科学技術振興機構
Japan Science and Technology Agency
行政院國家科學委員會
National Science Council

Hypervisor-Based Systems for Malware Detection and Prevention

Yoshihiro Oyama (大山 恵弘)
The University of Electro-Communications (電気通信大学), Tokyo, Japan

This Talk

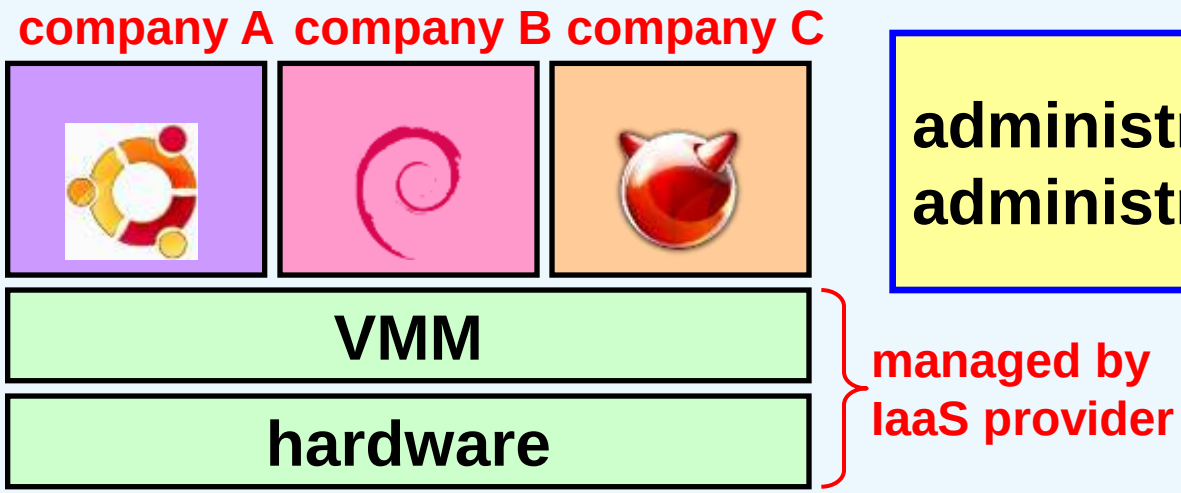
- I introduce two hypervisor-based security systems developed in our laboratory
  - HyperSlow: Extremely slowing malware execution by controlling the speed of virtual time
  - BVMD: Detecting malware signatures in a thin hypervisor

科学技術振興機構
Japan Science and Technology Agency
行政院國家科學委員會
National Science Council

HYPERSLOW: EXTREMELY SLOWING MALWARE EXECUTION BY CONTROLLING THE SPEED OF VIRTUAL TIME

Background

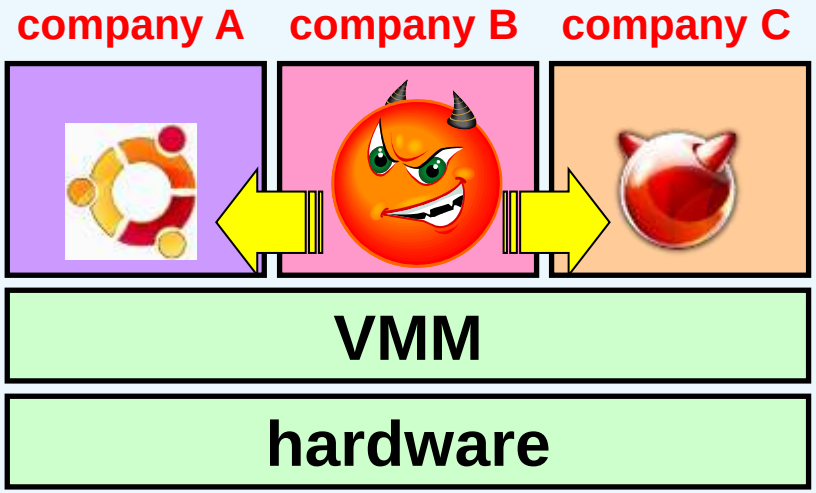
- IaaS (Infrastructure as a Service)
  - Ex.: Amazon EC2, Rackspace Cloud
  - IaaS provider lends VMs to customers
  - Customers freely use their own VMs



administrator of VMM ≠ administrator of guest OS

Malware Infection

- One guest OS is infected with malware that consumes resource
  - Wastes CPU, memory, …
  - May have bad effect on other guest OSes



