

IoT-Cloud and Blockchain

Two virtual RPis

Dr. Phillip G. Bradford

University of Connecticut, Stamford CT. USA

phillip.bradford@uconn.edu,

Outline

Two virtual RPIs

Simple network model

SSH on RPIs

Message-digest hashing and public keys systems

Two Raspberry Pis

Running two QEMU VMs
Communication with
Raspberry Pis

Running Two RPIs

Duplicate RPi folder

RPi_1

RPi_2

Both contain the files:

2021_01_11-raspios-...-lite.img

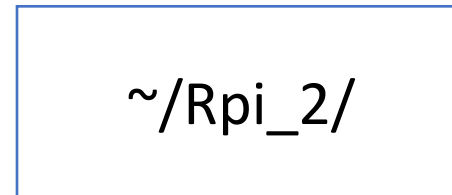
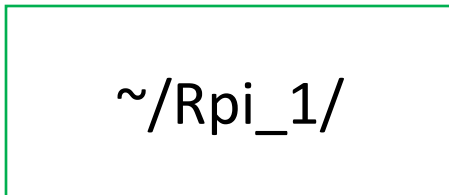
2021_01.qcow2

qemukernel

versatile-pb.dtb

QEMU State model

The system state is saved on the image (qcow2 file)
Multiple autonomous VMs require multiple images



Simple network model

Both RPIs



```
hostfwd=tcp::5022-:22
```

```
hostfwd=tcp::5023-:22
```

Simple network model

Both RPIs



hostfwd=tcp::**5022**-.22

hostfwd=tcp::**5023**-.22

Running Two RPIs

Two terminals

```
Term1> cd Rpi_1
```

```
Term2> cd Rpi_2
```

```
qemu-system-arm -M versatilepb
```

```
-cpu arm1176
```

```
-m 256 -hda "./2021-01.qcow2"
```

```
-net nic -net user,hostfwd=tcp::5022-:22
```

```
-dtb "./versatile-pb.dtb"
```

```
-kernel "./qemukernel"
```

```
-append "root=/dev/sda2 panic=1 rootfstype=ext4 rw" -no-reboot
```


