

LOOSE YOUR KEYS

Uncovering SSH Certificates

PRESENTED BY
Linda Ikechukwu

SSH KEYS MAY NOT BE
SO BAD, BUT

SSH KEYS MAY NOT BE SO BAD, BUT

They can get lost, stolen, or shared

SSH KEYS MAY NOT BE SO BAD, BUT

They can get lost, stolen, or shared

Just ask ShapeShift, Godaddy, Capital One, and the rest

Me:



Linda Ikechukwu

Developer Advocate



Smallstep Labs

What we will cover:

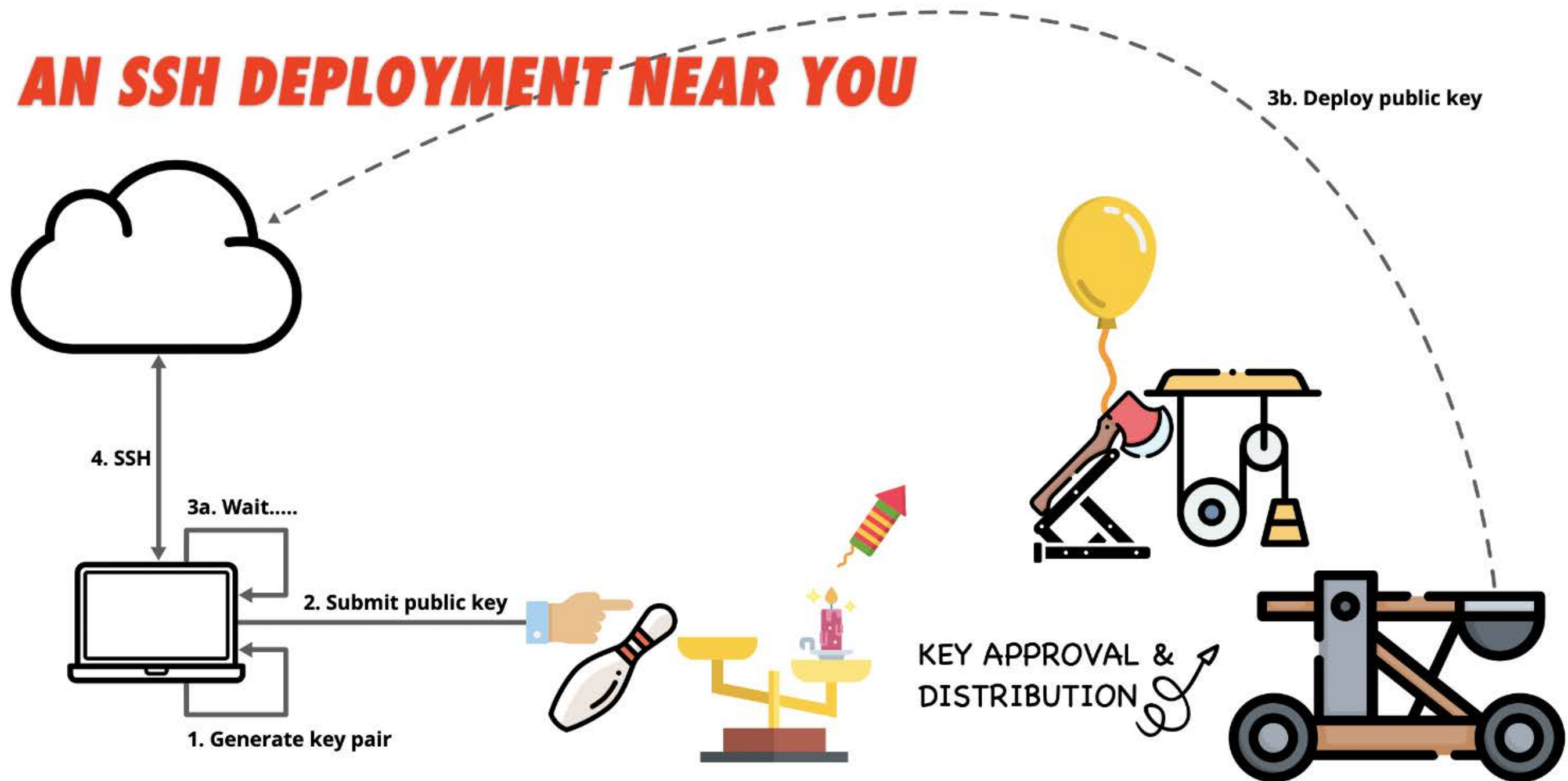
- 1 Reasons why SSH keys are bad for you**
- 2 What SSH certificates are, and how they work.**
- 3 Why SSH certificates are great for you.**
- 4 How you can start using SSH certificates**
- 5 Links to more learning resources**

