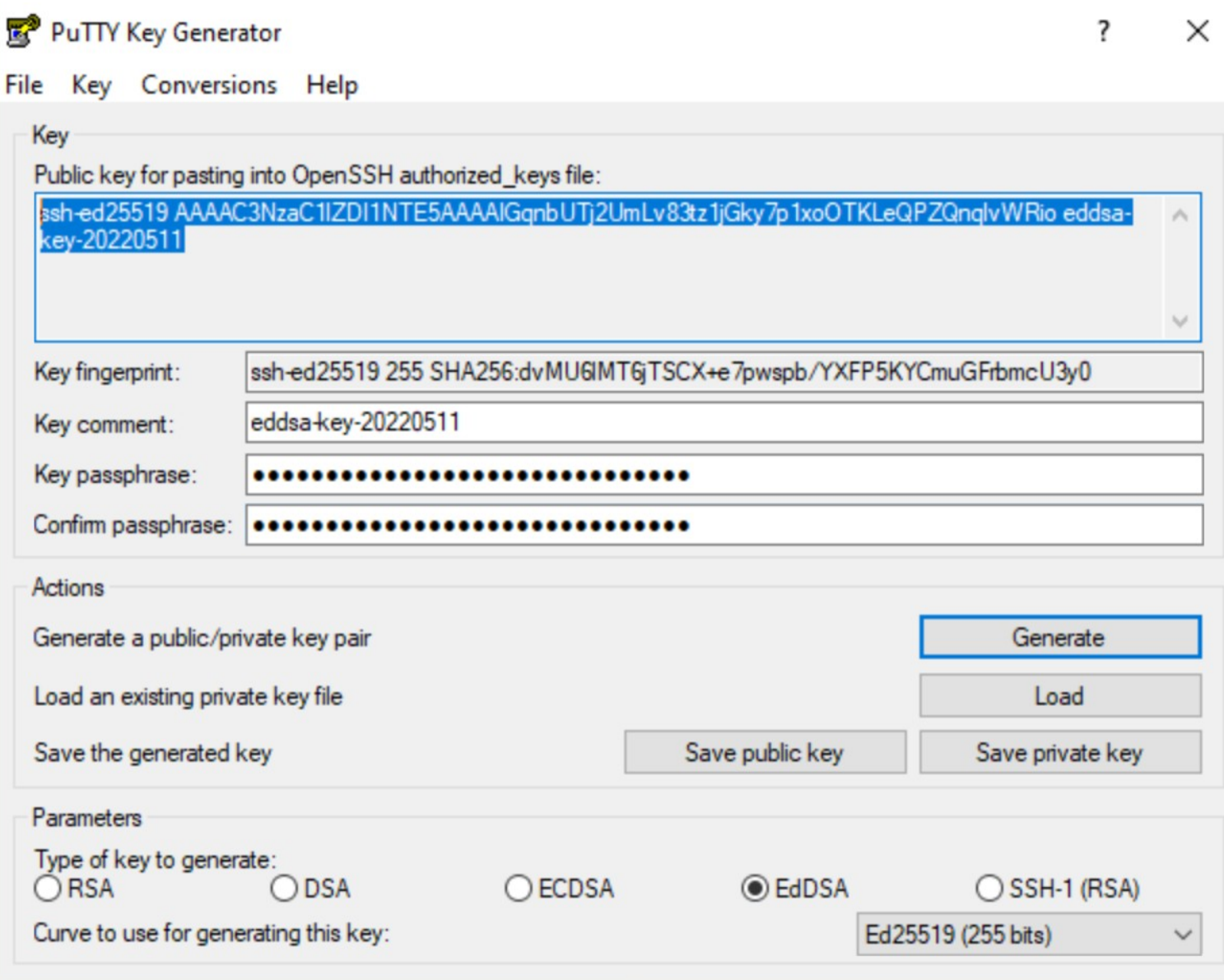
```
ssh-add ~/.ssh/demo.ed25519

ssh demo
```