Running Two RPIs

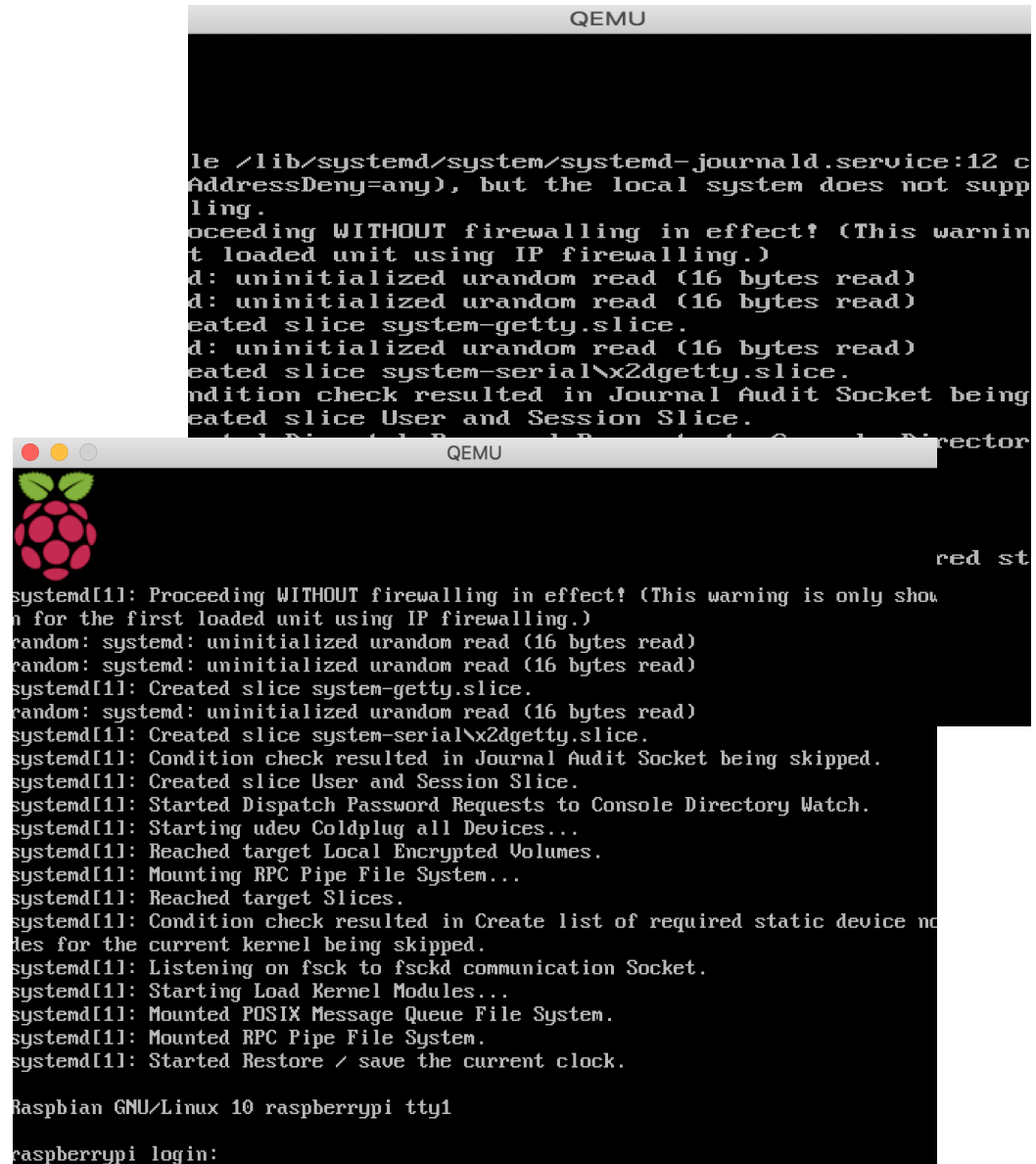
Duplicate RPi folder:

