

IoT and Blockchain

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



~/Rpi_1/

~/Rpi_2/

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

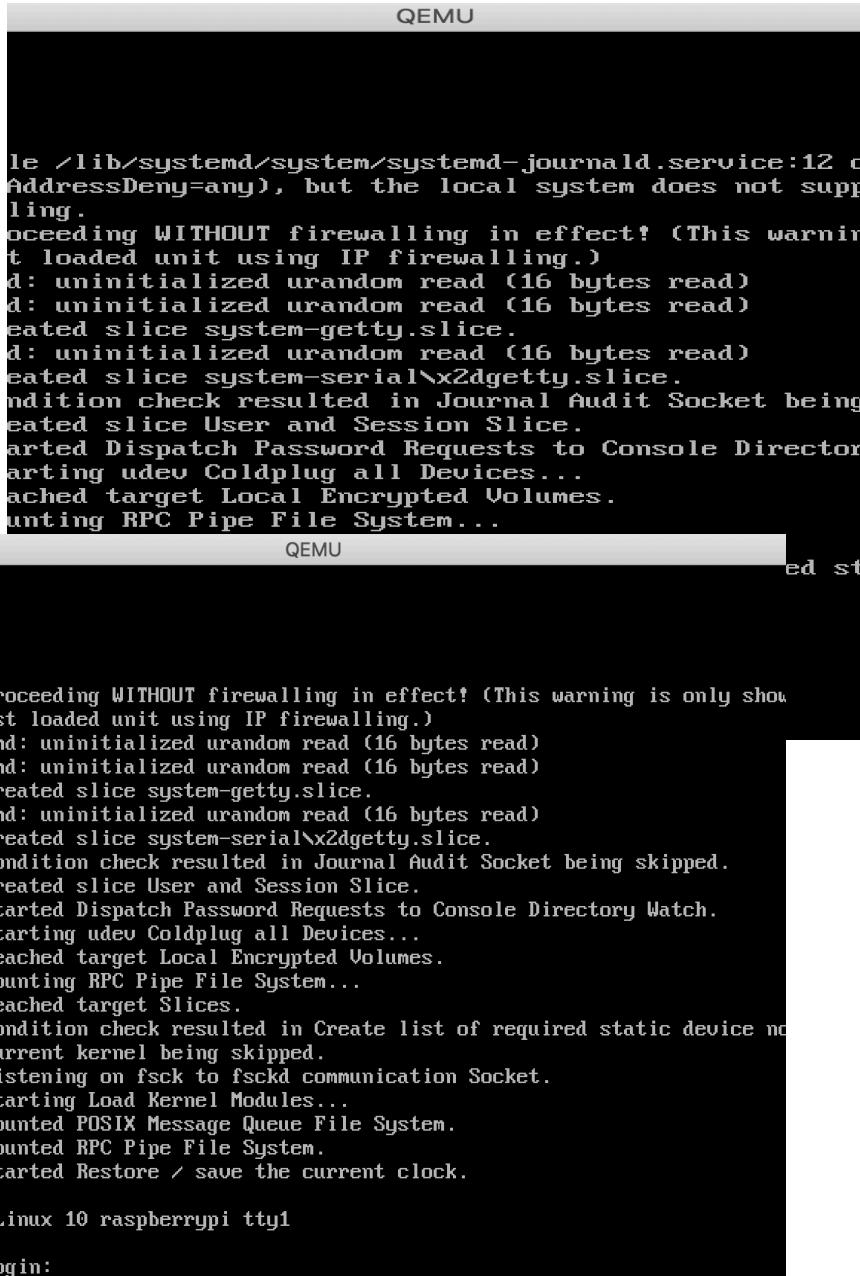
Duplicate RPi folder:

Rpi_1

Rpi_2

The qcows file should
be different for each RPi

Run qemu from both
folders



The image shows two separate QEMU windows side-by-side. Both windows have a title bar labeled "QEMU". The top window displays a black terminal-like background with white text, showing logs from a RPi boot process. The bottom window also has a black terminal-like background with white text, showing similar logs. In the bottom window, there is a small icon of a Raspberry Pi board in the top-left corner.