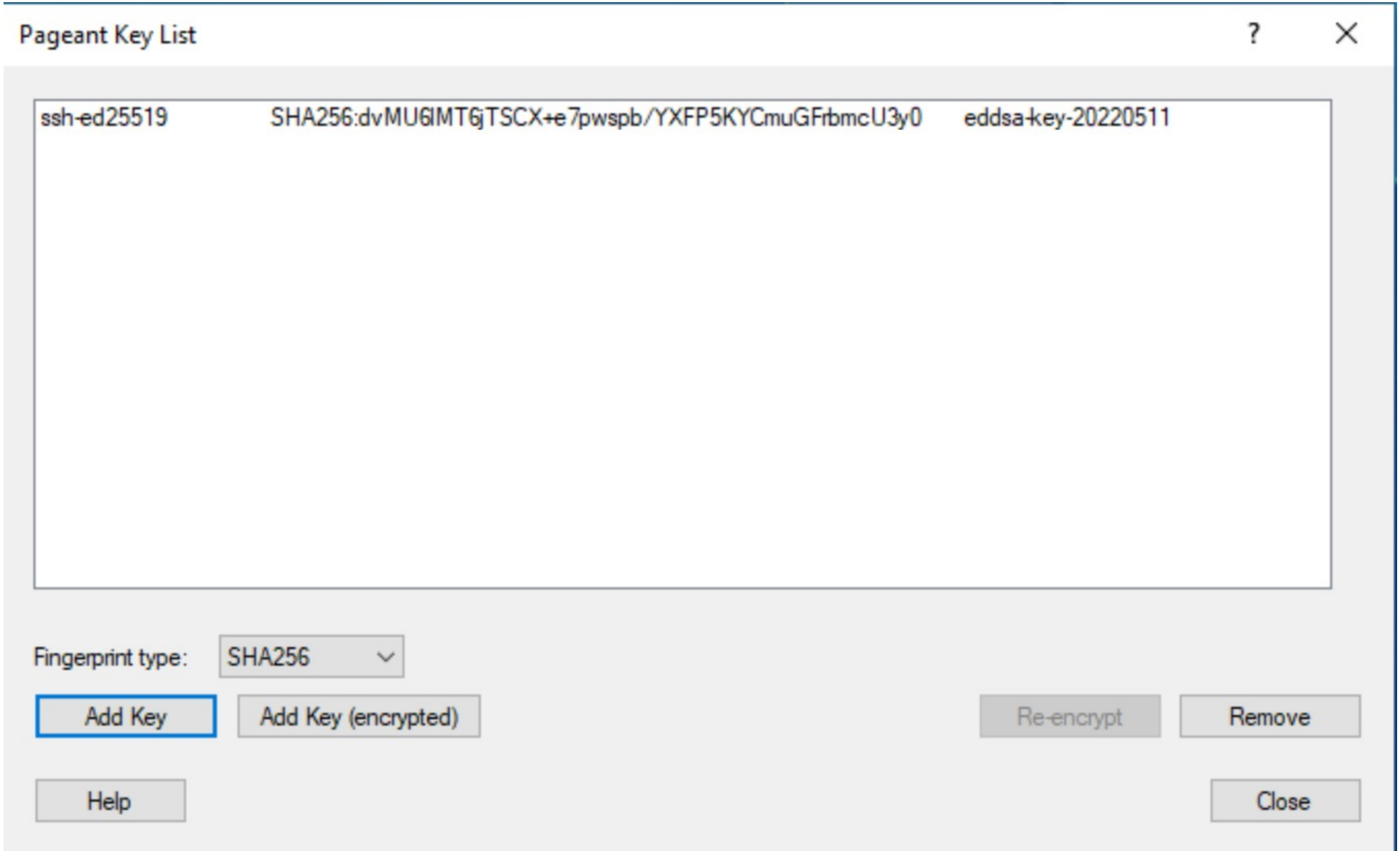
Passwordless Benefit

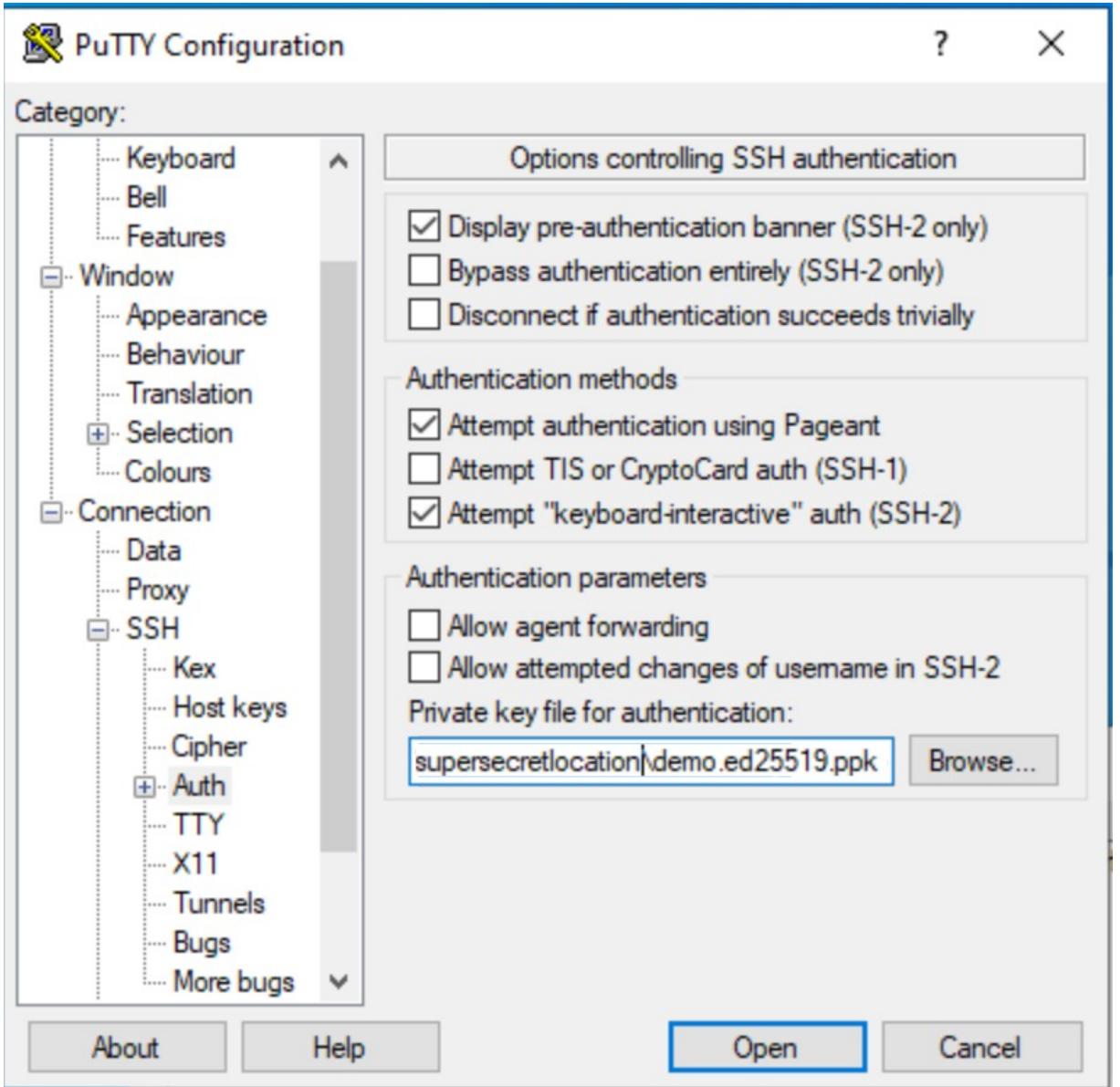
Remote auto completion !!!

Keys & Agent @ Windows

- puttygen
- pagent
- putty


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGqnbUTj2UmLv83tz1jGky7p1xoOTKLeQPZQnqlvWRio eddsa-key-20220511
Key fingerprint: ssh-ed25519 255 SHA256:dvMU6lMT6jTSCX+e7pwspb/YXFP5KYCmuGFrbmcU3y0
Key comment: eddsa-key-20220511
Key passphrase:
Confirm passphrase:
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA DSA ECDSA EdDSA SSH-1 (RSA)
Curve to use for generating this key:
Ed25519 (255 bits)


Pageant Key List
ssh-ed25519 SHA256:dvMU6lMT6jTSCX+e7pwspb/YXFP5KYCmuGFrbmcU3y0 eddsa-key-20220511
Fingerprint type:
SHA256
Add Key
Add Key (encrypted)
Re-encrypt
Remove
Help
Close


PuTTY Configuration
Category:
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Bugs
More bugs
Options controlling SSH authentication
Display pre-authentication banner (SSH-2 only)
Bypass authentication entirely (SSH-2 only)
Disconnect if authentication succeeds trivially
Authentication methods
Attempt authentication using Pageant
Attempt TIS or CryptoCard auth (SSH-1)
Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
Allow agent forwarding
Allow attempted changes of username in SSH-2
Private key file for authentication:
supersecretlocation\demo.ed25519.ppk
Browse...
About
Help
Open
Cancel

~/.ssh/config Magic

- For each parameter, the first obtained value will be used.
- more host-specific declarations should be given near the beginning of the file
- general defaults at the end