WHY SSH KEYS ARE BAD FOR YOU

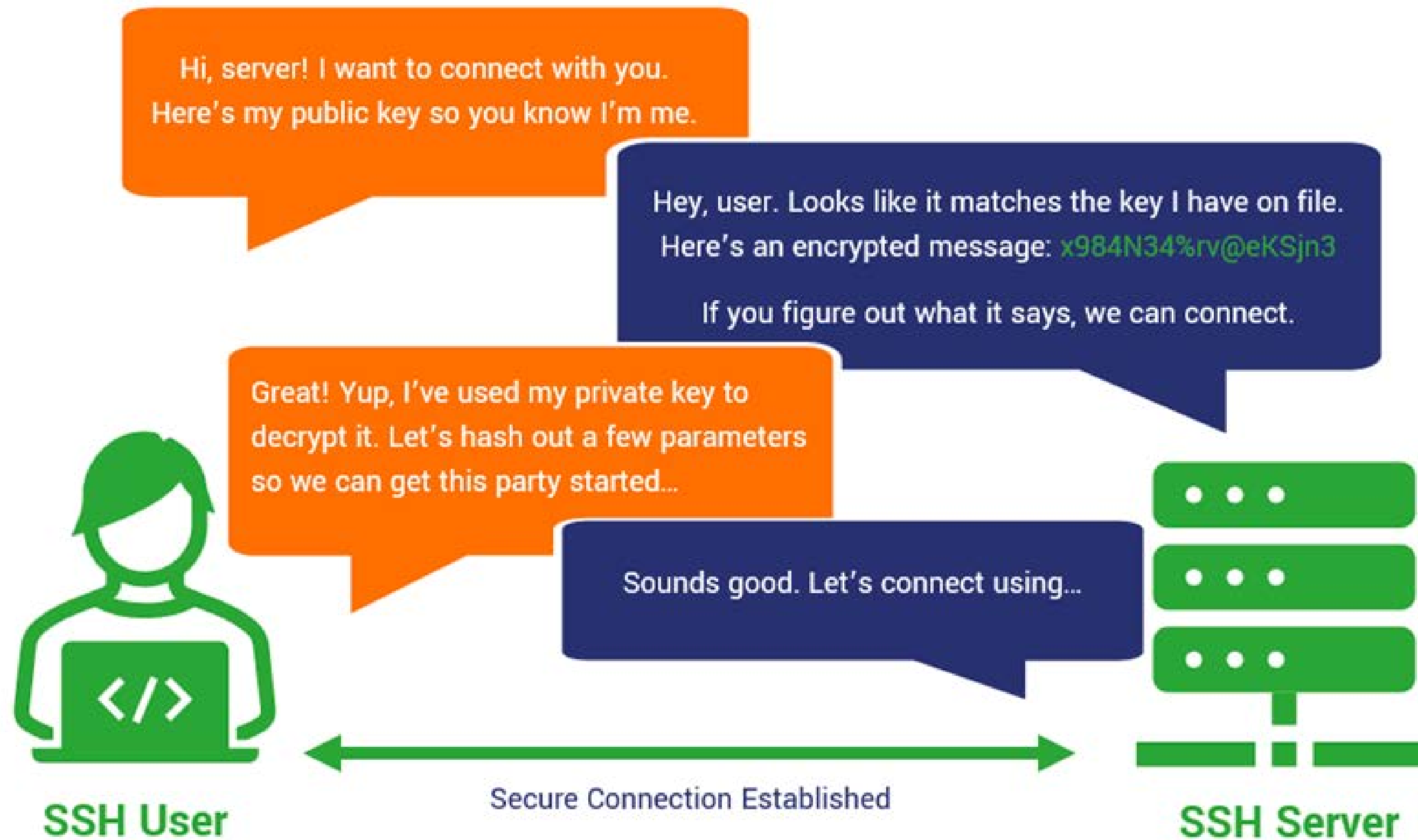
In a nutshell, they're just not scalable



