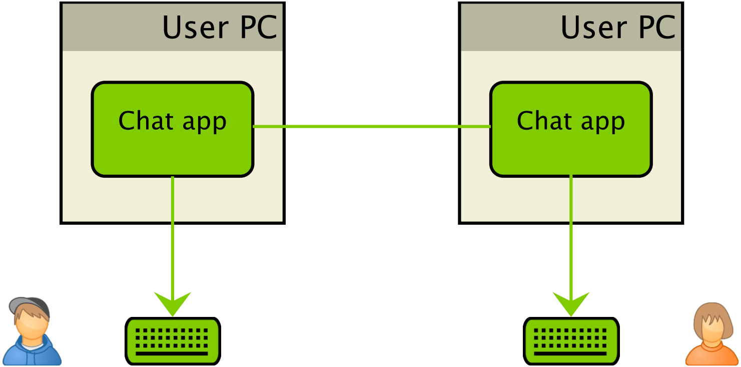
SCIENCE ▪ TECHNOLOGY ▪ PASSION

SGXIO: Generic Trusted I/O Path for Intel SGX

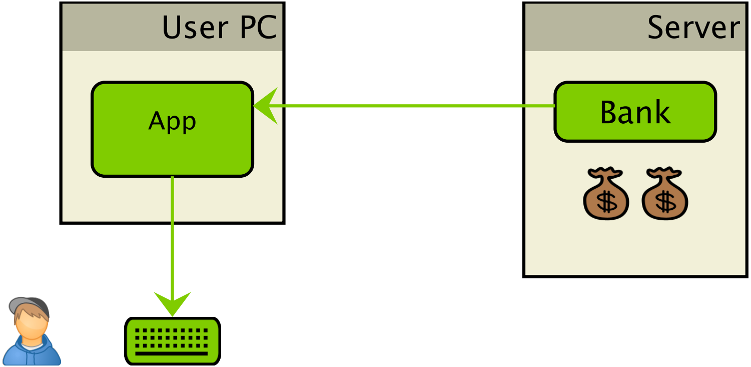
Samuel Weiser, Mario Werner,
Graz University of Technology - IAIK

March 23rd, 2017

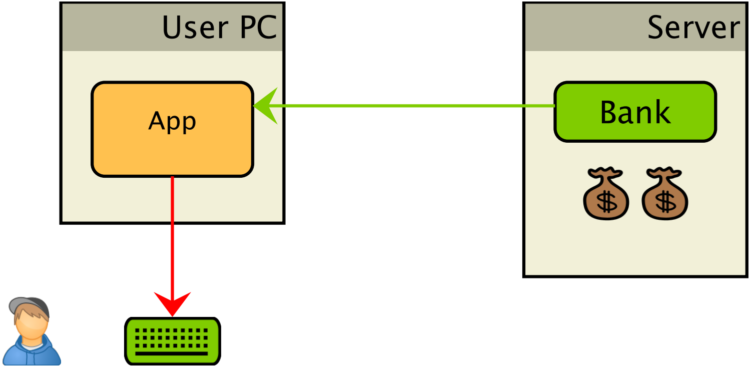
Application Scenario



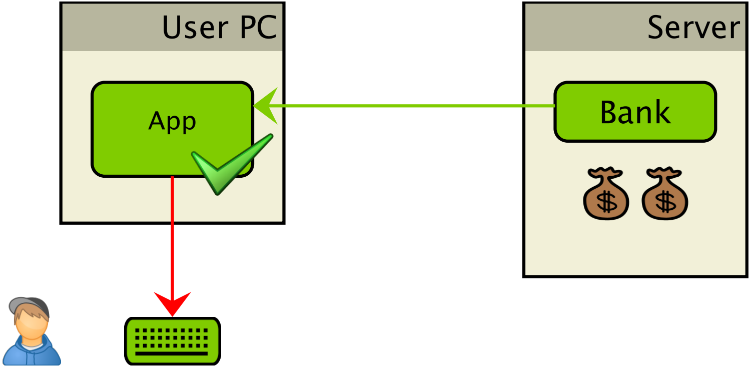
Application Scenario



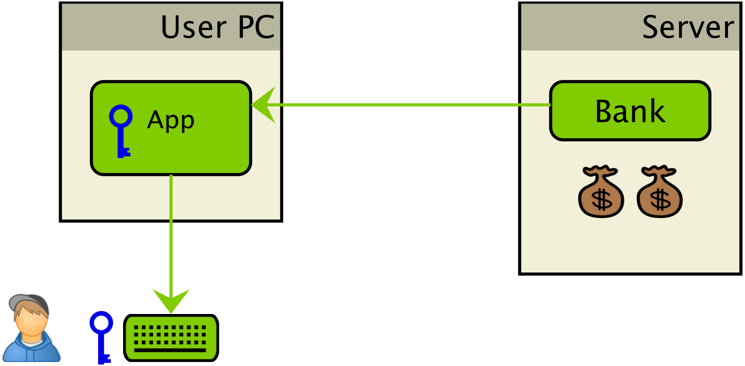
Current Situation



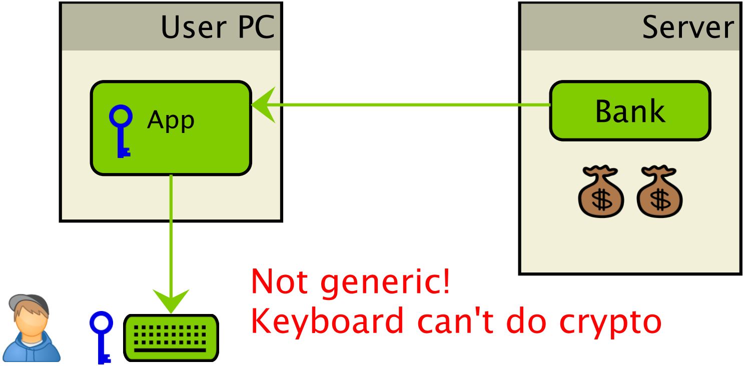
Intel SGX enhances app security



Trusted I/O path requires crypto



Trusted I/O path requires crypto



Not generic!
Keyboard can't do crypto

SGX does NOT support
Generic Trusted I/O Path!

SGXIO: Generic Trusted I/O Path for Intel SGX

SGXIO Architecture

- Conceptual work

SGXIO Architecture

- Conceptual work
  - Use **SGX** to protect user app

SGXIO Architecture

- Conceptual work
  - Use **SGX** to protect user app
  - Use **hypervisor** for trusted path [3]

SGXIO Architecture

- Conceptual work
  - Use **SGX** to protect user app
  - Use **hypervisor** for trusted path [3]
  - Use Trusted Platform Module (**TPM**) for verifying hypervisor

SGXIO Architecture

- Conceptual work
  - Use **SGX** to protect user app
  - Use **hypervisor** for trusted path [3]
  - Use Trusted Platform Module (**TPM**) for verifying hypervisor
  - Bind security domains of **SGX** and **TPM**

SGXIO Architecture

- Conceptual work
  - Use **SGX** to protect user app
  - Use **hypervisor** for trusted path [3]
  - Use Trusted Platform Module (**TPM**) for verifying hypervisor
  - Bind security domains of **SGX** and **TPM**
  - Make enclaves context-aware (enclave virtualization attacks)

SGXIO Architecture

- Conceptual work
  - Use **SGX** to protect user app
  - Use **hypervisor** for trusted path [3]
  - Use Trusted Platform Module (**TPM**) for verifying hypervisor
  - Bind security domains of **SGX** and **TPM**
  - Make enclaves context-aware (enclave virtualization attacks)

→ Achieve trusted path for SGX

SGXIO Architecture

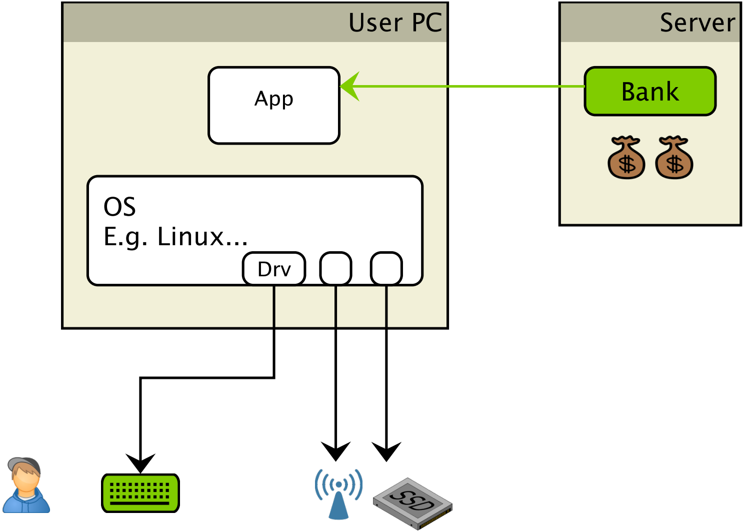
- Conceptual work
  - Use **SGX** to protect user app
  - Use **hypervisor** for trusted path [3]
  - Use Trusted Platform Module (**TPM**) for verifying hypervisor
  - Bind security domains of **SGX** and **TPM**
  - Make enclaves context-aware (enclave virtualization attacks)

