University of BRISTOL

Systems Security

COMSM1500

bristol.ac.uk

University of BRISTOL

Hardware Root of Trust

bristol.ac.uk

Plan

- Goal of trusted computing
- Rootkit
- Remote Attestation
- Return Oriented Programming
- Secure boot
- TPM

Trusted computing goal

- We need to trust computing devices
  - Laptop, phone, smart meter etc…
- You cannot know a computer is compromised by looking at it
  - Even harder remotely
- Malware/backdoor try to hide themselves
  - We are far from the Morris Worm ;-)

Arms race

- It is possible to detect malware from a higher privilege level
  - Malware wants to get there
  - Arms race to prevent this to happen
- Difficult to deal with rootkits
- What's a rootkit?

Arms race

- It is possible to detect malware from a higher privilege level
  - Malware wants to get there
  - Arms race to prevent this to happen
- Difficult to deal with rootkits
  - Set of software to maintain persistent present on a computer
  - Conceal their own presence or generally the presence of another software
  - Variant at different level (userspace -> kernel -> hypervisor -> firmware)
  - Can infect your boot sector… (formatting does not help!)

Rootkit

- Imagine malware X
- Rootkit will be a kernel module (i.e. think of something like a driver)
  - May filter ls results to exclude malware X files
  - May filter ps results to exclude malware X
  - May prevent deleting the files
  - May prevent killing the process
  - etc…
- You cannot kill (or notice) the rootkit from above!

Rootkit

Homework/exam question:
Explain the role of a rootkit
and how it works.

- Imagine malware X
- Rootkit will be a kernel module (i.e. think of something like a driver)
  - May filter ls results to exclude malware X files
  - May filter ps results to exclude malware X
  - May prevent deleting the files
  - May prevent killing the process
  - etc…
- You cannot kill (or notice) the rootkit from above!

University of BRISTOL

Software approach

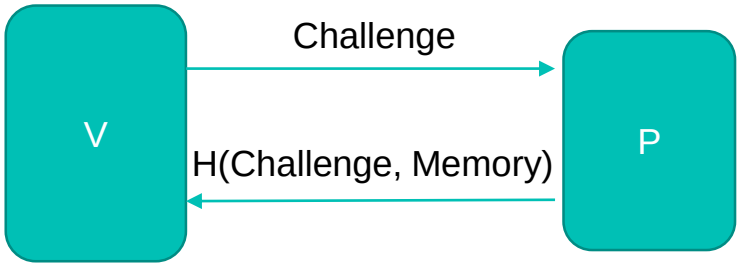
bristol.ac.uk

Remote attestation (software)

- Can we detect a compromised computer?
- We want to verify remotely the integrity of a system:
  - We know the hardware
  - We know the software that should be running
  - We want to verify the device is not compromised
- Problem: what is the sign of compromise
  - Code integrity

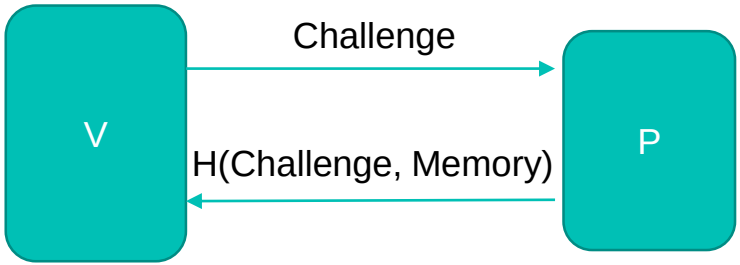
Remote attestation (software)

- Untrusted prover “P” and trusted verifier V
- V knows P memory content
- V send challenge with a nonce to P
- P compute a “checksum”
- V verify the checksum



Remote attestation (software)

- Untrusted prover “P” and trusted verifier V
- V knows P memory content
- V send challenge with a nonce to P
- P compute a “checksum”
- V verify the checksum



Homework/exam question:
Explain succinctly remote attestation

What remote attestation tells you

- Positive result
  - Correct memory content
  - Good device
- Negative result
  - Malfunctioning device
  - Malicious device
- No response
  - Malfunctioning device
  - Malicious device

University of BRISTOL

Problem?

bristol.ac.uk

University of BRISTOL

Problem?