Rpi_1

Rpi_2

The qcow2 file should
be different for each RPi

Run qemu from both
folders



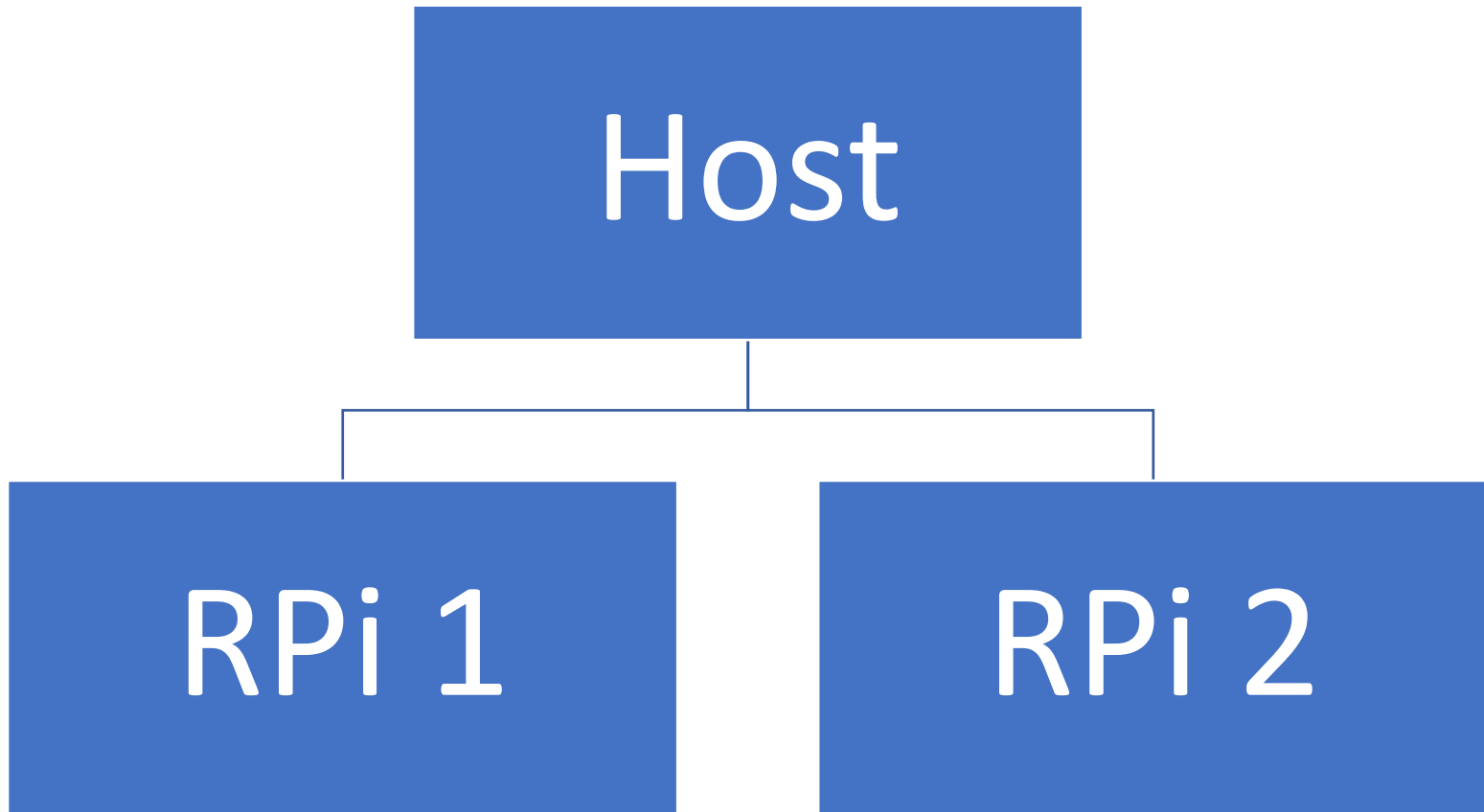
```
QEMU
le /lib/systemd/system/systemd-journald.service:12 c
AddressDeny=any), but the local system does not supp
ling.
ceeding WITHOUT firewalling in effect! (This warnin
t loaded unit using IP firewalling.)
d: uninitialized urandom read (16 bytes read)
d: uninitialized urandom read (16 bytes read)
eated slice system-getty.slice.
d: uninitialized urandom read (16 bytes read)
eated slice system-serial\x2dgetty.slice.
ndition check resulted in Journal Audit Socket being
eated slice User and Session Slice.
ted Dispatch Password Requests to Console Directory
red st

QEMU
systemd[1]: Proceeding WITHOUT firewalling in effect! (This warning is only show
n for the first loaded unit using IP firewalling.)
random: systemd: uninitialized urandom read (16 bytes read)
random: systemd: uninitialized urandom read (16 bytes read)
systemd[1]: Created slice system-getty.slice.
random: systemd: uninitialized urandom read (16 bytes read)
systemd[1]: Created slice system-serial\x2dgetty.slice.
systemd[1]: Condition check resulted in Journal Audit Socket being skipped.
systemd[1]: Created slice User and Session Slice.
systemd[1]: Started Dispatch Password Requests to Console Directory Watch.
systemd[1]: Starting udev Coldplug all Devices...
systemd[1]: Reached target Local Encrypted Volumes.
systemd[1]: Mounting RPC Pipe File System...
systemd[1]: Reached target Slices.
systemd[1]: Condition check resulted in Create list of required static device no
des for the current kernel being skipped.
systemd[1]: Listening on fsck to fsckd communication Socket.
systemd[1]: Starting Load Kernel Modules...
systemd[1]: Mounted POSIX Message Queue File System.
systemd[1]: Mounted RPC Pipe File System.
systemd[1]: Started Restore / save the current clock.

Raspbian GNU/Linux 10 raspberrypi tty1
raspberrypi login:
```