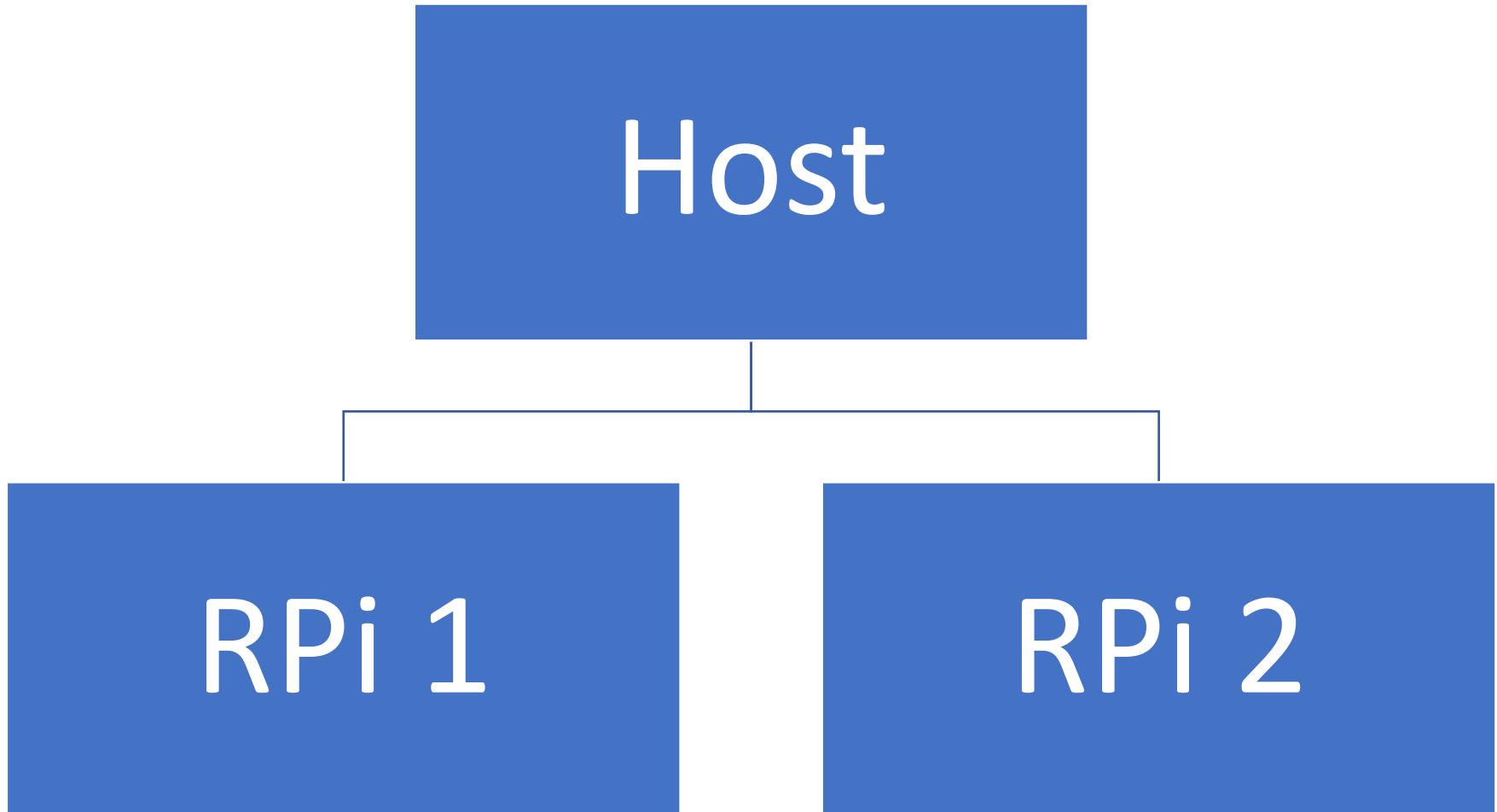
```
le /lib/systemd/system/systemd-journald.service:12 c
AddressDeny=any), but the local system does not supp
ling.
oceeding WITHOUT firewalls in effect! (This warnin
t loaded unit using IP firewalls.)
d: uninitialized urandom read (16 bytes read)
d: uninitialized urandom read (16 bytes read)
eated slice system-getty.slice.
d: uninitialized urandom read (16 bytes read)
eated slice system-serial\x2dgetty.slice.
dition check resulted in Journal Audit Socket being
eated slice User and Session Slice.
arted Dispatch Password Requests to Console Director
arting udev Coldplug all Devices...
ached target Local Encrypted Volumes.
unting RPC Pipe File System...  
  

QEMU
ed st