One example

bristol.ac.uk

Return oriented programming

- Return to libc (but better!)
- Gain control of return pointer via buffer overflow
- Do not insert any code
- Jump to pieces of code that do something you want
- Chains those pieces of code (can easily get up to 1000 instructions)
- Check paper on github for details (docs/rop folder)

Return oriented programming

Homework/exam question:
Explain ROP
(i.e. read the paper)

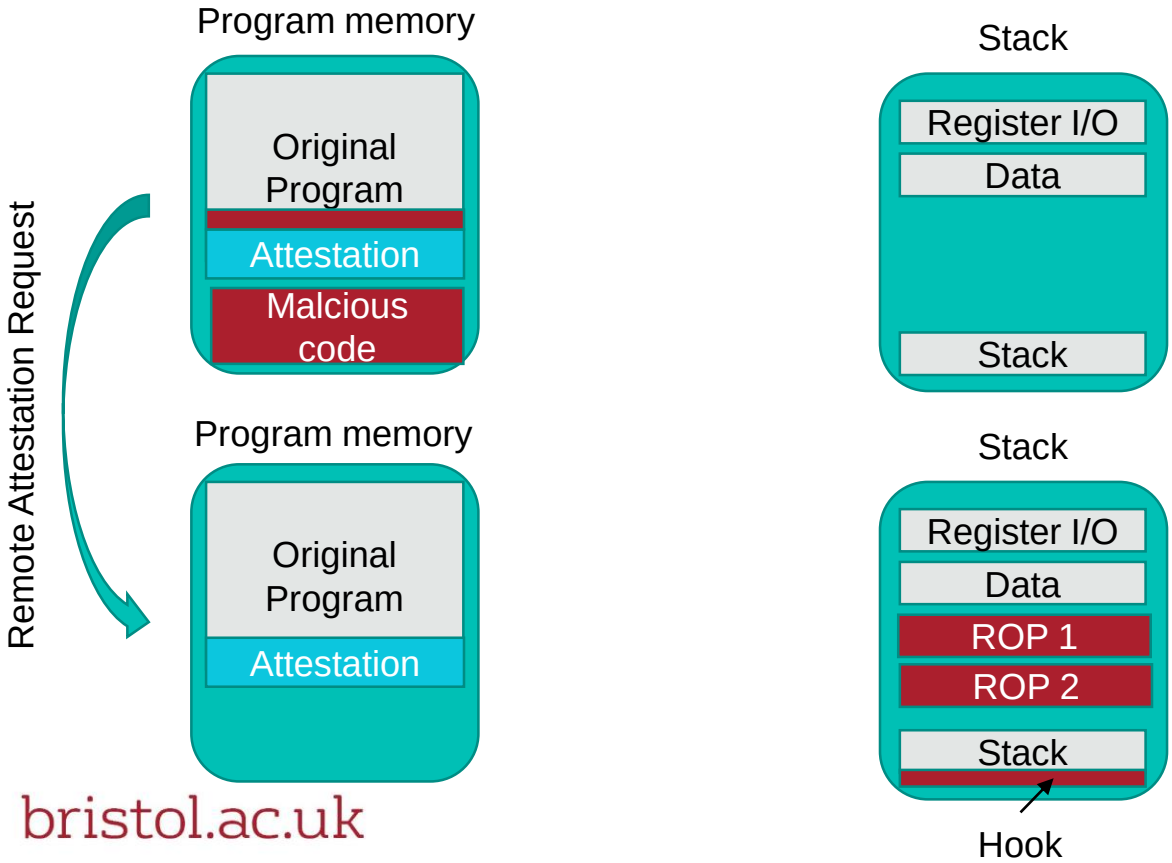
- Return to libc (but better!)
- Gain control of return pointer via buffer overflow
- Do not insert any code
- Jump to pieces of code that do something you want
- Chains those pieces of code (can easily get up to 1000 instructions)
- Check paper on github for details (docs/rop folder)