SSH Key Onboarding:



How SSH Authentication Works





1

SSH key authentication requires laborious key management activities

“Many organisations don’t even know how many SSH keys they have configured to grant access to their infrastructure or who has copies of those keys”

-NISTIR 7966



1

SSH key authentication requires laborious key management activities

“Many organisations don’t even know how many SSH keys they have configured to grant access to their infrastructure or who has copies of those keys”

-NISTIR 7966

- Do you have an up to date inventory of all enabled SSH keys in your organization?
- Do you know which SSH keys belong to which users?



2

SSH Keys do not expire

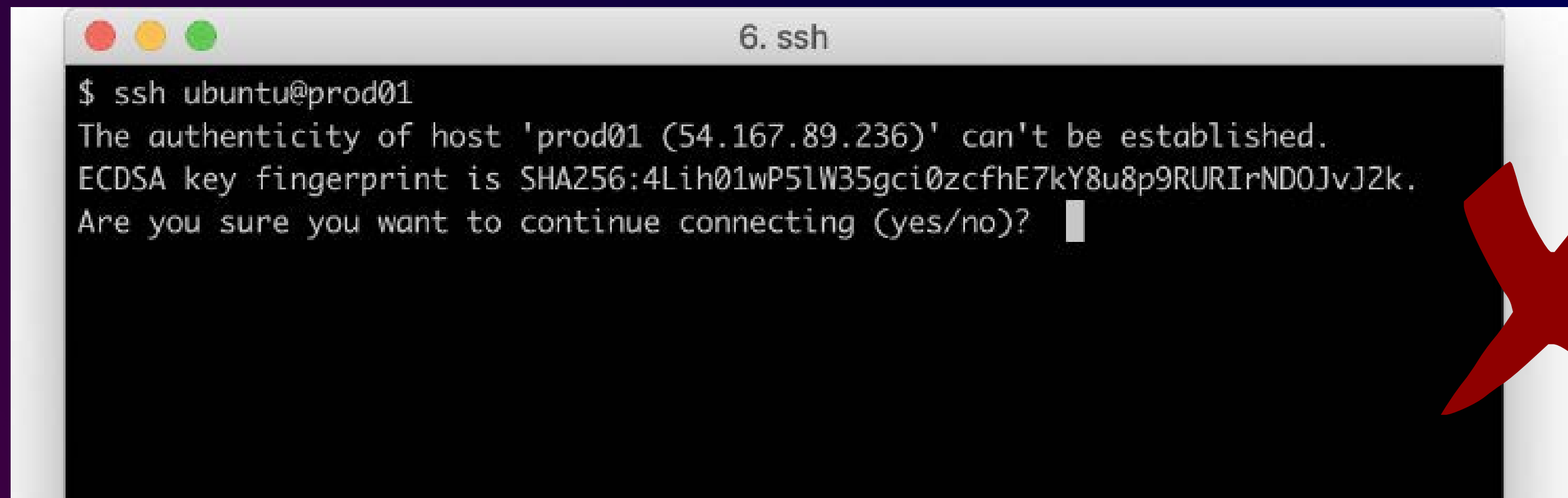
In an analysis of 15000 servers, out of the three million SSH keys that granted access to live production servers, 90% were no longer used
~ssh.com

SSH Keys do not expire

In an analysis of 15000 servers, out of the three million SSH keys that granted access to live production servers, 90% were no longer used
~ssh.com

- How do you keep track of unused or stale SSH keys
- Replacing a key means updating every server
- Can you easily update all your servers?