- Much literature showed that some malware could be detected from the VMM layer
  - VMwatcher [Jiang et al. '10], Lares [Brian et al. '08], Lycosid [Jones et al. '08]

Problem

- VMM administrator cannot modify the data managed by the guest OS
  - Ethically: Customer possesses the guest OS
  - Technically: Kind and version of the guest OS is unknown --- Windows? Linux? What version?
- Unfortunately, existing countermeasures are limited to coarse-grain ones
  - Ex.1: Stop whole VM
  - Ex.2: Drop all network packets from/to the VM
  - They affect even good processes running in the guest OS!

Goal

- Develop a VMM-based method for deactivating malware
  - Pinpointing the malware process
  - Mostly guest-OS-independent
  - Specifically, the method slows a malware process significantly
- Implement a system based on the method, HyperSlow, and demonstrate the effectiveness

HyperSlow

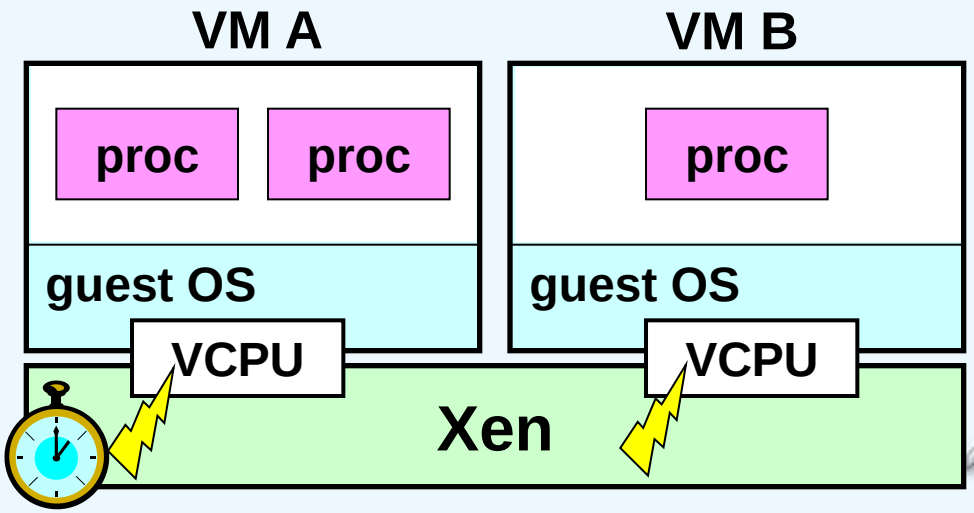
- Increases the speed of virtual time only while a malware process is running
  - It changes the rate of virtual timer interrupts and system time (elapsed time from boot)
- Is not a malware detection system, but a prevention system
  - It must cooperate with other detectors
- Xen-based

Process Scheduling Basics

- OS kernels (incl. Linux and Windows) use timer interrupts and/or system time for process scheduling
- Linux case:
  - When kernel receives a timer interrupt, it calculates the time consumed by the current process
    - Consumed time is calculated using system time
    - System time is calculated from hardware clock such as TSC
  - If the process has consumed all of the assigned time slice, it is preempted