→ Achieve trusted path for SGX

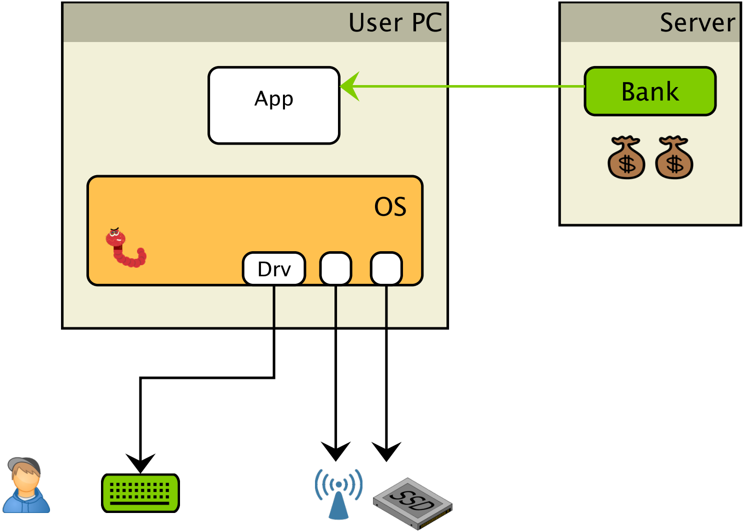
→ Support verification of the trusted path

Why do we need SGX?

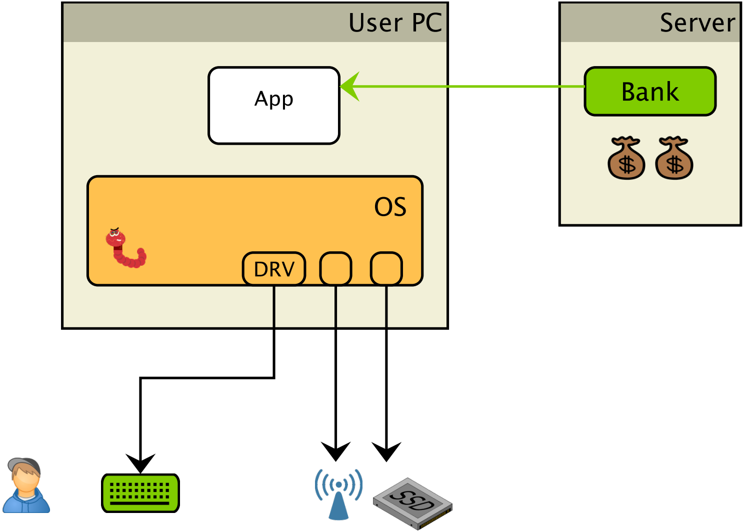
Setup: Commodity Operating System (OS)