ROP rootkit

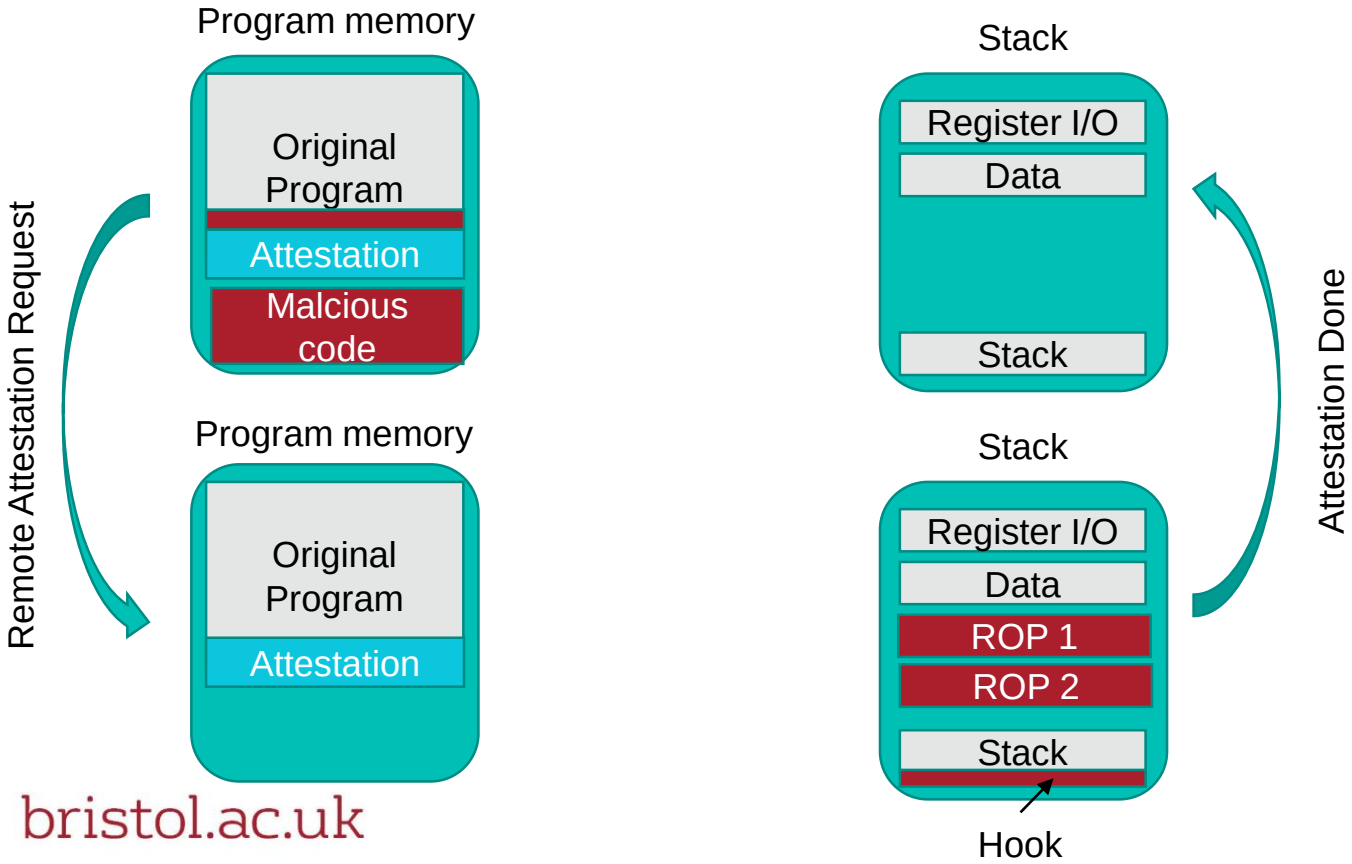

Program memory
Original Program
Attestation
Malcious code


Stack
Register I/O
Data
Stack

ROP rootkit


Program memory
Original Program
Attestation
Malcious code
Remote Attestation Request
Program memory
Original Program
Attestation
Stack
Register I/O
Data
Stack
Stack
Register I/O
Data
ROP 1
ROP 2
Stack
Hook

ROP rootkit


Program memory
Original Program
Attestation
Malcious code
Program memory
Original Program
Attestation
Remote Attestation Request
Stack
Register I/O
Data
Stack
Stack
Register I/O
Data
ROP 1
ROP 2
Stack
Hook
Attestation Done

ROP rootkit

Homework/exam question:
Explain how an ROP rootkit works.

- Powerful attack
  - Standard and well understood techniques
  - Difficult to prevent (arms race)
- Existing example
  - See github (docs/rop) for example of ROP rootkit on windows
- Hard to detect
  - Smart attacker trap code that check integrity…
  - … so when it is checked program memory is correct

Software-based Attestation

- Software-based attestation is difficult
  - impossible?
- We need hardware-based attestation

University of BRISTOL

Hardware approach

bristol.ac.uk

Hardware-based trusted computing

- Rely on hardware to provide some strong guarantee:
  - Prevent booting modified image (secure boot)
  - Proving integrity of running software (attestation)
  - Protecting secret from a modified OS (sealing)
  - Proving identity (authentication)