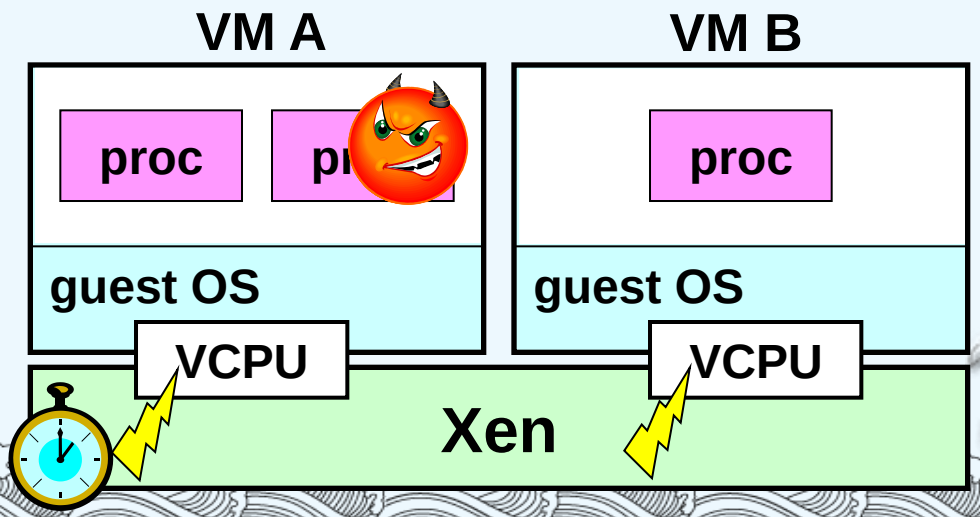
Virtual Timer Interrupts in Xen

- Xen periodically injects virtual timer interrupts to virtual CPUs
  - Based on them, guest OS schedules processes and performs timekeeping

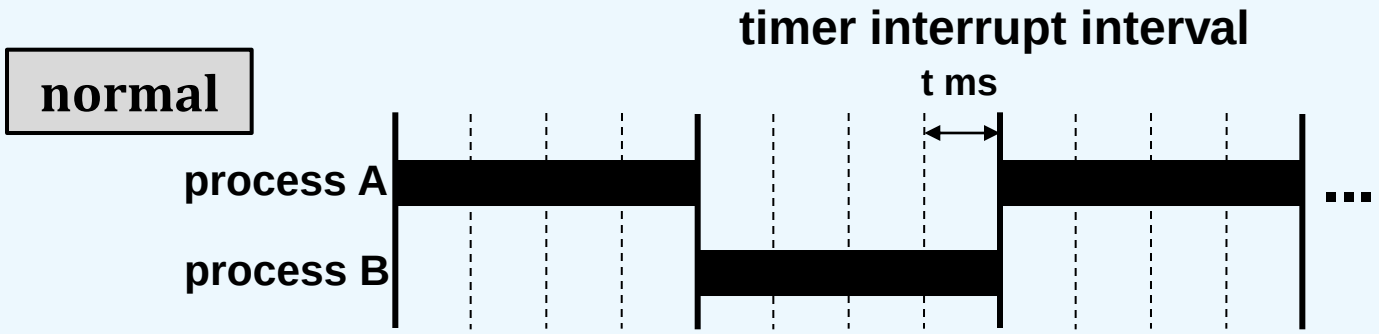


Basic Idea

- HyperSlow extremely shortens the interval between virtual timer interrupts, only while a malware process is scheduled
  - Promotes context switch by having guest OS kernel misunderstand the elapsed time
  - Uses CR3 register as a pseudo PID [Jones et al. '06]

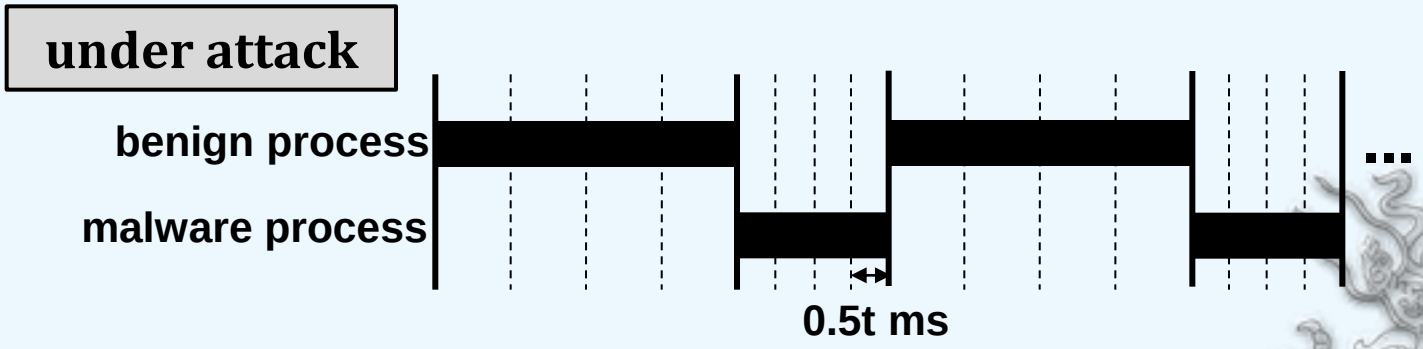


Example



While the malware is running, HyperSlow sets

- **Timer interrupt interval: ½**
- **Speed of system time: twice**



Malware obtains only half of the original time slice!