Guests and Hosts

System virtualization



SSH

Secure shell via ports 502X

Communication path

Host → RPI_1

Host → RPI_2

Installation

Install git on RPIs

Windows 10: easy ssh

MacOS: generate public-secret key pairs
put public key hash in known_hosts
secret key in keychain
push public key to RPIs

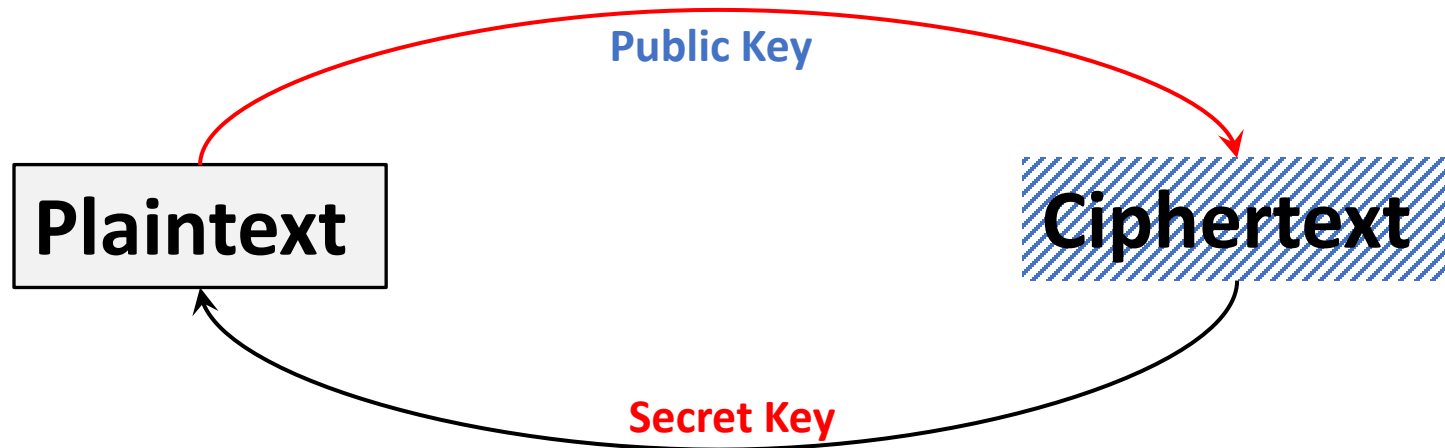
Message digest hash functions

Public key systems

Public key systems: RSA

SHA message digest function

Public Key Systems: secret message



Generate two keys

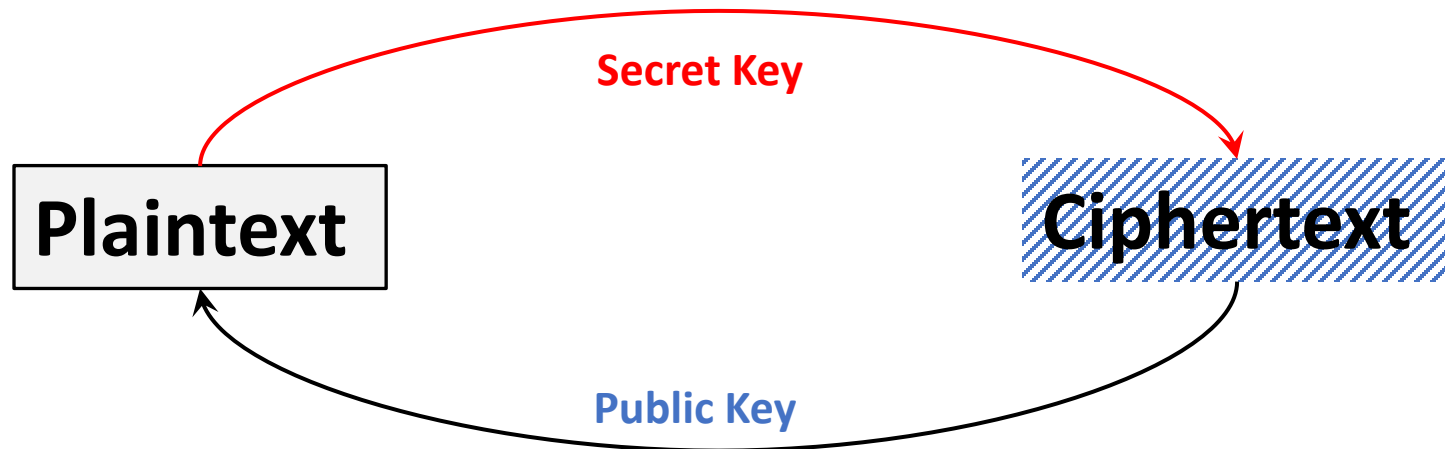
The public Key is open to the world

The **secret** key is secret, **only** you know it

Knowing the public key, intractable to find secret key

Each key is the other's cryptographic inverse

