Gettin’ Fancy with SSH Keys

Ed Cashin

CHUGALUG
January 2001

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

Why

We would like to safely . . .

- run commands on remote machines
- transfer files over the network
- single sign-on

. . . with everything encrypted.

Why

Secure Shell

Keys

Single Sign-on

Tying Keys . . .

Applications

Close

Secure Shell

SSH is the secure shell.

- encryption
- replaces rsh, rcp
- uses advanced cryptography
  - several algorithms
  - aware of man-in-the-middle, etc.
  - See “RFC” in distro.

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

Keys

loose definition: *a sequence of bytes for use with a cryptographic algorithm*

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

types of keys

- host key

  generated at time of ssh installation

- session keys

  generated automatically each time you use ssh

- identity keys

  user level; generated with *ssh-keygen* utility

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

Single Sign-on

Why

Secure Shell

Keys

Single Sign-on

Tying Keys . . .

Applications

Close

set up keys

procedure:

- create key pair, “foo,” on localhost
- append *foo.pub* contents to remotehost’s *authorized_keys* file
- edit entry in remotehost’s *authorized_keys* file if needed

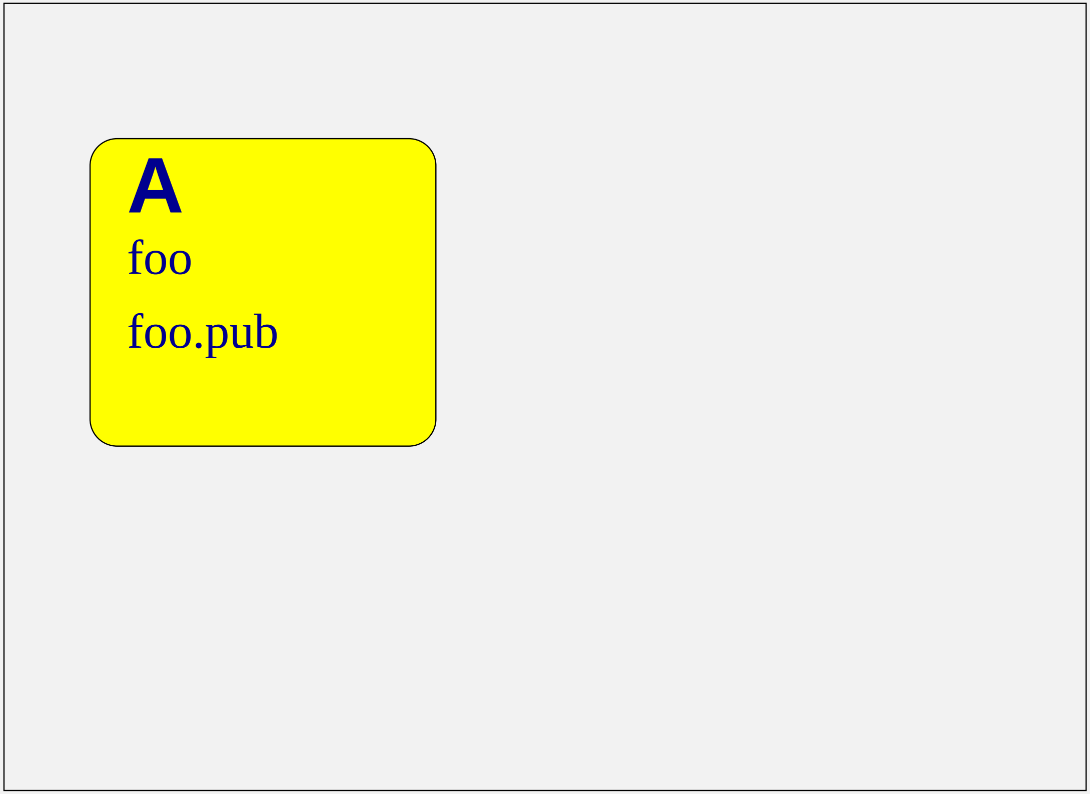
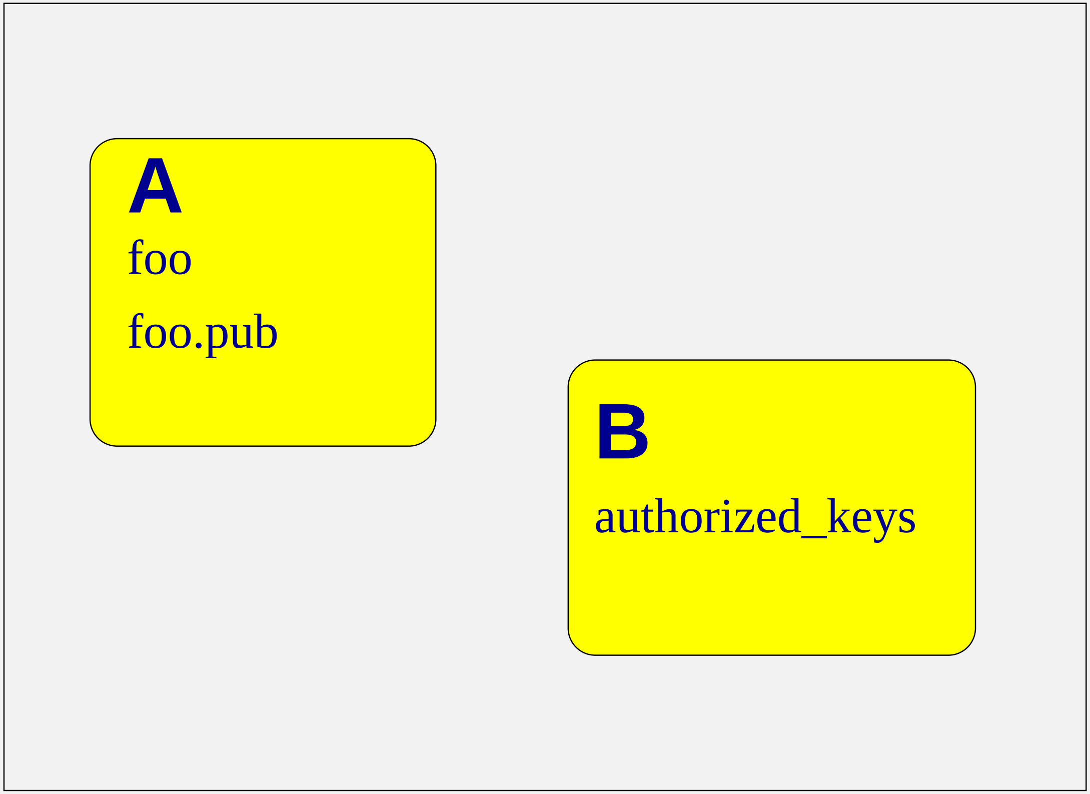