科学技術振興機構
Japan Science and Technology Agency
行政院國家科學委員會
National Science Council

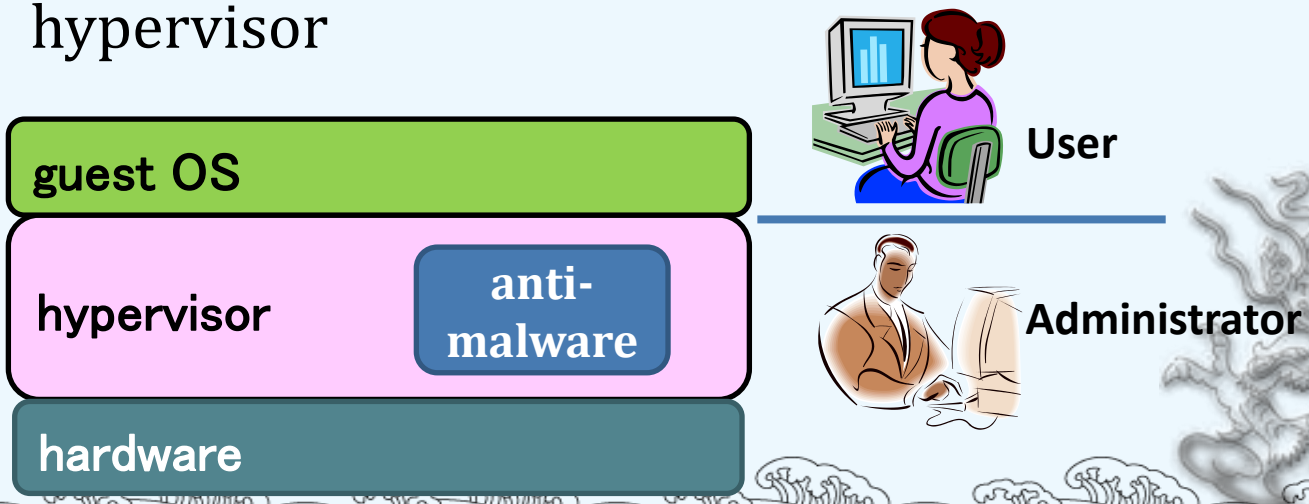
BVMD: DETECTING MALWARE SIGNATURES IN A THIN HYPERVISOR

Background

- Anti-malware software is effective to detect and/or prevent attacks of malware
- But, the anti-malware approach has several problems
  - Malware can deactivate anti-malware operations if it compromises the OS kernel
  - Some users may uninstall or turn off anti-malware
    - Intentionally or due to operation error

Our Approach

- Running anti-malware under OS (in hypervisor layer)
  - Administrator installs a hypervisor and guest OS in the machine of each member
  - Member receives the root privilege of the guest OS
  - Administrator keeps the admin privilege of the hypervisor