Public Key Systems: proving secret key

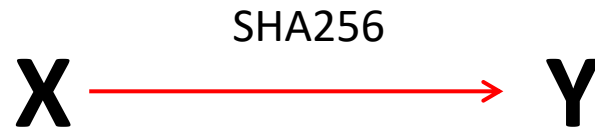
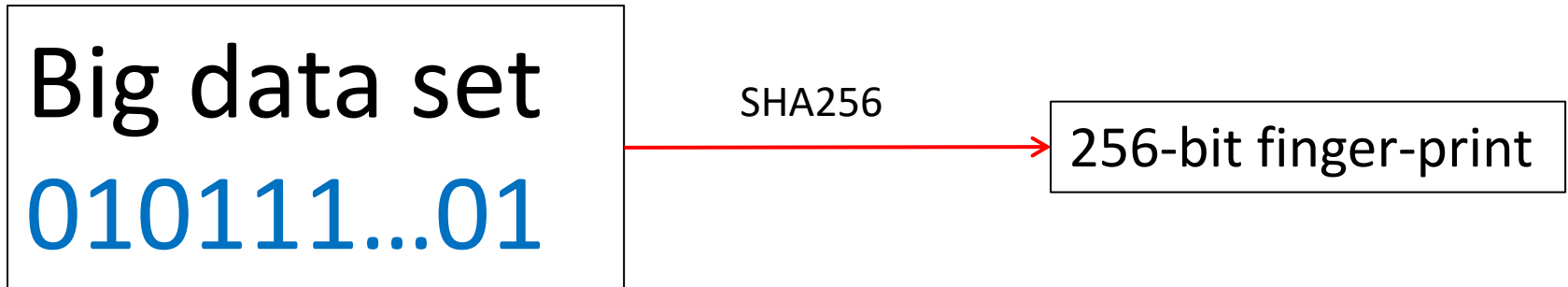


Take today's news as plaintext

Encrypt it with secret key and post cipher text

Anyone with public key can get the plaintext

SHA256 finger-prints



Given SHA256 output **Y**, not feasible to find input **X**
where **SHA256(X) = Y**

Hash functions: simplified

Outputs three decimal digits

Hash("Buy ETH") → 372

Hash("Buy BTC") → 281

Hash("Sell ETH") → 870

Any integer x in $\{0, 1, 2, \dots, 999\}$ and any string X we have:

$$P[\text{Hash}(X) = x] = \frac{1}{1000}$$

Public key systems

Proving you have a secret key

$$E_S[T] = C$$

$$D_P[C] = T$$

Letting anyone send you a secret message

$$E_P[T] = C$$

$$D_S[C] = T$$