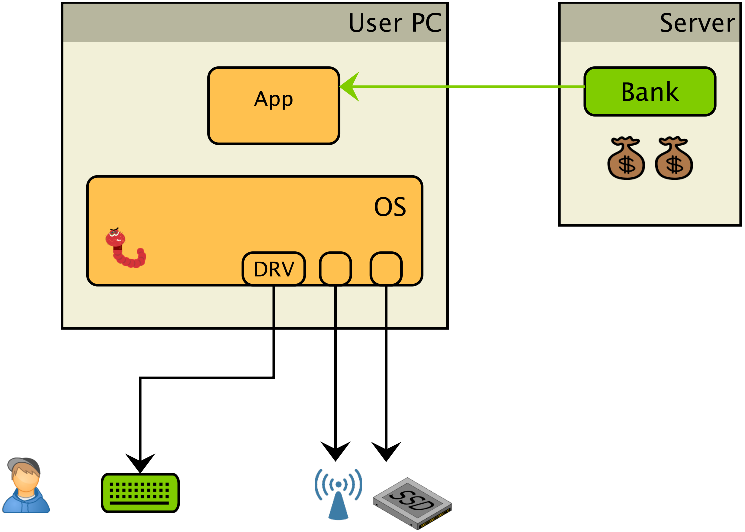
OS is untrusted



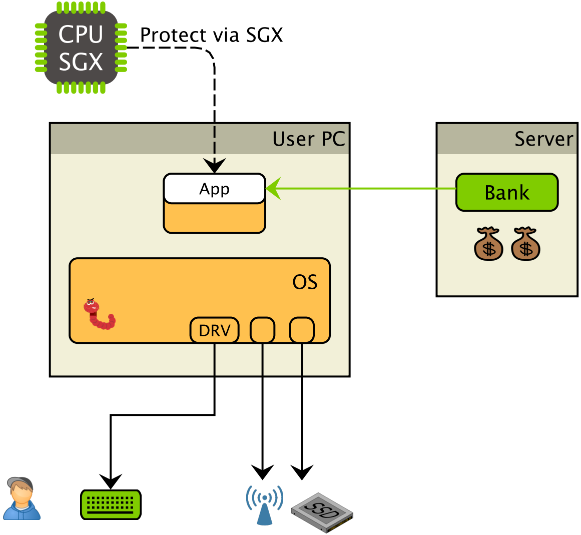
Driver is untrusted



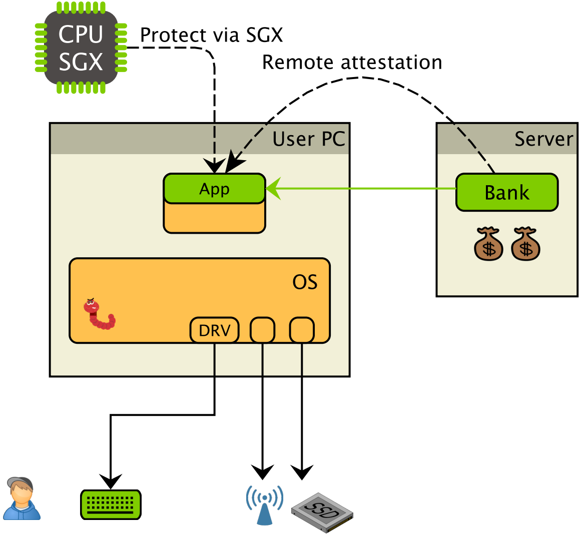
App is untrusted



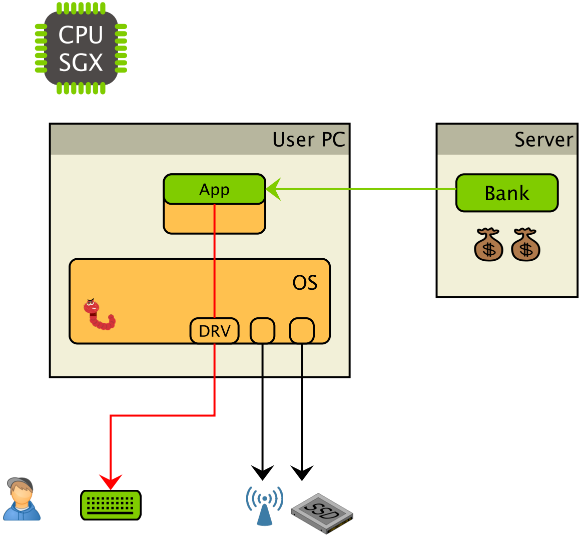
Protect app with SGX



Verify app with SGX

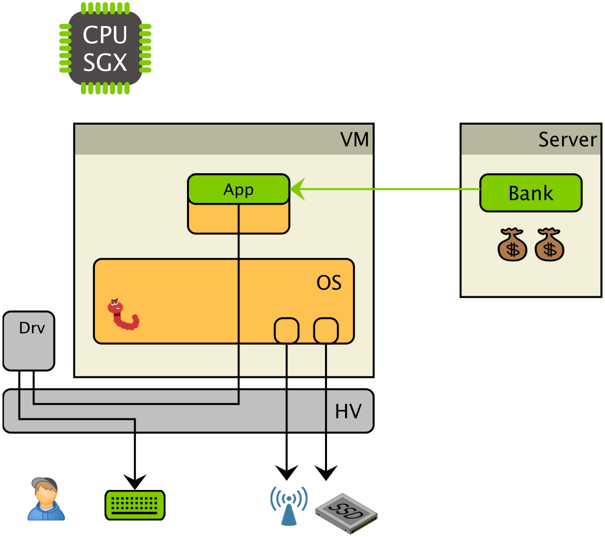


We want trusted path to user

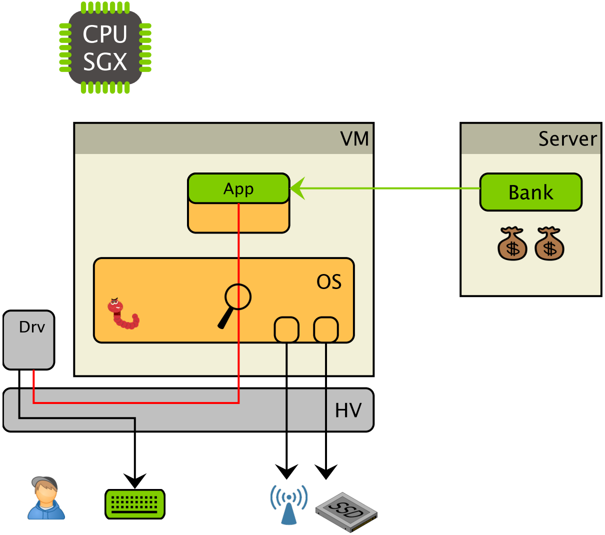


SGXIO

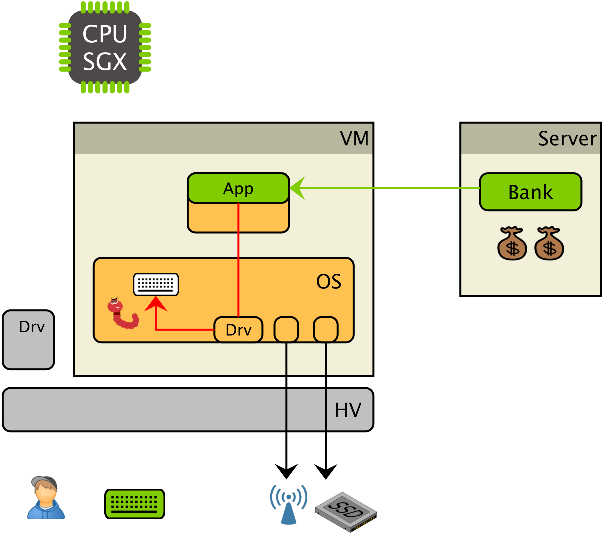
Isolate driver with hypervisor



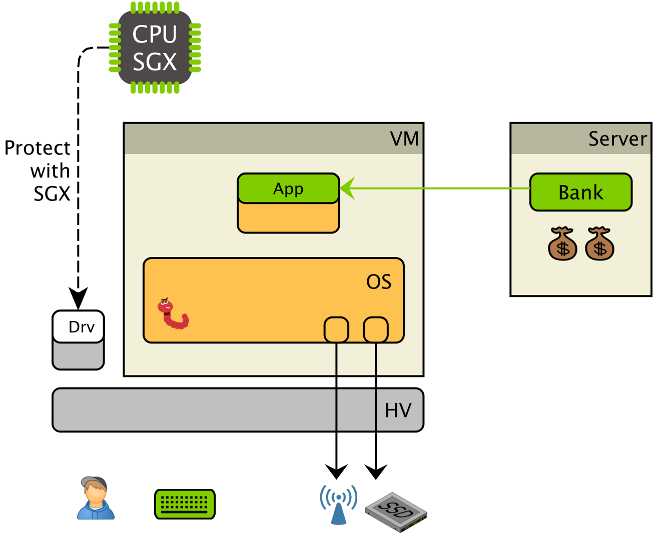
OS can intercept trusted path



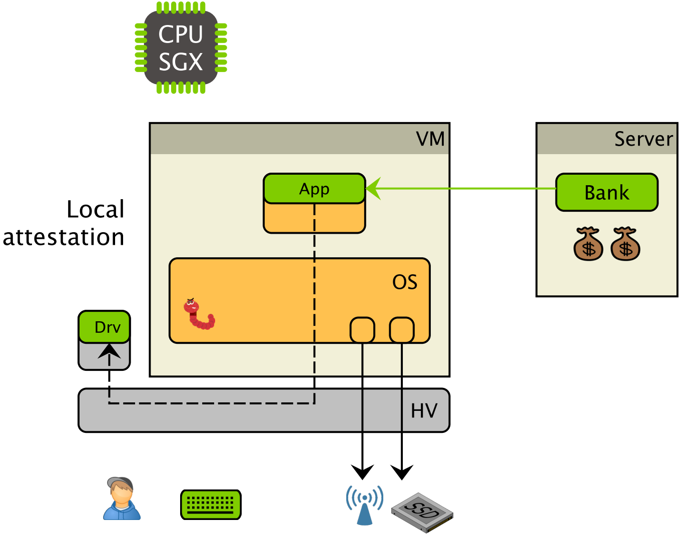
OS can intercept trusted path



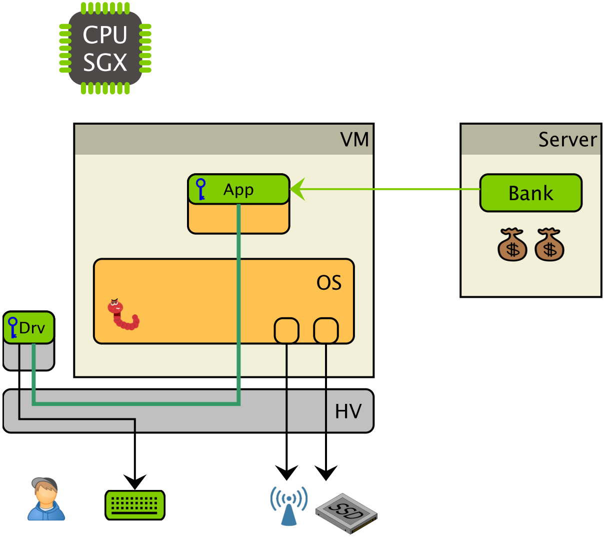
Run driver in SGX enclave



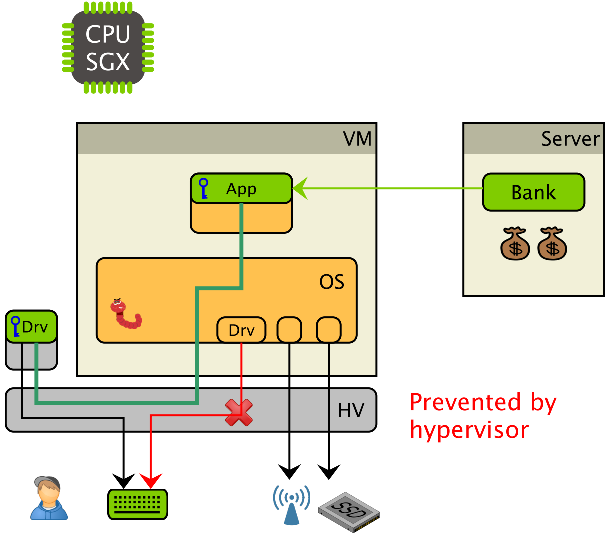
Run driver in SGX enclave



Encrypt trusted path

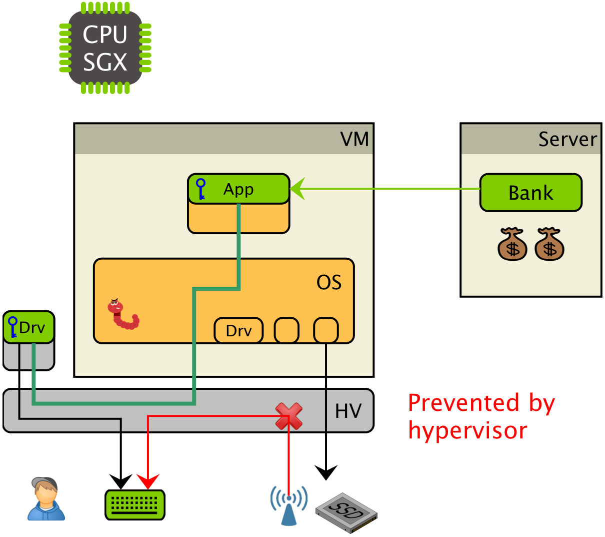


Isolate user device with Hypervisor (HV)



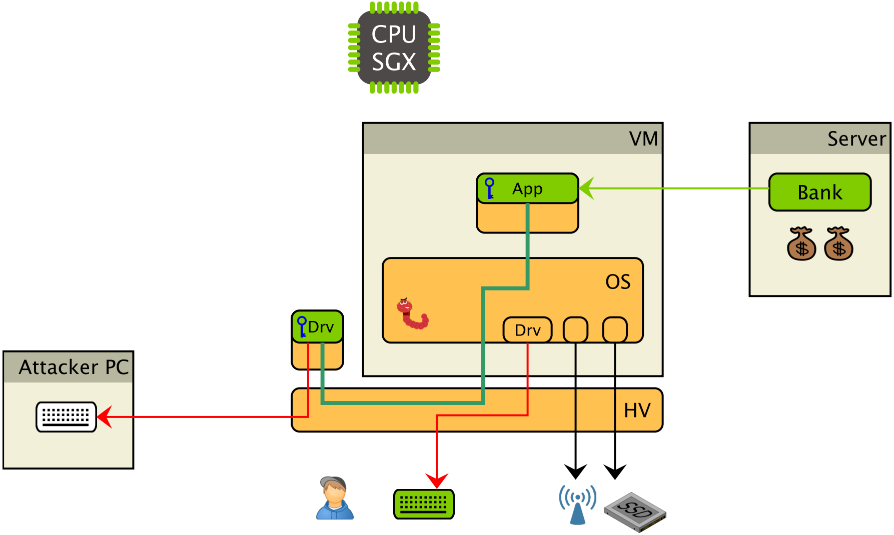
Prevented by hypervisor

Isolate user device with Hypervisor (HV)