Per Domain

```
Host *.test.example.com
    IdentityFile ~/.ssh/test-id.ed25519

Host unicorn.prod.example.com
    IdentityFile ~/.ssh/pinkfluffy-id.ed25519

Host *.prod.example.com
    IdentityFile ~/.ssh/prod-id.ed25519
```

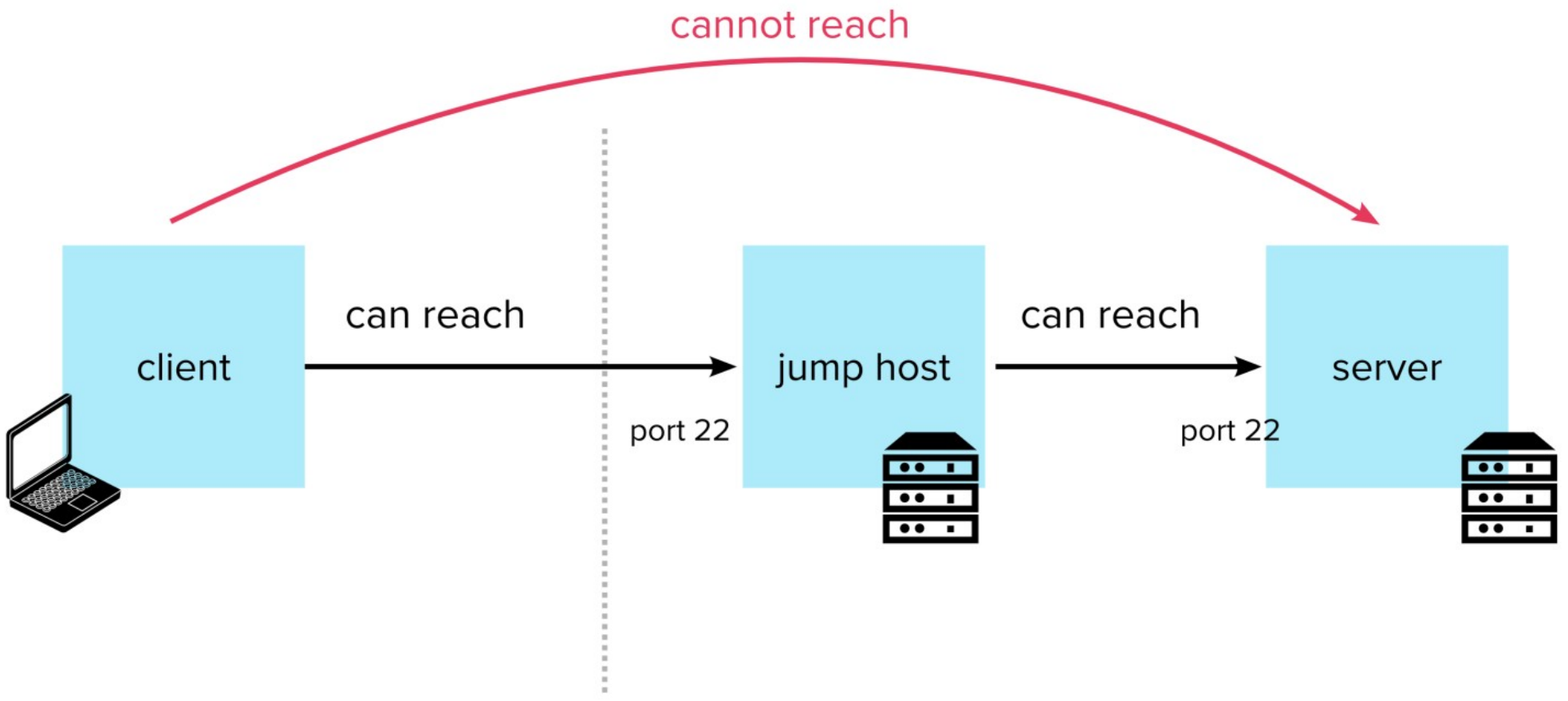
Sane Defaults

```
Host demo bastion
    HostName ssh.example.com
    User leyrer
    PreferredAuthentications publickey
    IdentityFile ~/.ssh/demo.ed25519
…
Host *
    IdentitiesOnly yes
    UseRoaming no
    SendEnv LANG LC_*
    Compression yes
```

~/.ssh/conf.d/

- multiple files (eg. one per customer/service)
- same syntax as ssh_config(5)
- sorted by filename