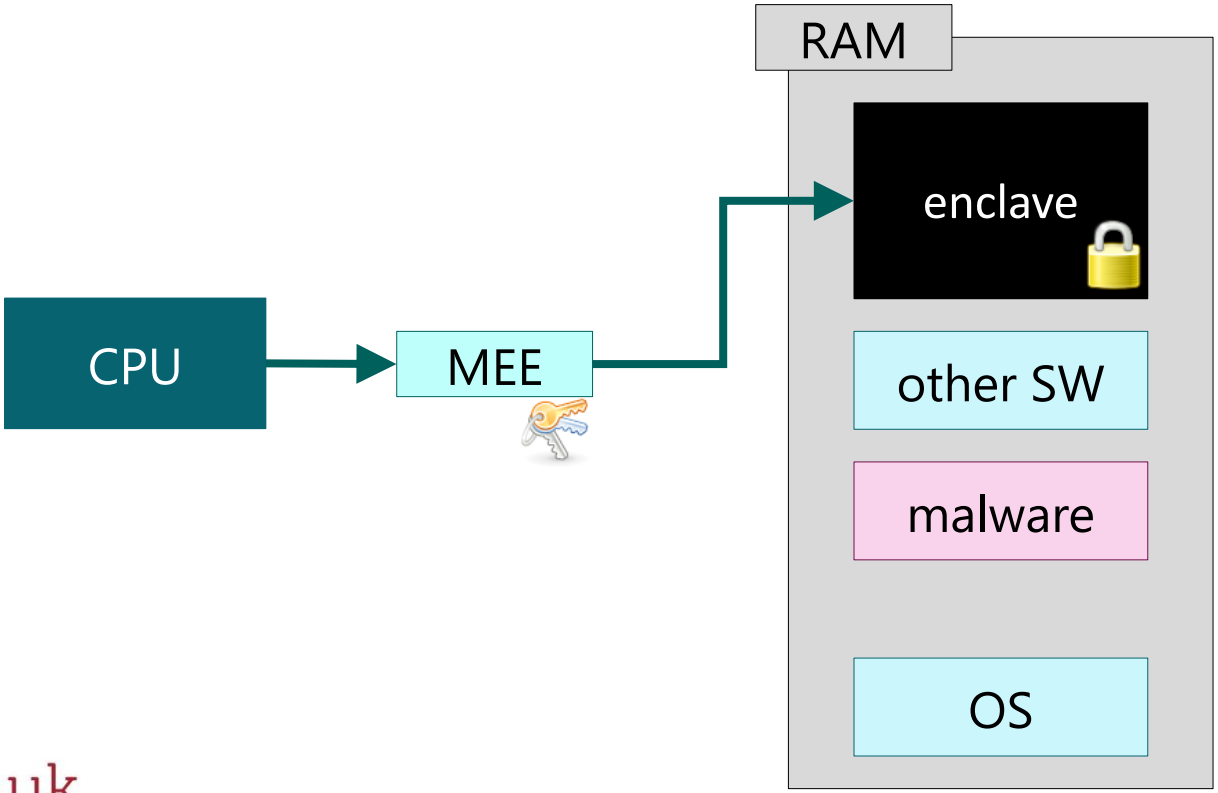
Hardware-based trusted computing

- Rely on hardware to provide some strong guarantee:
  - Prevent booting modified image (secure boot)
  - Proving integrity of running software (attestation)
  - Protecting secret from a modified OS (sealing)
  - Proving identity (authentication)
- Where is Intel SGX?

Hardware-based trusted computing

- Rely on hardware to provide some strong guarantee:
  - Prevent booting modified image (secure boot)
  - Proving integrity of running software (attestation)
  - Protecting secret from a modified OS (sealing)
  - Proving identity (authentication)
- Where is Intel SGX?
  - Sealed execution

Memory protection



Static Root of Trust

- Provide measurement of code at loading time
- Example TPM v1.1
  - Hashes code before loading
  - Store the hash in a TPM register
  - Value can be used as proof of the status of the system

Static Root of Trust

- Provide measurement of code at loading time
- Example: TPM v1.1
  - Hashes code before loading
  - Store the hash in a TPM register
  - Value can be used as proof of the status of the system
- Example: Secure Boot
  - Fixed boot loader in ROM
  - Contain a public key
  - Loads code
  - Verify the signature with the public key
  - if valid execute, otherwise halt

Static Root of Trust

Homework/exam question: Give examples of static roots of trust

- Provide measurement of code at loading time
- Example: TPM v1.1
  - Hashes code before loading
  - Store the hash in a TPM register
  - Value can be used as proof of the status of the system
- Example: Secure Boot
  - Fixed boot loader in ROM
  - Contain a public key
  - Loads code
  - Verify the signature with the public key
  - if valid execute, otherwise halt

University of BRISTOL

Problem?

bristol.ac.uk

Static Root of Trust problem

- Verifies only static information
  - Code at loading time
- Long running application
  - Do we reboot the system to do a sensitive operation?
- Runtime status of a device is not known
  - Attacker can compromise a system during execution
- Reboot not sufficient
  - iPhone has secure boot
  - … so only safe code is executed
  - yet permanent jailbreak
  - Configuration file loaded during boot exploit a vulnerability…
  - … solution verify configuration? Then it is not really a configuration…