SSH keys encourage unsafe user behaviour and bad security practices

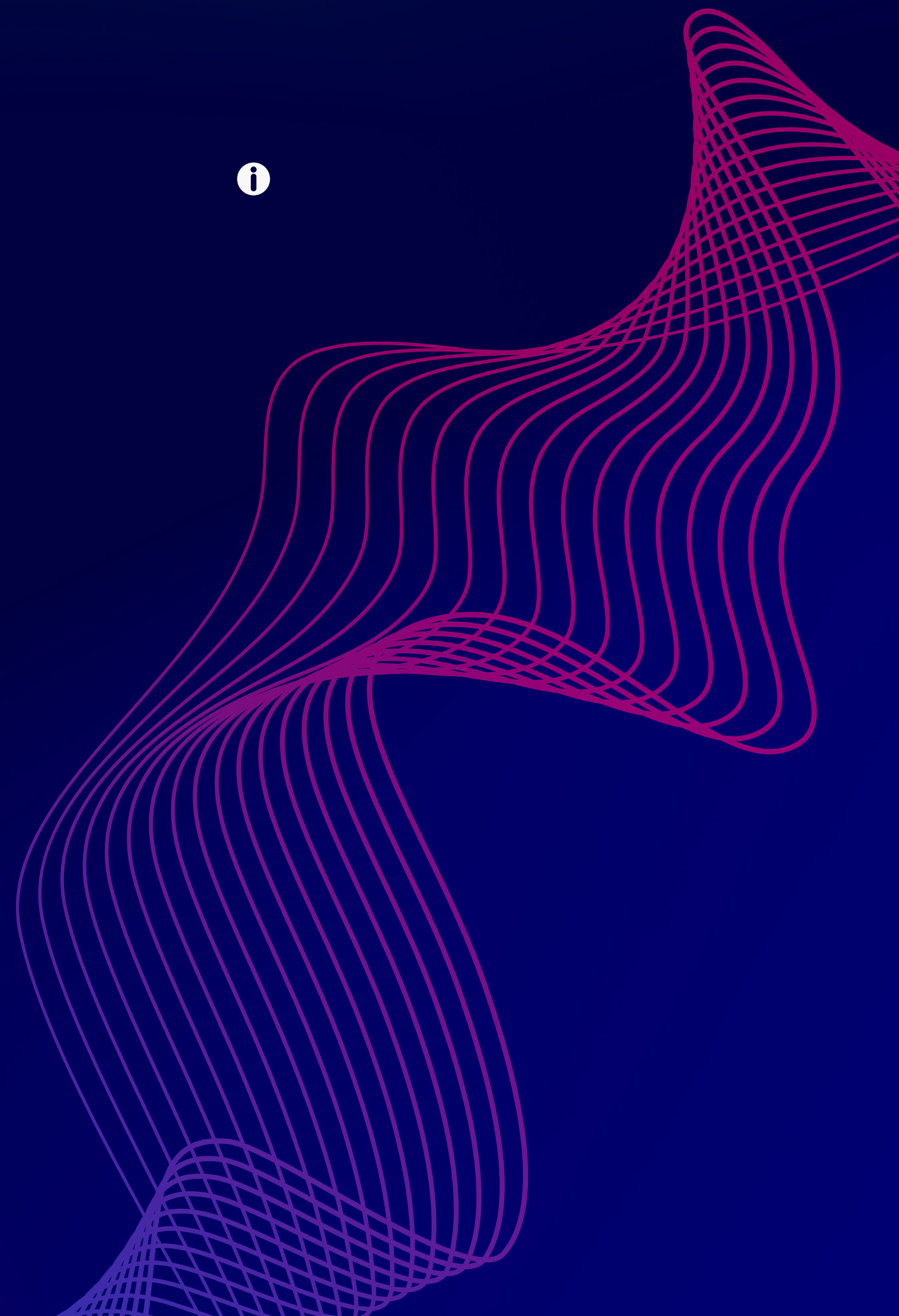
A terminal window titled "6. ssh" with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows the command "\$ ssh ubuntu@prod01" and the following output: "The authenticity of host 'prod01 (54.167.89.236)' can't be established. ECDSA key fingerprint is SHA256:4Lih01wP5lW35gci0zcfhE7kY8u8p9RURIrND0JvJ2k. Are you sure you want to continue connecting (yes/no)?". A cursor is visible after the question mark.

```
$ ssh ubuntu@prod01
The authenticity of host 'prod01 (54.167.89.236)' can't be established.
ECDSA key fingerprint is SHA256:4Lih01wP5lW35gci0zcfhE7kY8u8p9RURIrND0JvJ2k.
Are you sure you want to continue connecting (yes/no)?
```