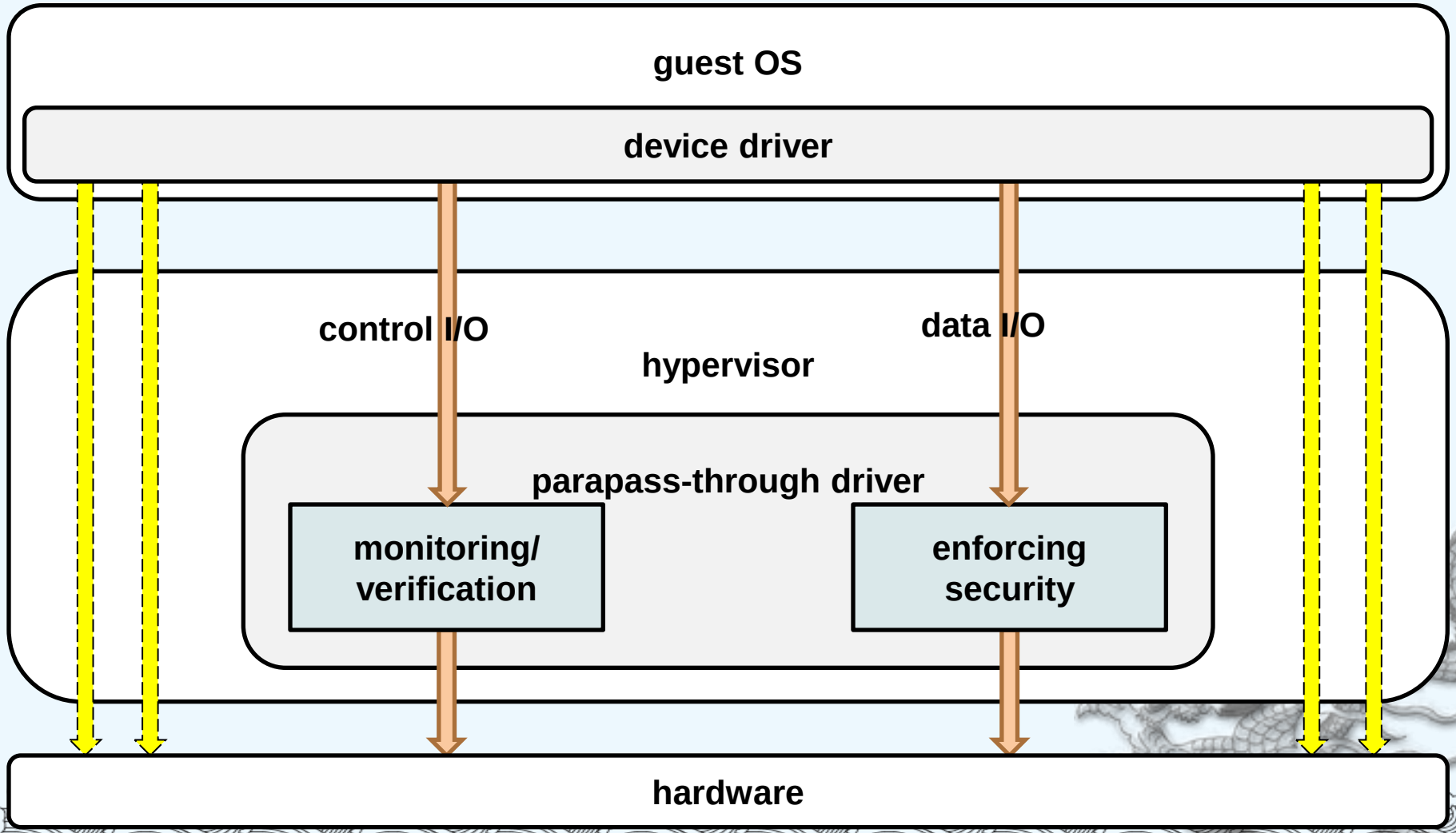


BVMD: the Proposed System

- A thin hypervisor provides malware detection facility
  - Use BitVisor [Shinagawa et al., VEE09], a parapass-through hypervisor
  - Detect malware signatures in I/O data transferred between guest OS and devices
- Advantage
  - Malware detection does not depend on guest OS
    - Windows, Linux, ...
  - Malware is still detected if the guest OS is compromised
  - User of guest OS (even an administrator) cannot deactivate anti-malware
  - Modification of guest OS or execution of special daemon is not needed