systemd[1]: Proceeding WITHOUT firewalls in effect! (This warning is only show
n for the first loaded unit using IP firewalls.)
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

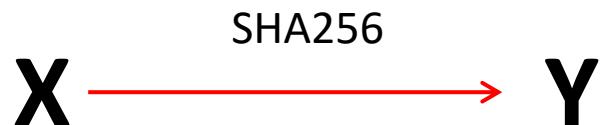
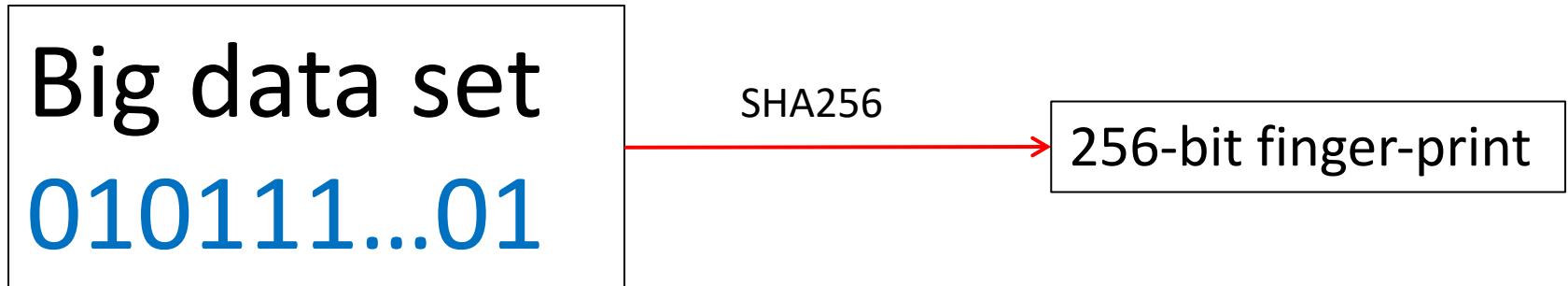
secret key in keychain

push public key to RPIs

Message digest hash functions
Public key systems

SHA message digest function
Public key systems: RSA

SHA256 finger-prints



Given SHA256 output Y , not feasible to find input X
where $\text{SHA256}(X) = Y$

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:

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

Collision intractability

SHA256 digests giant strings into 256 bits

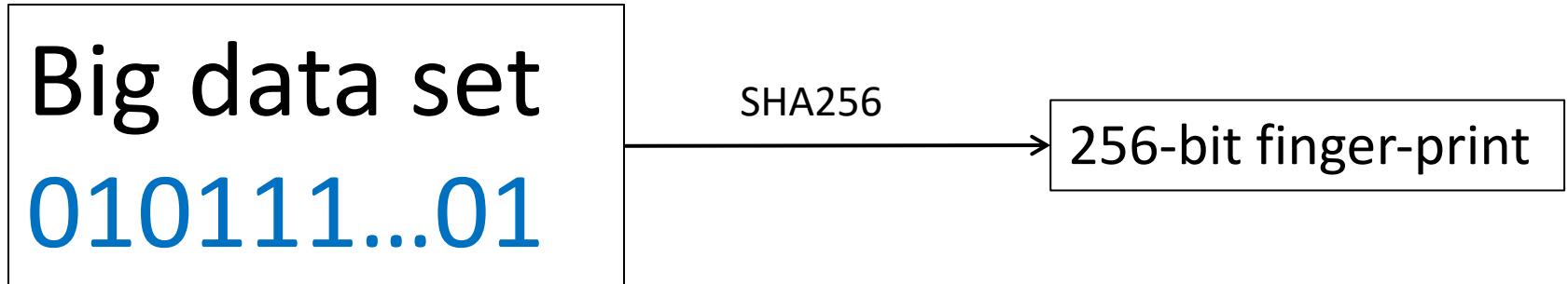
How many decimal digits in 2^{256} ?