Bastion/Jump Hosts


cannot reach
client
can reach
port 22
jump host
can reach
port 22
server

Manual “Jumping”

```
$ ssh bastion
Welcome to demo.example.com
leyrer@demo:~$ ssh target.local

Welcome to target.local
leyrer@target:~$
```

Explicit “Jumping”

```
$ ssh -J bastion target.local
```

Elegant “Jumping”

```
Host demo bastion
    HostName ssh.example.com
    User leyrer
    PreferredAuthentications publickey
    IdentityFile ~/.ssh/demo.ed25519

Host internal
    HostName target.local
    ProxyJump bastion
    User leyrer
    PreferredAuthentications publickey
    IdentityFile ~/.ssh/demo.ed25519
```

Jump

```
$ ssh internal
```

Jump Fallback

```
ssh \
    -o ProxyCommand="ssh -W %h:%p bastion" \
     target.local


.ssh/config:
    Host internal
        ProxyCommand ssh -W %h:%p bastion
```

Agent Forwarding

```
ssh -A \
    -o ProxyCommand="ssh -W %h:%p bastion" \
     target.local


.ssh/config:
    Host *
        ForwardAgent yes
```

ssh-audit

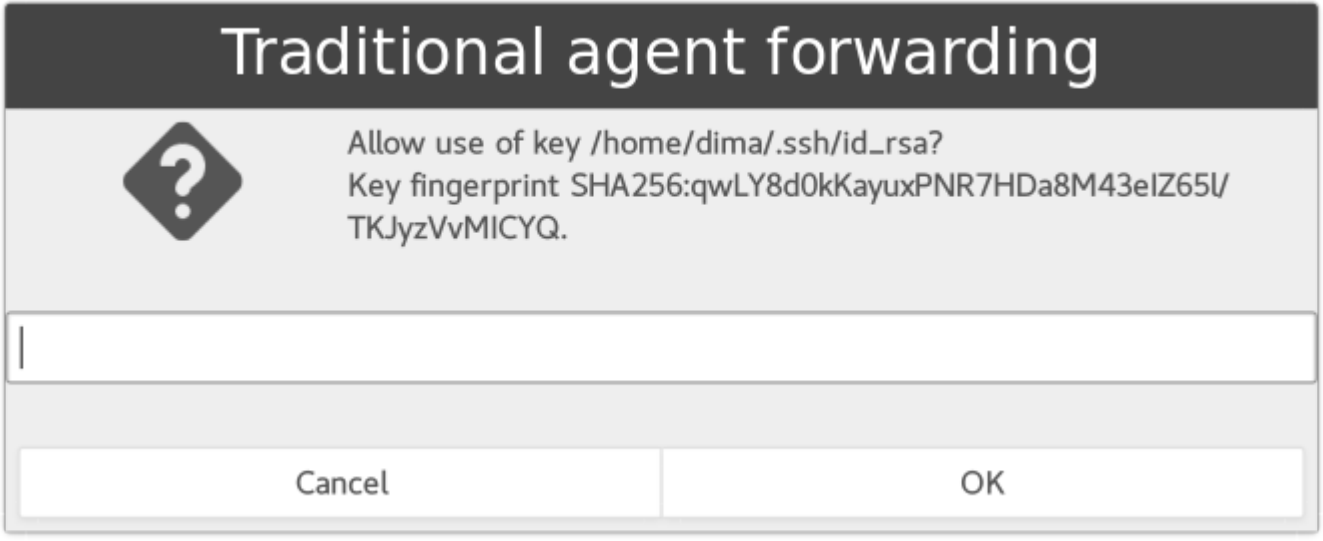
- output algorithm information
- output algorithm recommendations
- output security information
- analyze SSH version compatibility

https://github.com/arthepsy/ssh-audit/

guardian-agent

- more-constrained agent forwarding
- can safely be enabled on any connection
- can be used alongside Mosh or SSH

https://github.com/arthepsy/ssh-audit/


Traditional agent forwarding
Allow use of key /home/dima/.ssh/id_rsa?
Key fingerprint SHA256:qwLY8d0kKayuxPNR7HDa8M43eIZ65l/
TKJyzVvMICYQ.
Cancel
OK


Guardian Agent
Allow aws to run 'git-upload-pack dimakogan/private' on git@gitlab.com:22?
1) Disallow
2) Allow once
3) Allow forever
4) Allow aws to run any command on git@gitlab.com:22 forever
Answer (enter a number):
Cancel
OK

May contain Answers

Martin Leyrer

https://martin.leyrer.priv.at

@leyrer
@leyrer@chaos.social