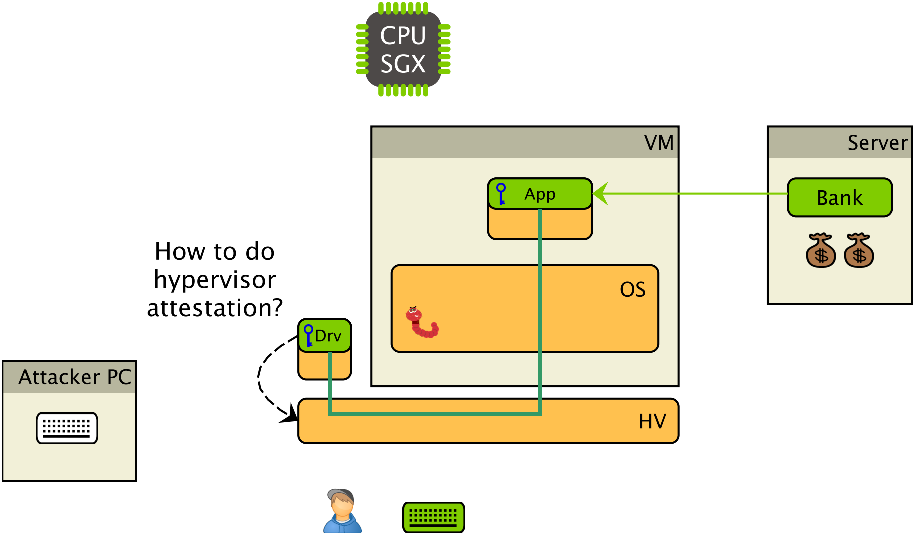


Prevented by hypervisor

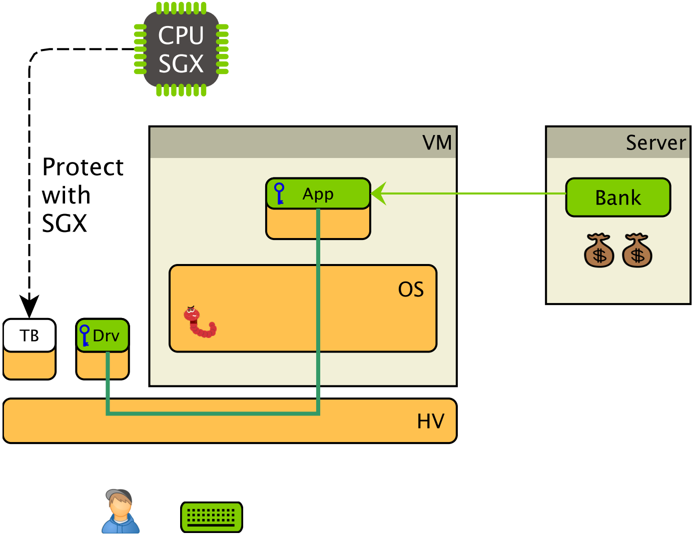
Compromised HV can intercept trusted path



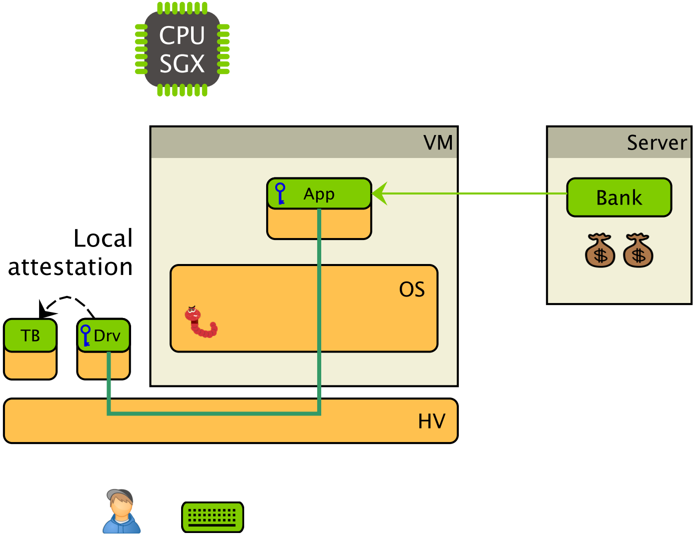
Hypervisor attestation required



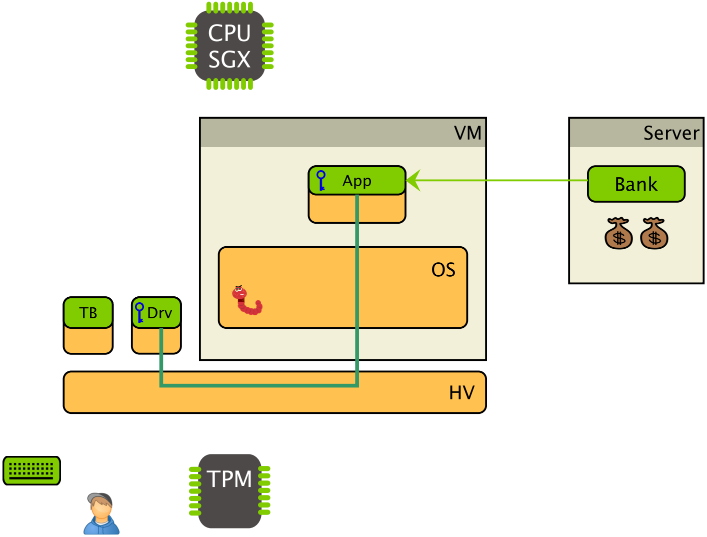
Trusted Boot (TB) Enclave



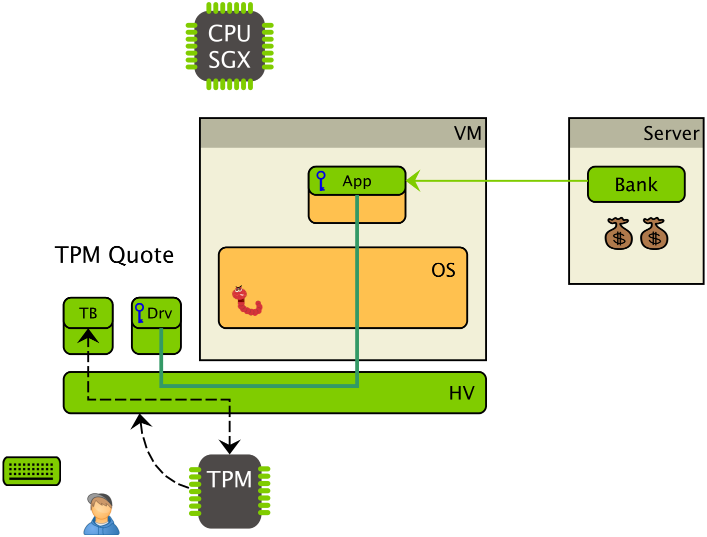
Trusted Boot (TB) Enclave



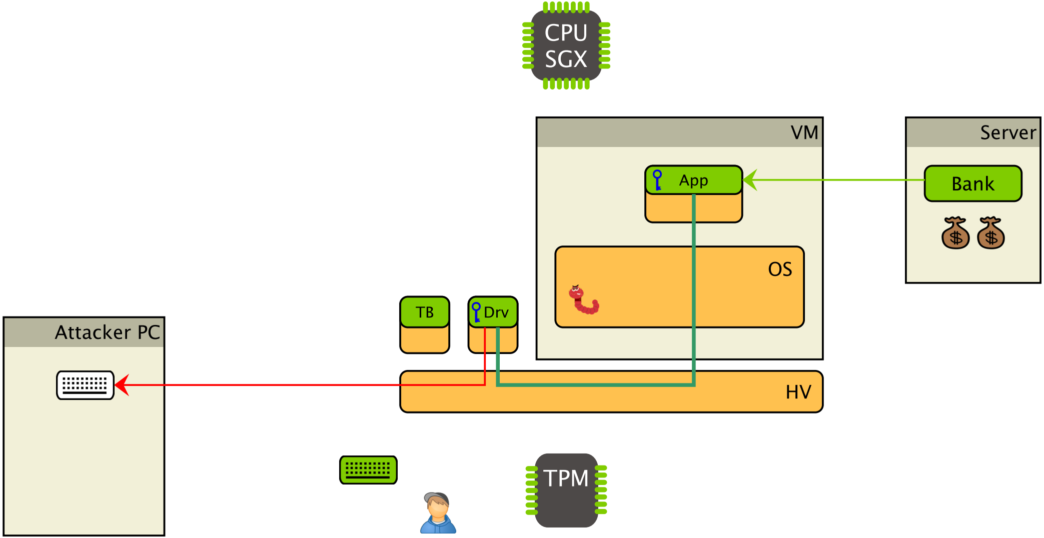
Trusted Platform Module (TPM)