X TOFU



USE SSH CERTIFICATES



An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```


An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```

An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```

An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```

An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```


An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```

An SSH user certificate

```
$ step ssh inspect id_ecdsa-cert.pub
```

```
id_ecdsa-cert.pub:
```

```
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
```

```
  Public key: ECDSA-CERT SHA256:O6M6oIjDm5gPm1/aTY619BgC3KSpS4c3aHVWxYh/uGQ
```

```
  Signing CA: ECDSA SHA256:EY2EXJGoPv2LA6yEbjH+sf9JjG9Rd45FH1Wt/6H1k7Y
```

```
  Key ID: "linda@example.com"
```

```
  Serial: 4309995459650363134
```

```
  Valid: from 2022-07-11T14:50:01 to 2022-07-11T18:50:01
```

```
  Principals:
```

```
    linda
```

```
  Critical Options: (none)
```

```
  Extensions:
```

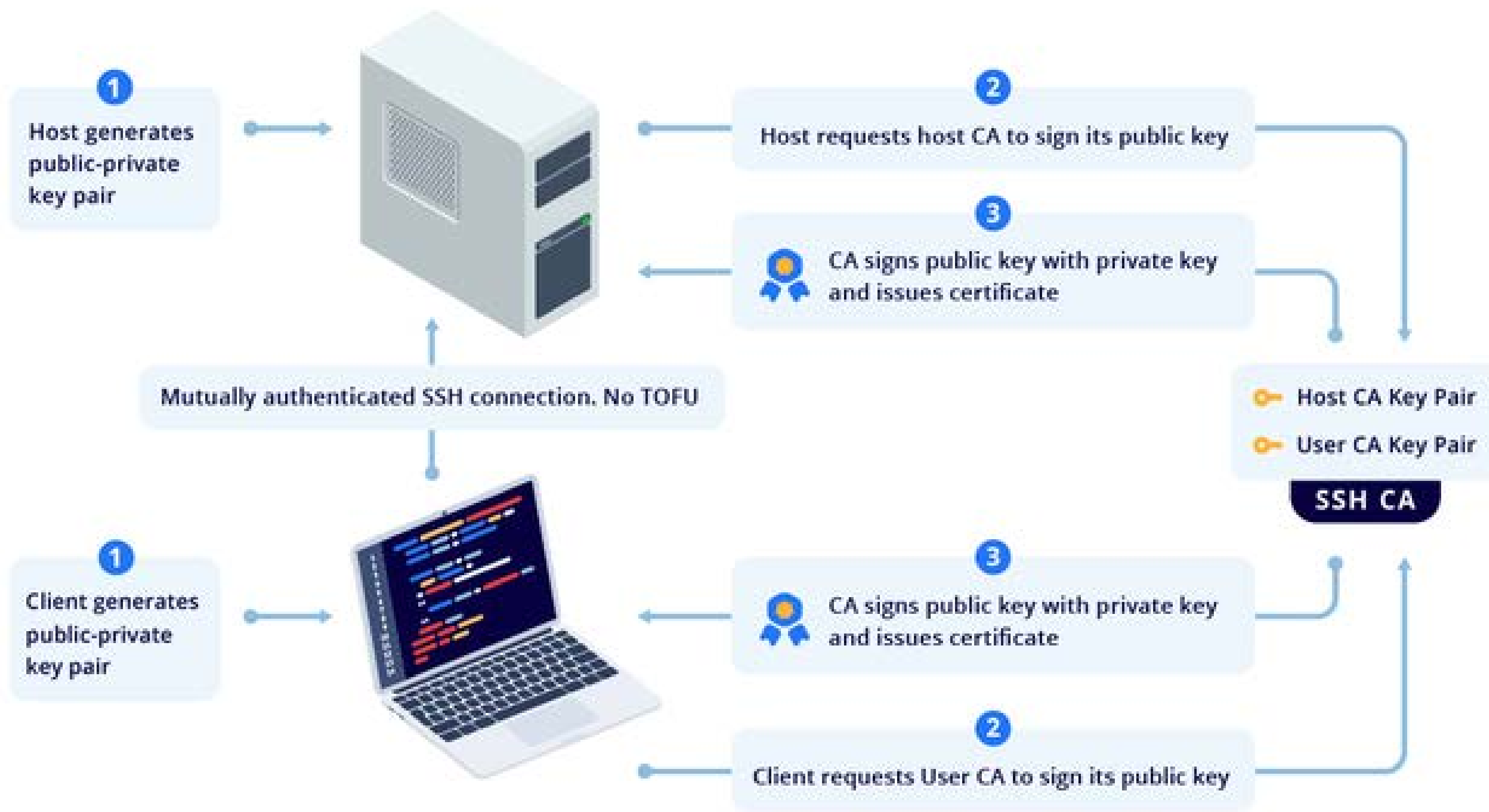
```
    permit-X11-forwarding
```

```
    permit-agent-forwarding
```

```
    permit-port-forwarding
```

```
    permit-pty
```

```
    permit-user-rc
```