Static Root of Trust problem

- Verifies only static information
  - Code at loading time
- Long running application
  - Do we reboot the system to do a sensitive operation?
- Runtime status of a device is not known
  - Attacker can compromise a system during execution
- Reboot not sufficient
  - iPhone has secure boot
  - … so only safe code is executed
  - yet permanent jailbreak
  - Configuration file loaded during boot exploit a vulnerability…
  - … solution verify configuration? Then it is not really a configuration…
- Solution SGX/ARM trust zone (context dependents etc…)

Secure boot

- Good to prevent booting untrusted image
- In control of the owner of the key
  - In general the device manufacturer
- Does not tell much about the runtime status of the device
  - Can be defeated after every reboot

Secure boot

Homework/exam question:
Discuss the limitation of
Secure boot.

- Good to prevent booting untrusted image
- In control of the owner of the key
  - In general the device manufacturer
- Does not tell much about the runtime status of the device
  - Can be defeated after every reboot

University of BRISTOL

TPM-based Trusted Computing

bristol.ac.uk

TPM (Trusted Platform Module)

- Trusted Computing Group
  - Microsoft, Intel, IBM etc…
- Promoting standard for more trusted computing
  - Additional chip on the motherboard
  - … called TPM
- Used for
  - Disk encryption
  - System Integrity
  - Password protection
  - … and more

Trusted Computing vs Secure Boot

- Secure boot allows signed software to execute
- Authenticated boot
  - Make measurement of the software being executed
  - e.g. can be verified by third parties

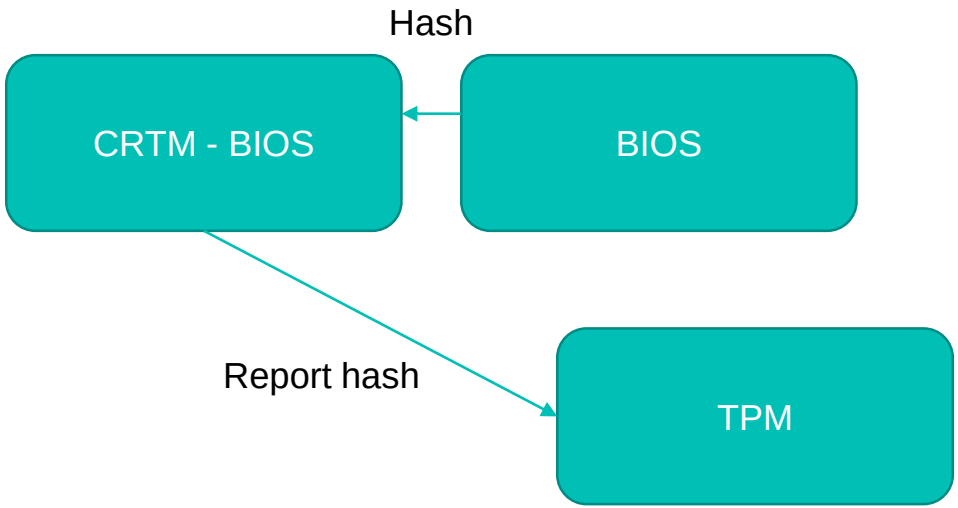
- You could combine both

Requirements

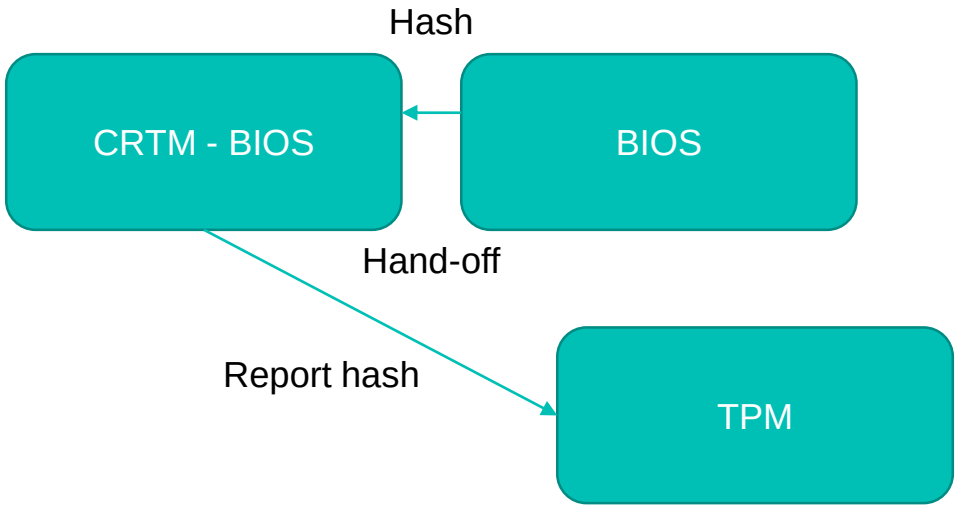
- We can achieve trust if we can verify the system has booted correctly
- We assume the pc hardware has not been modified
  - Key function is in the hardware TPM
- We need to monitor the boot process
  - Initial boot measure by the “Core Root of Trust” (ROM)
  - Hash the BIOS, store results in TPM, start the BIOS
  - BIOS do its job, load the next stage, hash it store in TPM etc…