Figure 4.1 key pair: generate

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close



A

foo

foo.pub

B

authorized_keys

Figure 4.2 key pair: install

Why

Secure Shell

Keys

Single Sign-on

Tying Keys . . .

Applications

Close

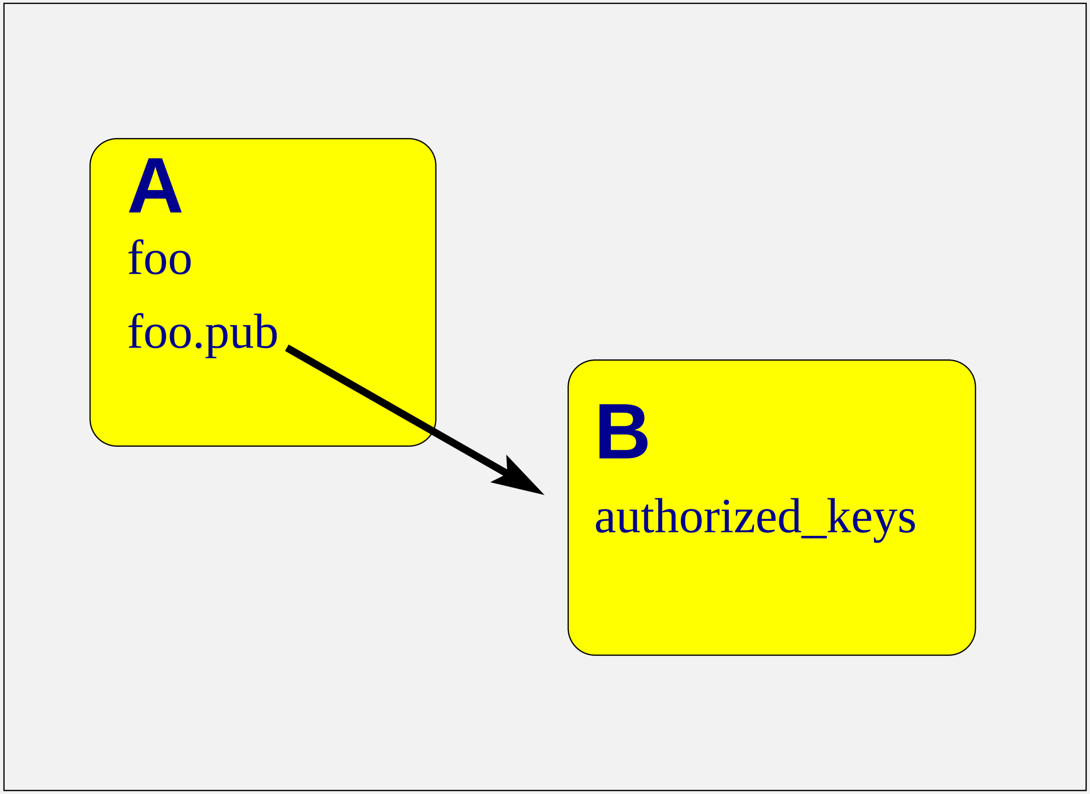


Figure 4.3 key pair: install

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

remote host, “B,” will recognize an identity key if it’s in its *authorized_keys* file.

You can have as many keys as you want.

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

ssh-agent

gives multiple processes access to identity keys

- uses sockets
- uses the environment to point to the sockets
- *ssh-add* utility hands the keys to the *ssh-agent*

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

```
# .xinitrc

exec ssh-agent icewm

# or ...
# exec ssh-agent sh -c \
  ’ssh-add < /dev/null && exec wmaker’
```

- easy, but dangerous
- xlock, xscreensaver help

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

Tying Keys to Commands

specify command in *authorized_keys* file on remote host

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

examples

- echo ’Hello, World!’
- mt -f /dev/nst0 rewind
- nice -n 19 dd of=/dev/nst0
- nice -n 19 dd if=/dev/nst0

Why

Secure Shell

Keys

Single Sign-on

Tying Keys . . .

Applications

Close

restrictions for safety

- limit key to one command
- limit key to one host
- no *pty*, etc.

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

Applications

- unattended remote secure backup
- secure remote logging
- sync data across machines
- etc.

Why
Secure Shell
Keys
Single Sign-on
Tying Keys . . .
Applications
Close

fin