WHY YOU SHOULD USE SSH CERTIFICATES

No more TOFU!!!!!!

- Hard trust that you're connecting to vetted servers

WHY YOU SHOULD USE SSH CERTIFICATES

Easier to maintain and manage

- No more static keys in keys in `~/.ssh/authorized_keys`
- Eliminate mundane key approval and distribution
- You'll know who owns a certificate, so better logs

WHY YOU SHOULD USE SSH CERTIFICATES

**SSH certificates expire
and can be revoked**

- Reduced risk surface
- On a need-to temporary access

HOW YOU CAN START USING SSH CERTIFICATES




Tooling

- ssh-keygen
- netflix/bless
- nsheridan/cashier
- uber/pam-ussb
- step-ca

SSO WITH SSH?

- You bring your IDP, we bring the SSH
- Removing a user from your identity provider terminates their SSH access in seconds

 Alice
\$ step ssh login

1. Sign in with Google
This yields an OAuth ID token for Alice, signed by Google



OAuth FLOW



2. Get an SSH user certificate

The CA exchanges Alice's OIDC token for an SSH Certificate



SMALLSTEP CA



3. SSH using Certificates
Alice's certificate is added to SSH agent. She can now SSH to any server she's allowed to use

Alice's ID token is verified using Google's public OAuth key.







```
$ ssh myserver
Welcome to Ubuntu ...
alice@myserver$
```



SMALLSTEP CA

myserver

On demand SSH Access on Slack?

Petition	Events
<div><div>Who are you requesting access for?</div><div><div><input checked="" type="radio"/>  Fouad Martin fouad@indent.com</div><div><input type="radio"/>  Someone else</div></div></div> <div><div>What do you need access to?</div><div><div> Google Group</div><div>Smallstep SSH Dev</div><div>▼</div></div></div> <div><div>Reason</div><div><div>to view the system logs</div><div>Why do you need these permissions?</div></div></div> <div><div>Preferred duration</div><div><div> 6 hours</div><div>How long do you need access for?</div></div></div> <div><div>Submit</div></div>	<div>Tip: create a request and grant access, then petition event history will show up here.</div>

**Convinced
Yet?**

- <https://smallstep.com/sso-ssh/how-it-works/index.html>
- <https://www.youtube.com/watch?v=ZhxLRlcNUM4>
- <https://www.youtube.com/watch?v=u2NSb12mzYI>

Thank You