Authenticated Boot

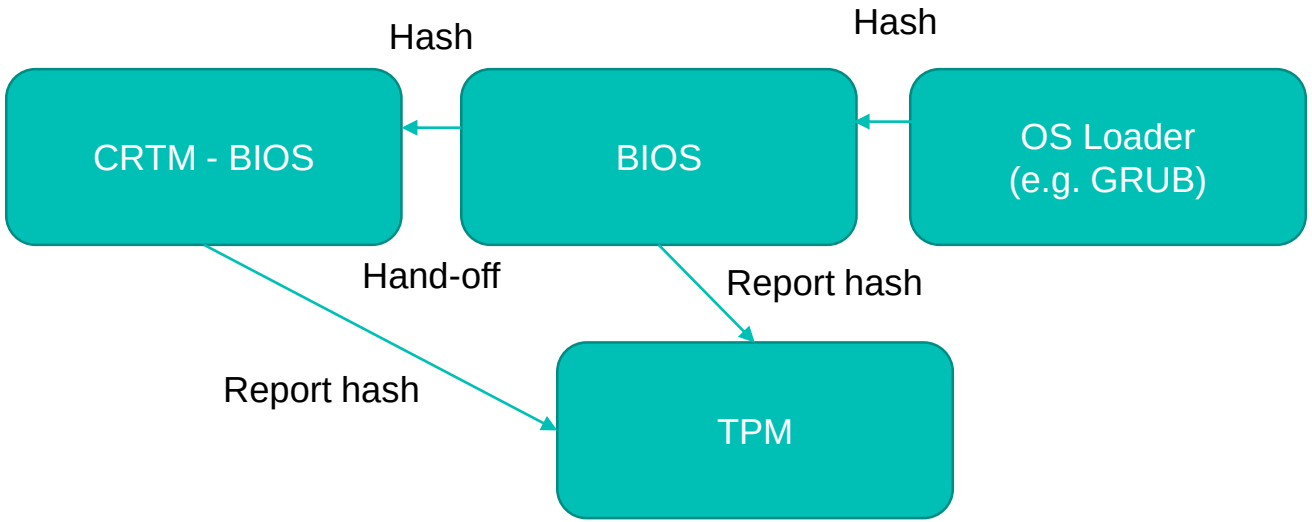
Authenticated Boot



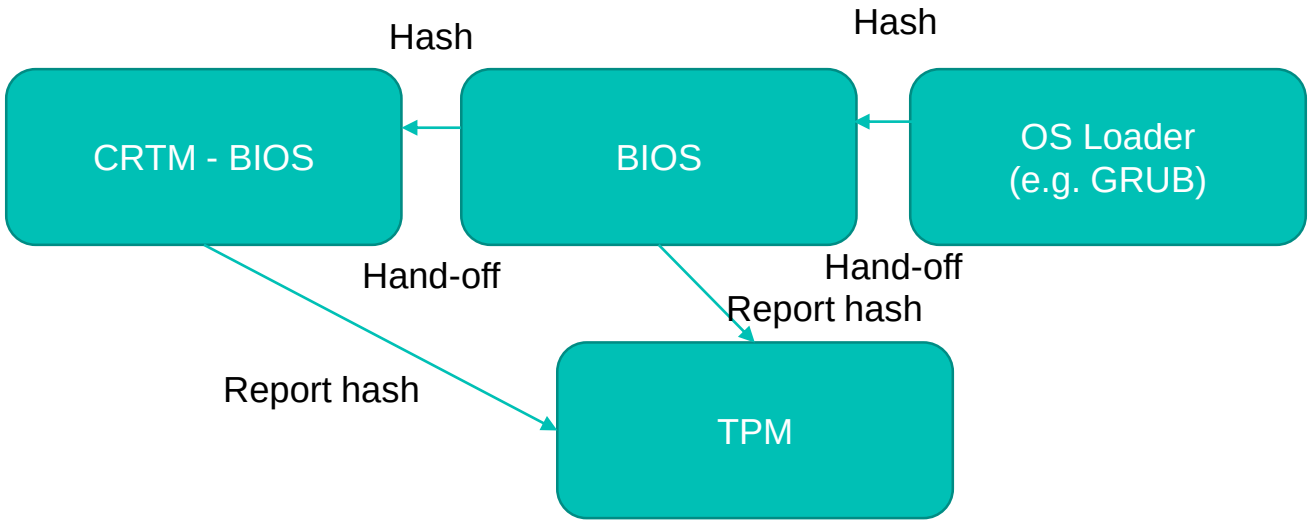
Authenticated Boot



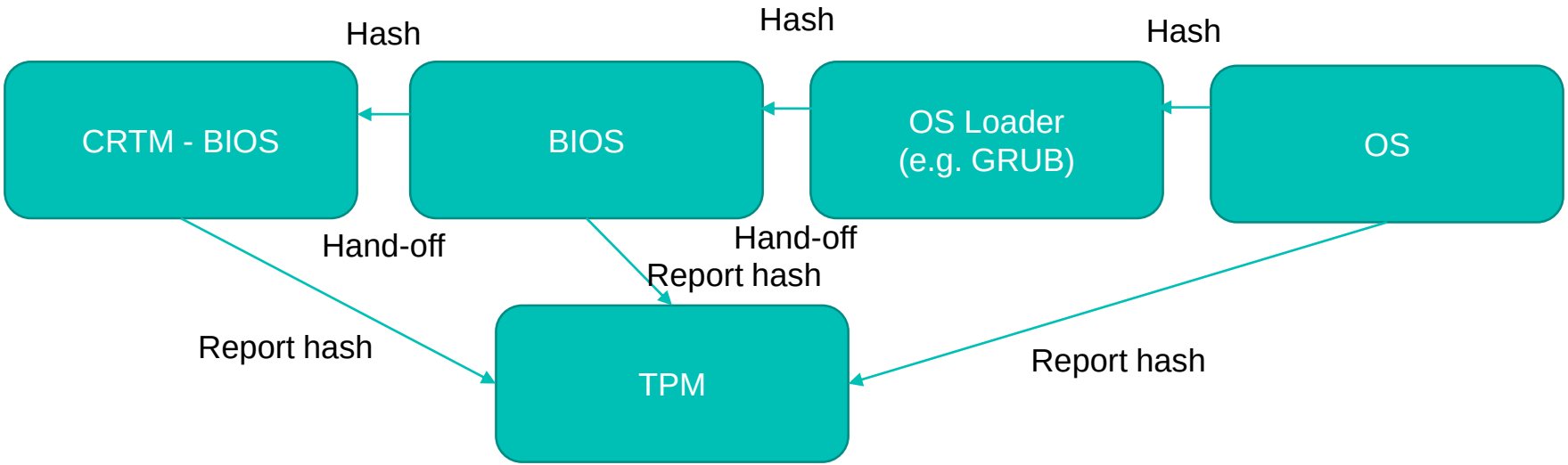
Authenticated Boot



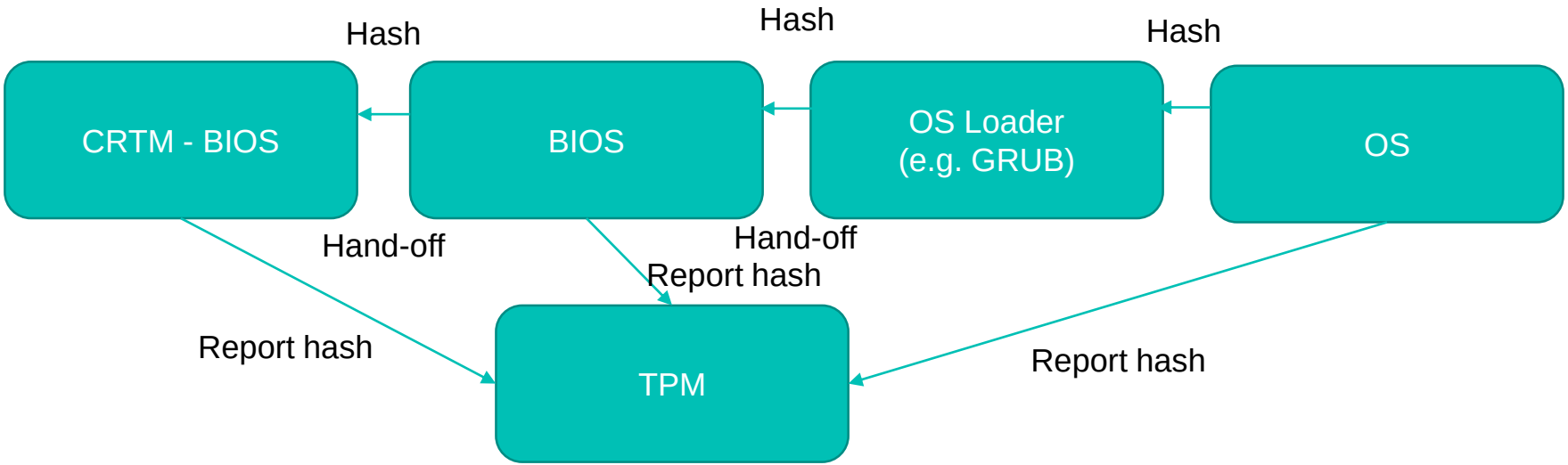
Authenticated Boot



Authenticated Boot



Authenticated Boot (simplified)



TPM registers

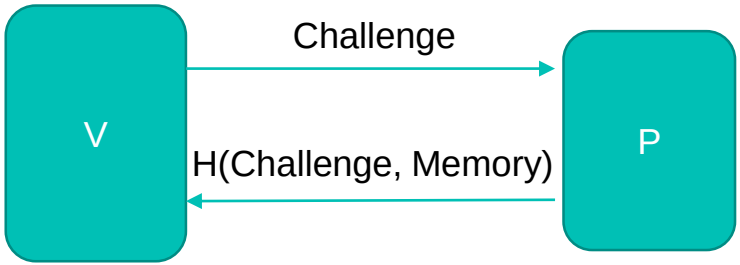
- Platform configuration registers (PCRs)
  - Used to store platform integrity metrics
- A PCR hold a summary of a series of value
  - Not the entire chain of hash
  - The chain can be infinite
- A PCR register is extended
  - PCR = HASH(PCR | new measurement)
  - Shielded TPM location (i.e. cannot be modified from outside)
  - Measurement are provided by software

TPM and Remote Attestation