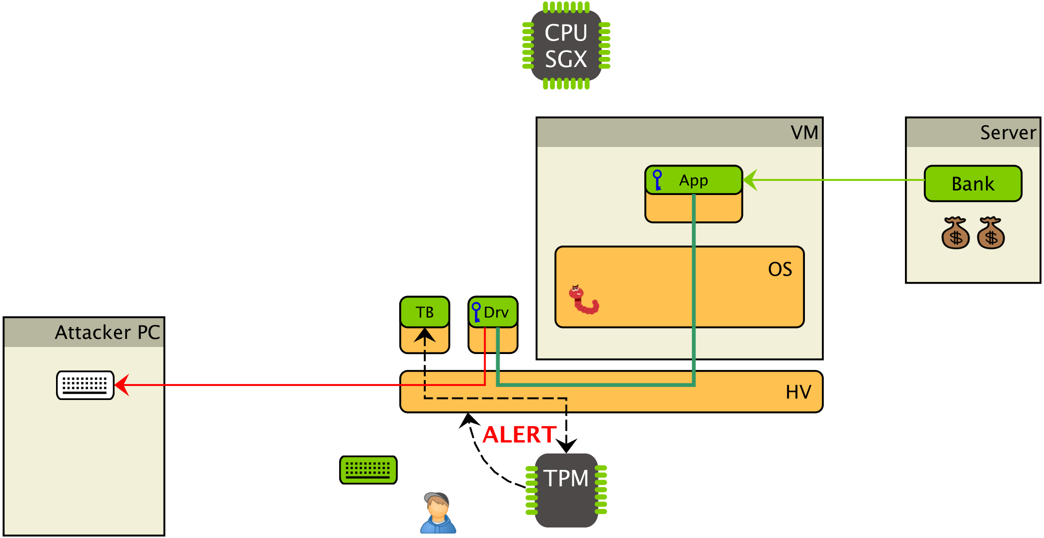
Trusted Boot



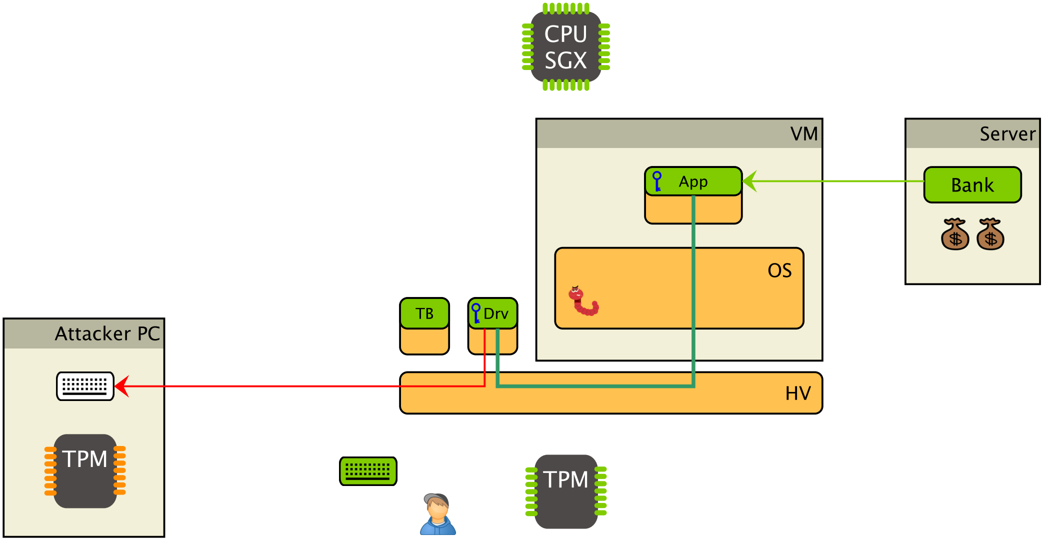
Can HV compromise be detected?



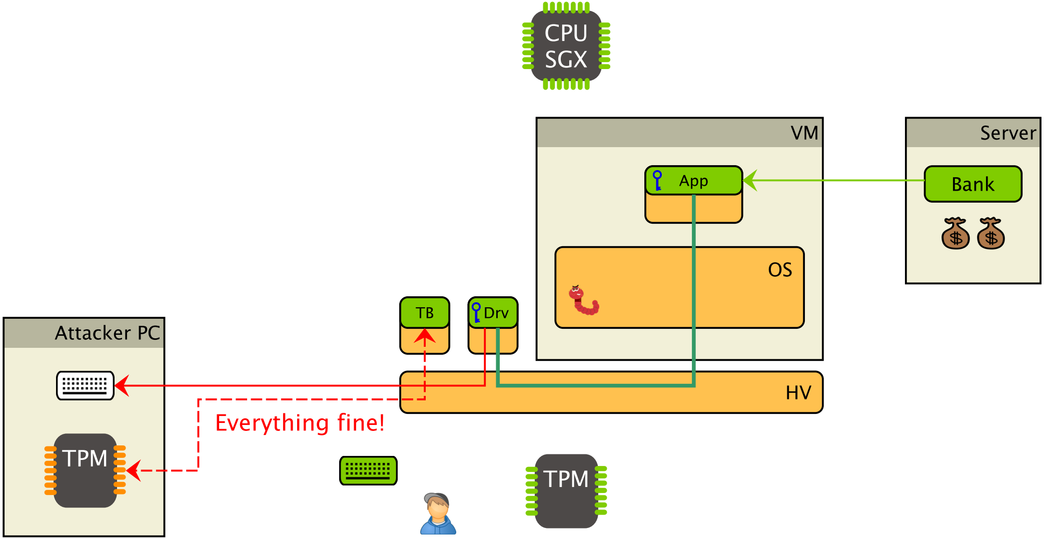
Can HV compromise be detected? Yes



Wait! Remote TPM attack (cuckoo attack)



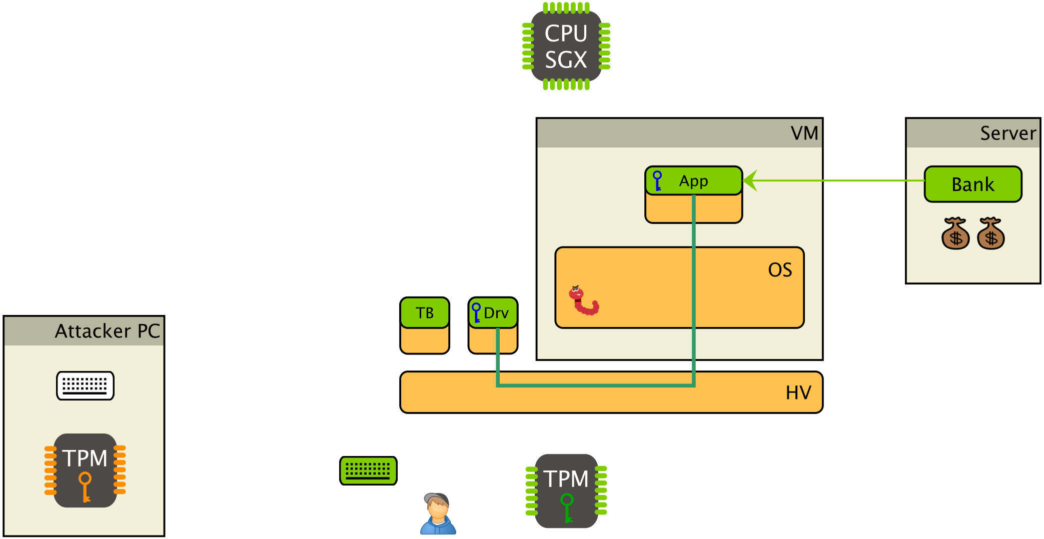
Wait! Remote TPM attack (cuckoo attack)



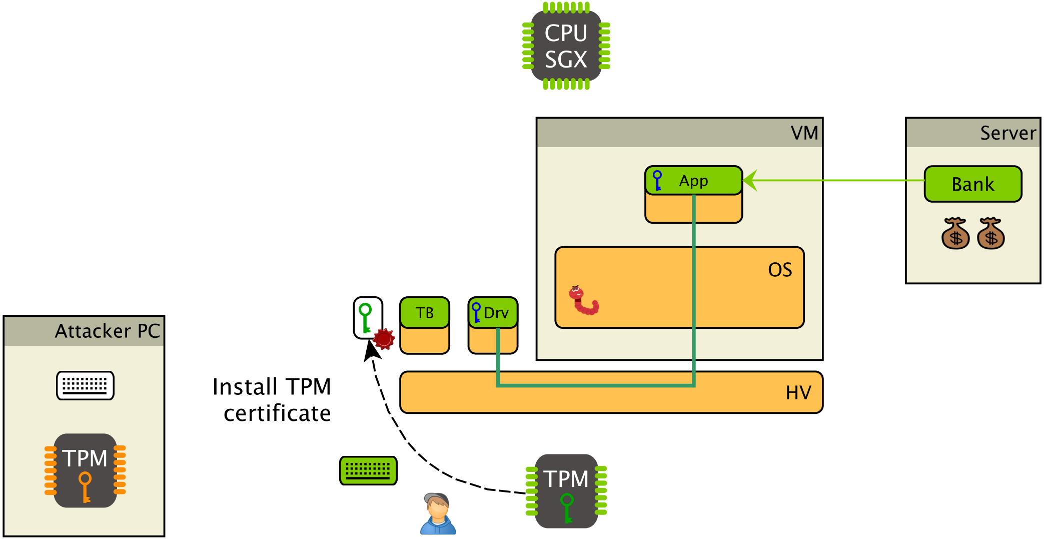
TB enclave needs to know correct TPM!