$$\log_{10}(2^{256}) > 77$$

One third of 77 is about 25 decimal digits

$$P[SHA_{1/3}(X) = y] = \frac{1}{10000000000000000000000000000000}$$

SHA256 standard



SHA-256: Secure Hash (Digest) Algorithm

Designed by NSA and standardized by NIST (1994-2001)

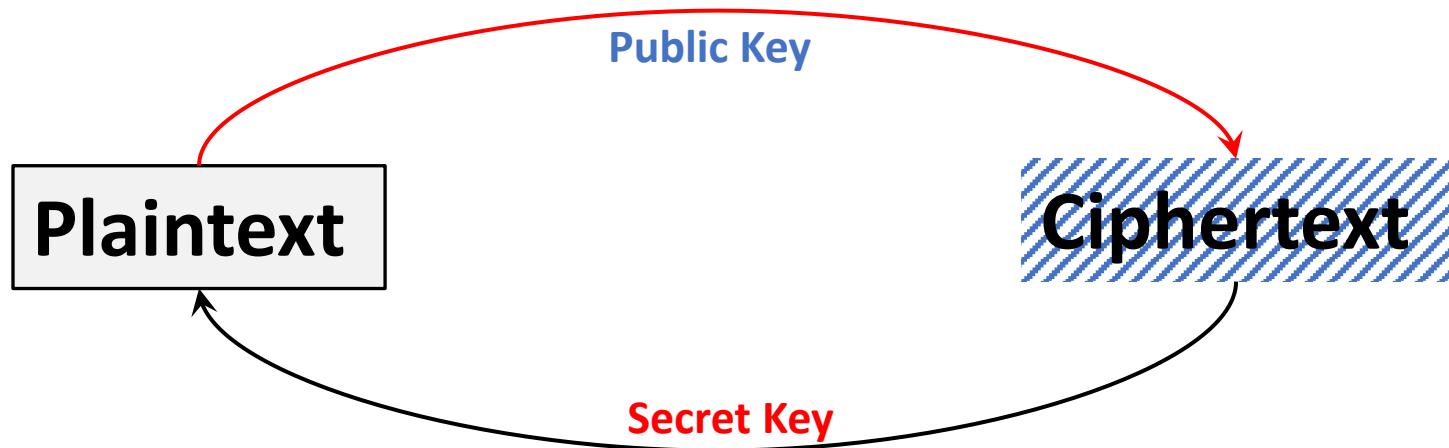
SHA-256 Hashing

One-way

Digital finger-prints

Digest takes up to $2^{64}-1$ bit input and outputs 256 bit output

Public Key Systems: secret message



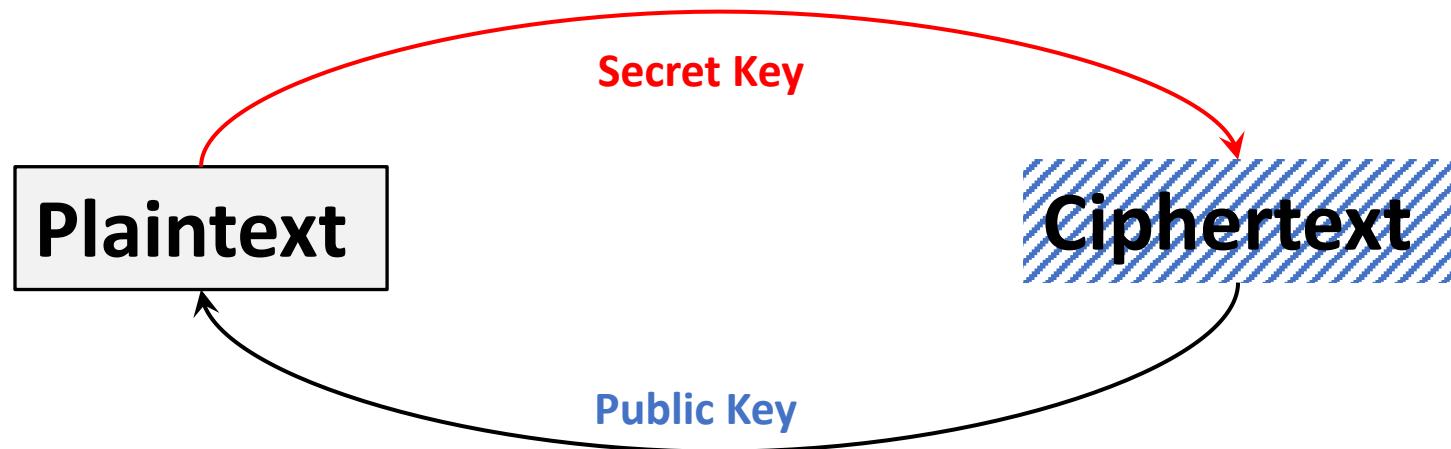
Generate two keys

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

Public key systems

Proving you have a secret key

$$E_S[T] = C$$

$$D_P[C] = T$$

Letting anyone send you a secret message

$$E_P[T] = C$$

$$D_S[C] = T$$