BitVisor

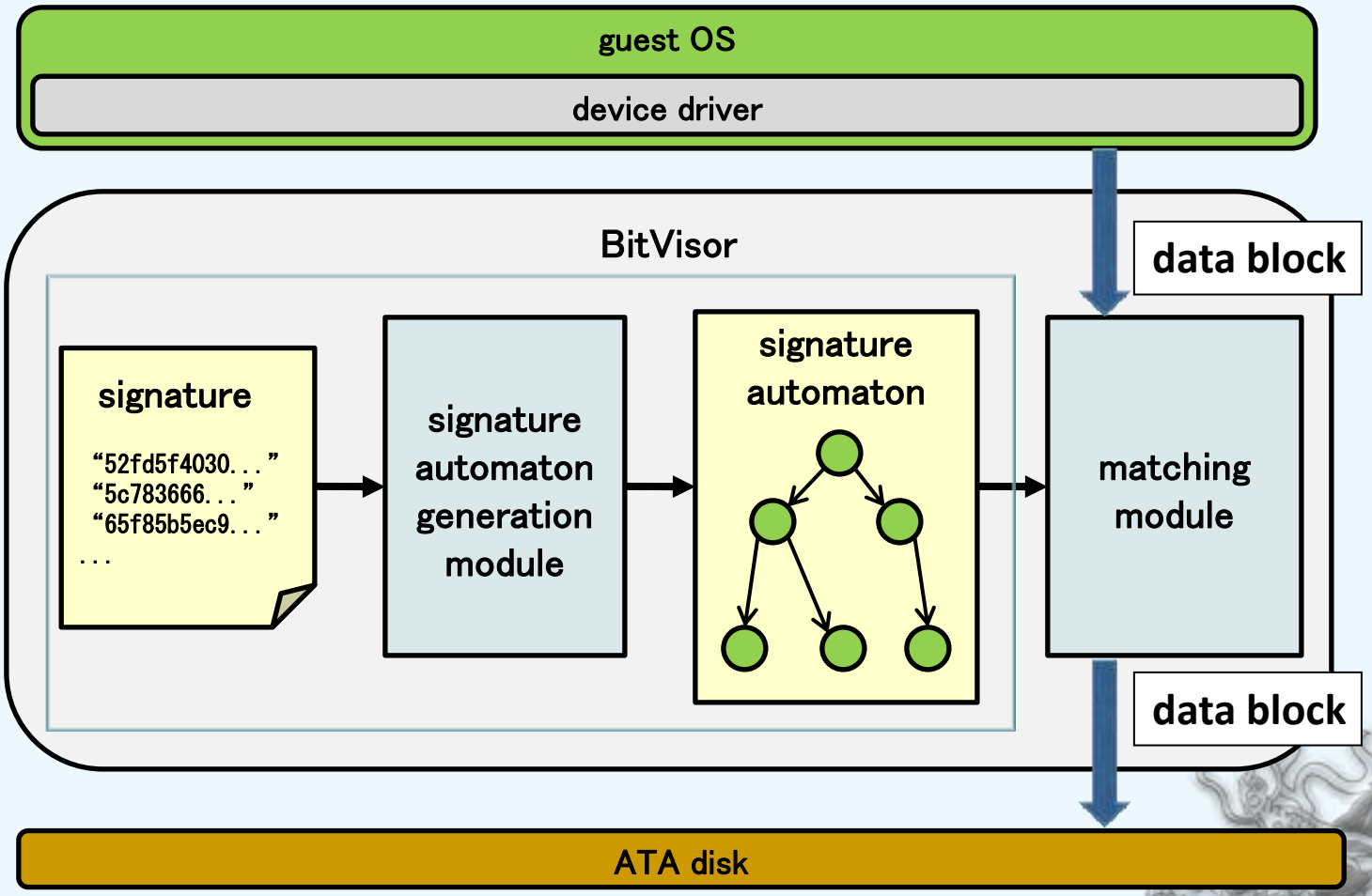
- Parapass-through (most accesses pass through, minimum accesses are mediated)
- Small TCB (only 20K+ lines of code as of 2009)



BVMD

- Detects malware signatures appearing in the I/O data of the guest OS
  - By intercepting I/O operations
- Contains a database of malware
  - Reuses the malware signature of free anti-malware software ClamAV
  - Signatures are embedded in the source code of BVMD
- Performs fast string matching using the Aho-Corasick algorithm

Structure


guest OS
device driver
BitVisor
data block
signature
"52fd5f4030..."
"5c783666..."
"65f85b5ec9..."
...
signature automaton generation module
signature automaton
matching module
data block
ATA disk

Conclusion

- We proposed a malware deactivation method
  - Mostly OS-independent
    - Because it changes only the behavior of virtual hardware
  - Process granularity
  - It cannot completely stop malware, but will be useful for the first mild countermeasure in IaaS context
- We proposed BVMD, a extension to BitVisor that can detect malware signature in a hypervisor layer
  - Malware detection in a host-OS-less hypervisor
  - We confirmed that BVMD could detect most of Windows/Linux malware in the wild we collected