We need a domain binding between SGX and the TPM

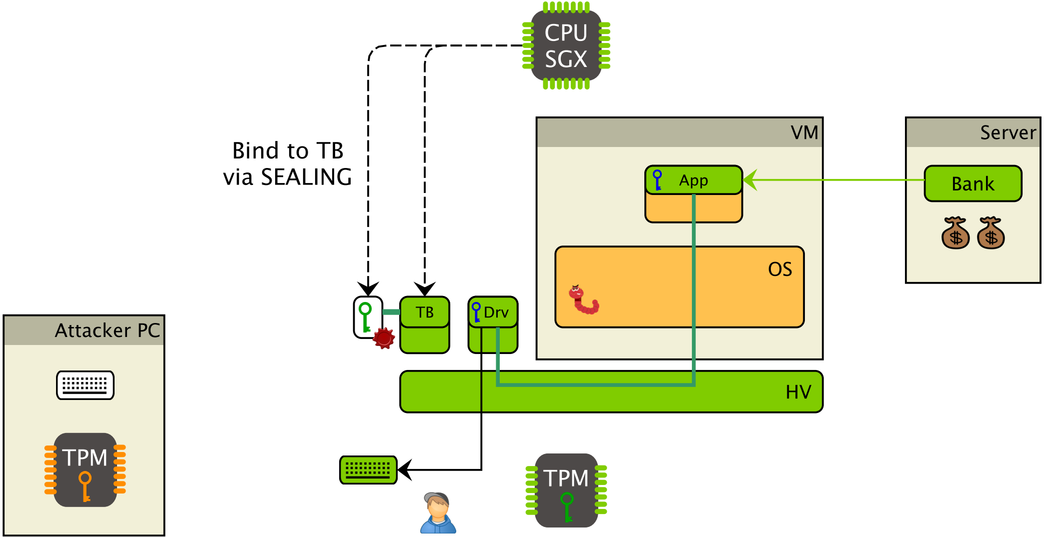
TPM Attestation Key



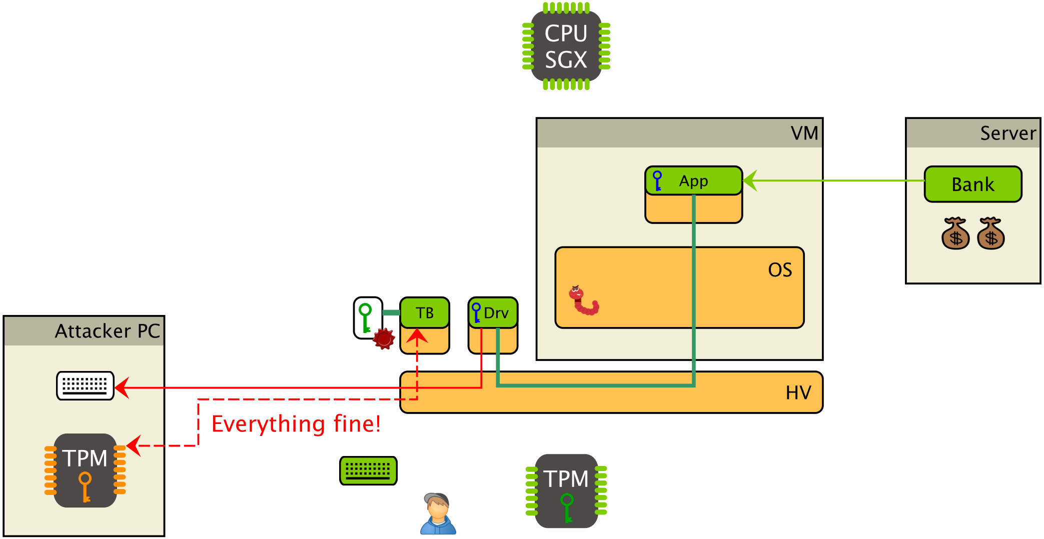
Install TPM certificate



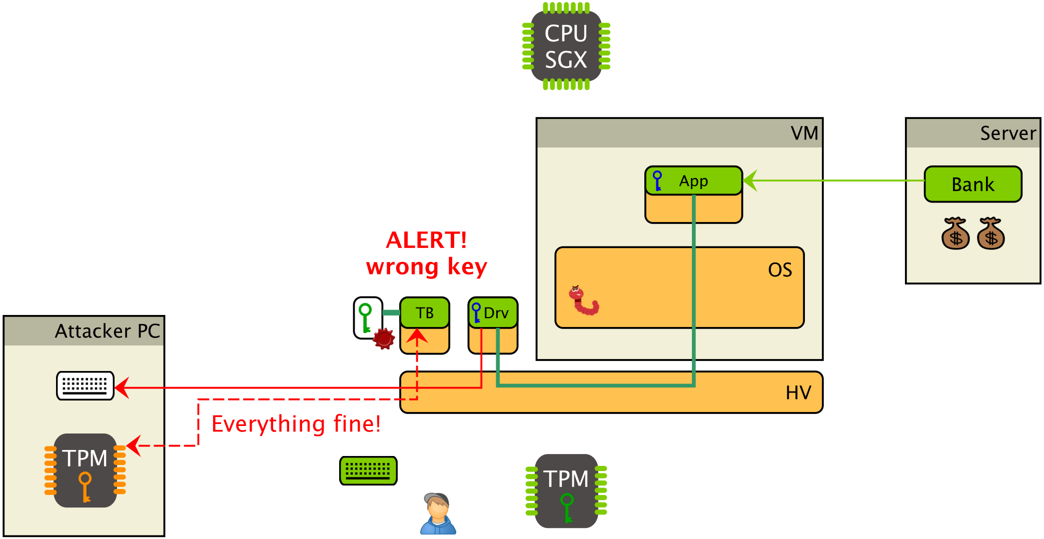
Seal TPM certificate



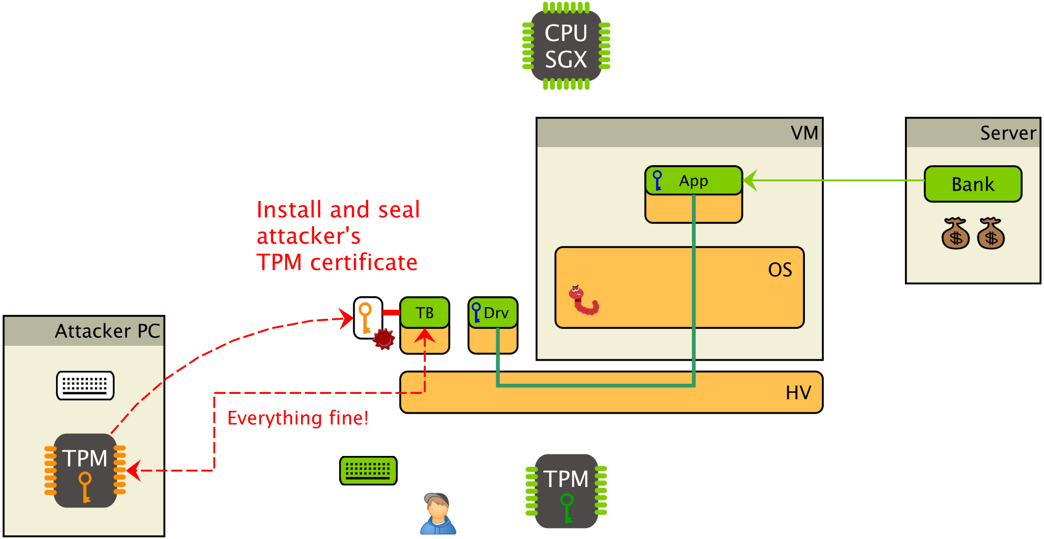
Are remote TPM attacks thwarted?