- PCR cannot be modified
  - Only reset at reboot
- TPM contains a key used to sign the attestation
- Verifier
  - Verify the TPM certificate/key
  - Verify the PCRs
- Attestation
  - PCRs value
  - sign(PCRs, challenge[nonce])

Remote attestation

- Untrusted prover “P” and trusted verifier V
- V knows P memory content
- V send challenge with a nonce to P
- P compute a “checksum”
- V verify the checksum



TPM and Remote Attestation

- You need not to stop at the OS
  - Can attest kernel modules (e.g. drivers)
  - Applications…
  - Configurations…
  - Scripts…
  - Where to stop?
  - Problem with load order? (remember it is a chain)
- Check IMA paper on github (docs/ima)
  - Linux implementation by IBM

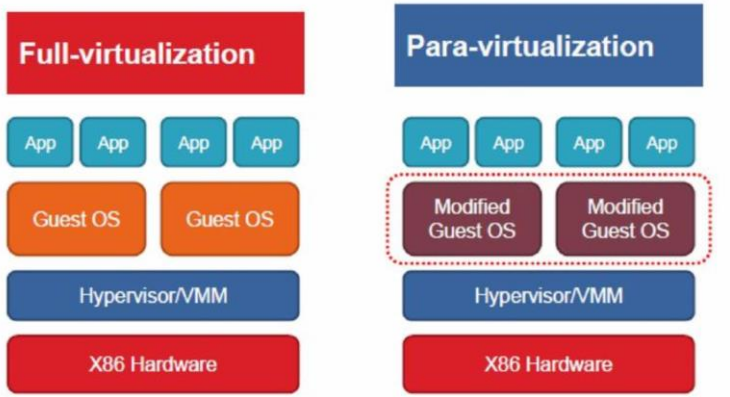
TPM and Remote Attestation

Homework/exam question:
Explain how to perform
RA with TPM.

- You need not to stop at the OS
  - Can attest kernel modules (e.g. drivers)
  - Applications…
  - Configurations…
  - Scripts…
  - Where to stop?
  - Problem with load order? (remember it is a chain)
- Check IMA paper on github (docs/ima)
  - Linux implementation by IBM

TPM and Remote Attestation

- You need not to stop at the OS
  - Can attest kernel modules (e.g. drivers)
  - Applications…
  - Configurations…
  - Scripts…
  - Where to stop?
  - Problem with load order? (remember it is a chain)
- Can also work for VM (check vTPM by IBM)



Plan

- Goal of trusted computing
- Rootkit
- Remote Attestation
- Return Oriented Programming
- Secure boot
- TPM

University of BRISTOL

Thank you, questions?

Office MVB 3.26

bristol.ac.uk