Are remote TPM attacks thwarted? Yes



Wait! Just install attacker's TPM certificate



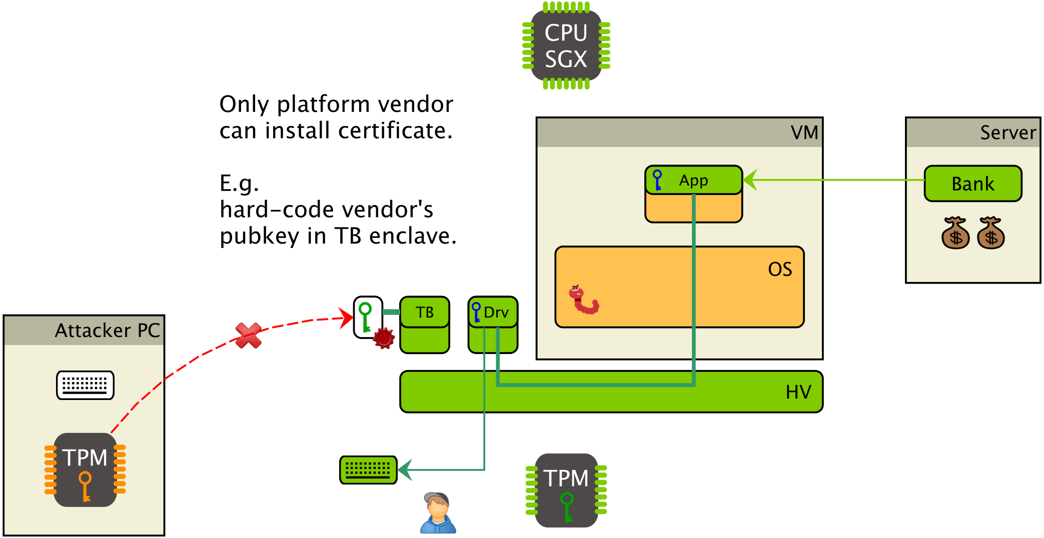
Install and seal attacker's TPM certificate

Everything fine!

Certificate installation policy required

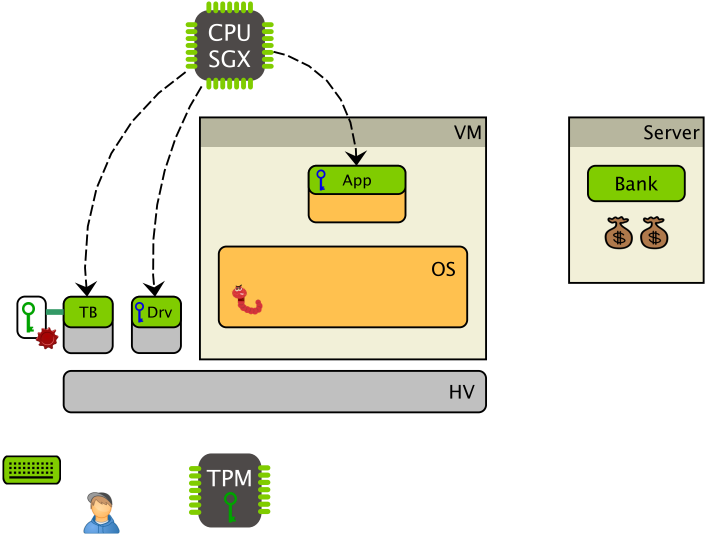
Only platform vendor can install certificate.

E.g.
hard-code vendor's pubkey in TB enclave.

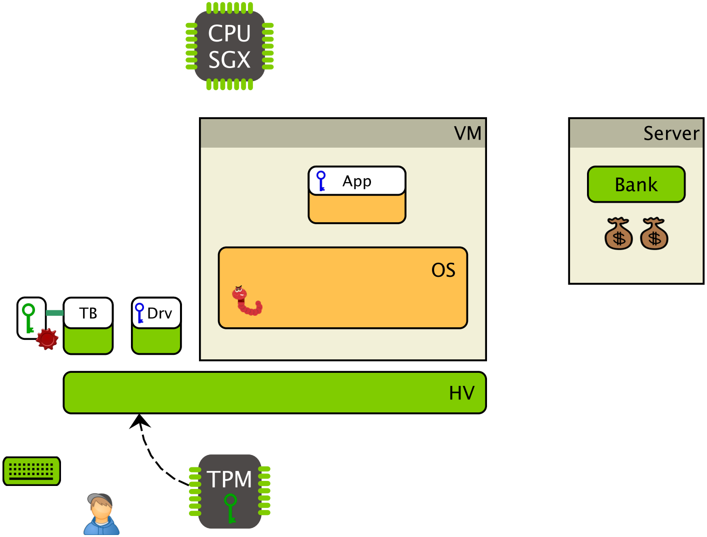


What did we achieve so far?

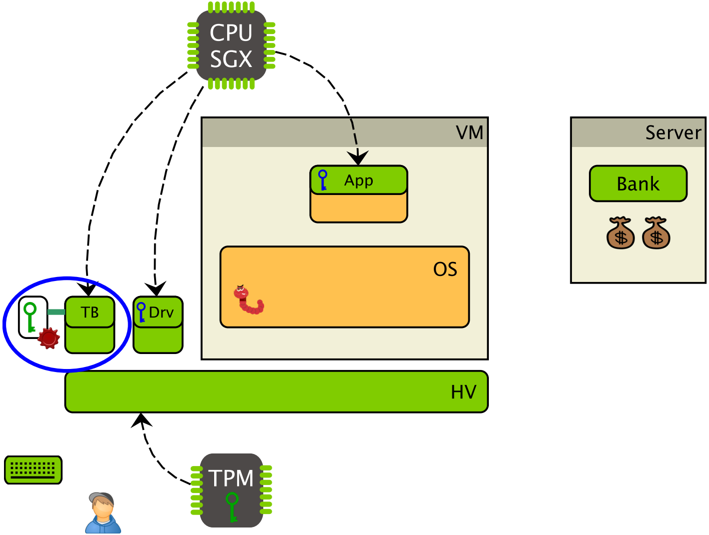
Recap: SGX protects enclaves



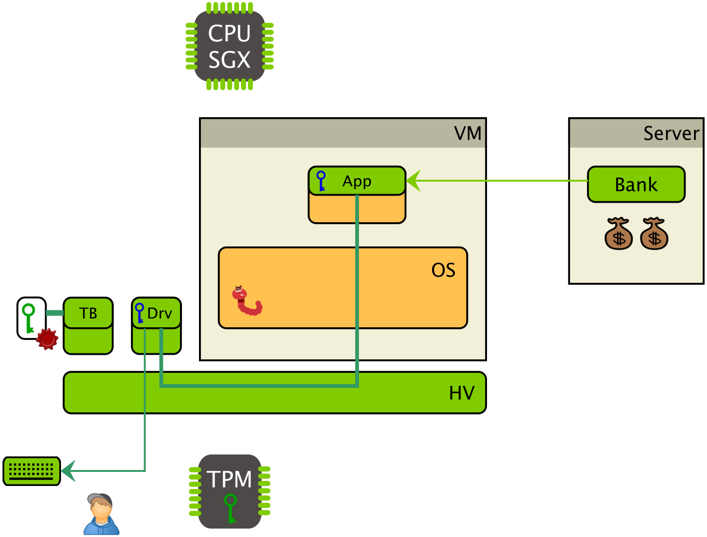
Recap: TPM attests hypervisor



We achieved Domain Binding: SGX — TPM



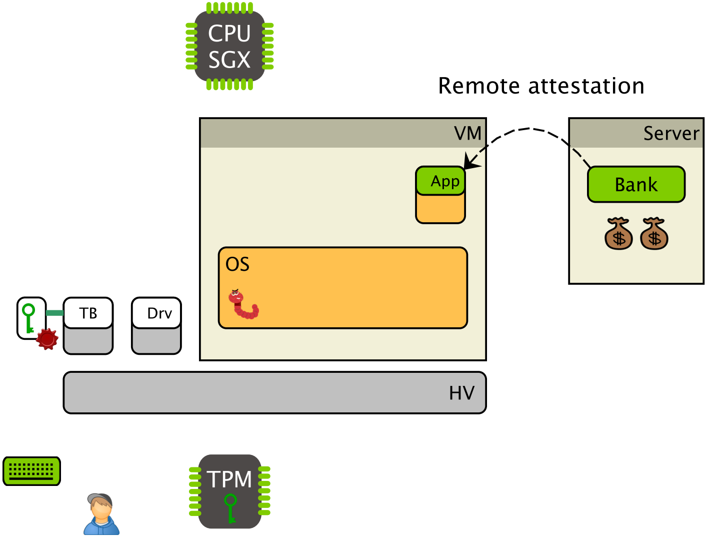
We achieved attestable trusted path



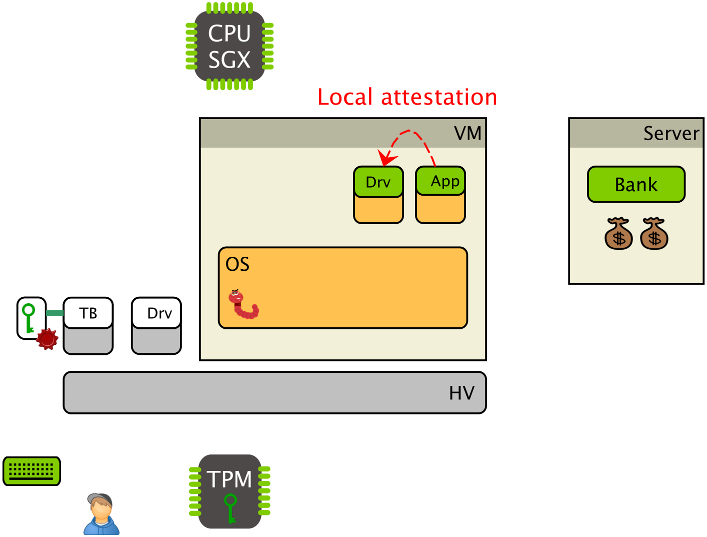
Well, almost...

Enclave Virtualization Attacks

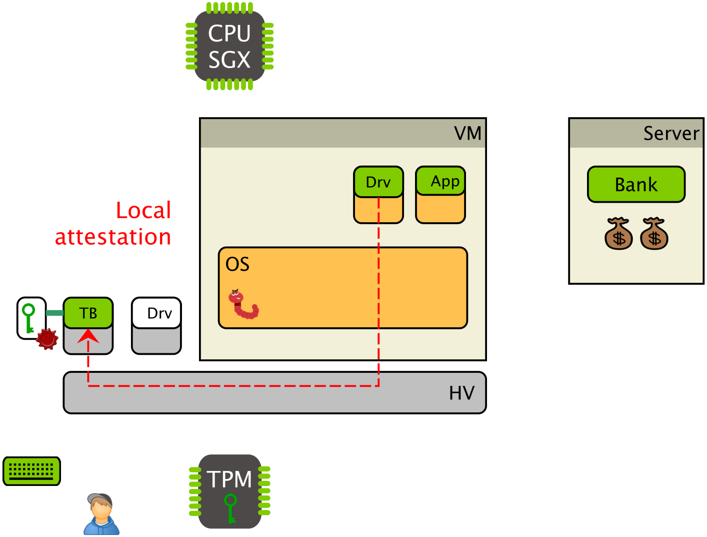
Driver Enclave Virtualization Attack



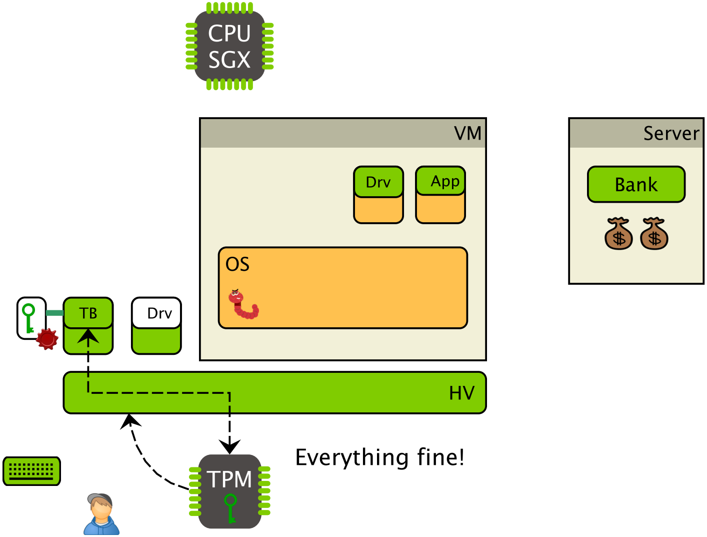
Driver Enclave Virtualization Attack



Driver Enclave Virtualization Attack

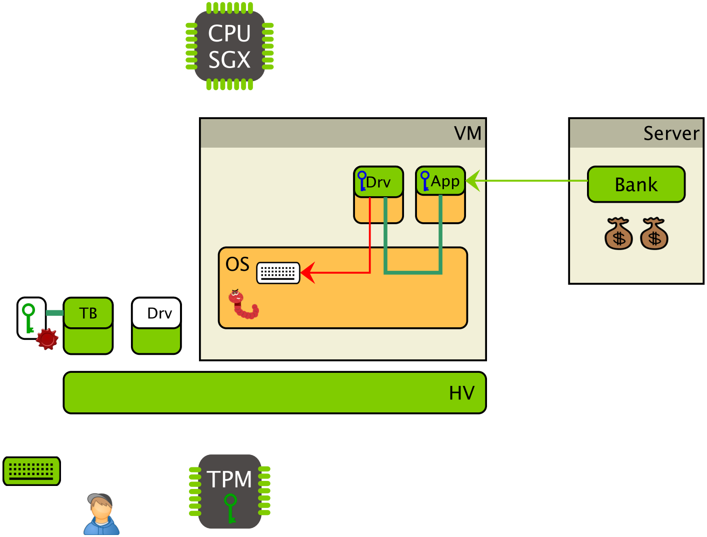


Driver Enclave Virtualization Attack

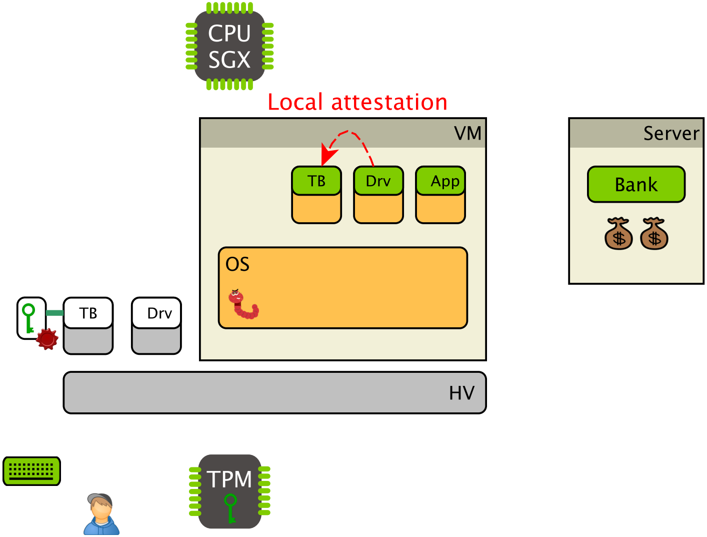


Everything fine!

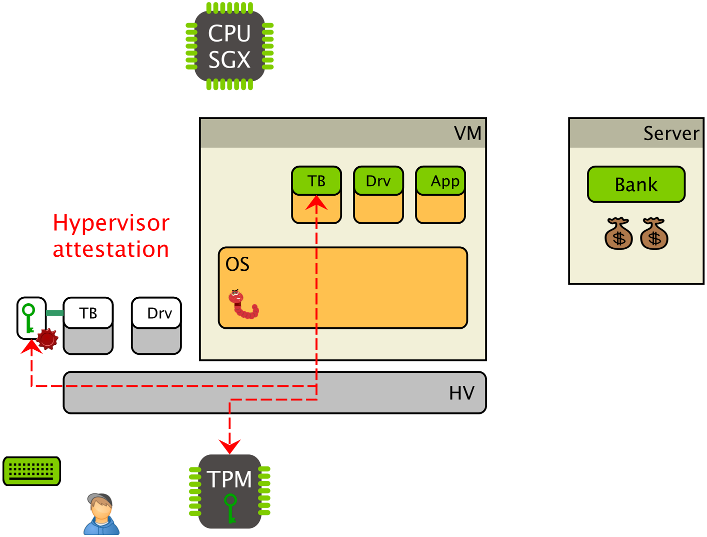
Driver Enclave Virtualization Attack



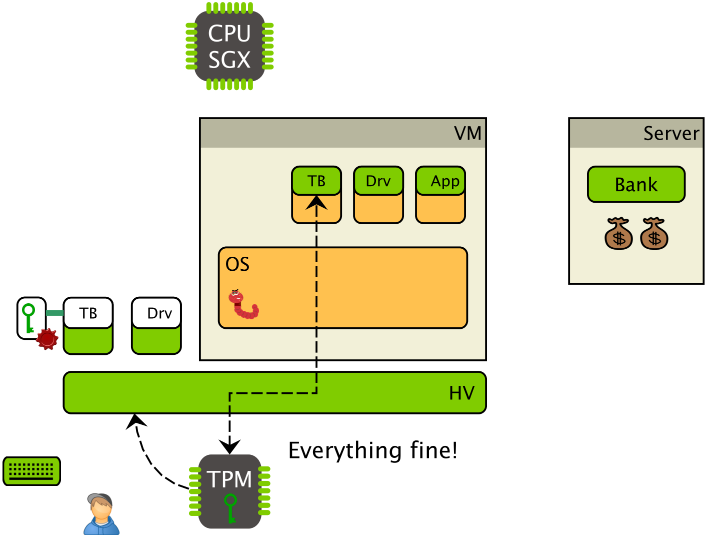
TB Enclave Virtualization Attack



TB Enclave Virtualization Attack

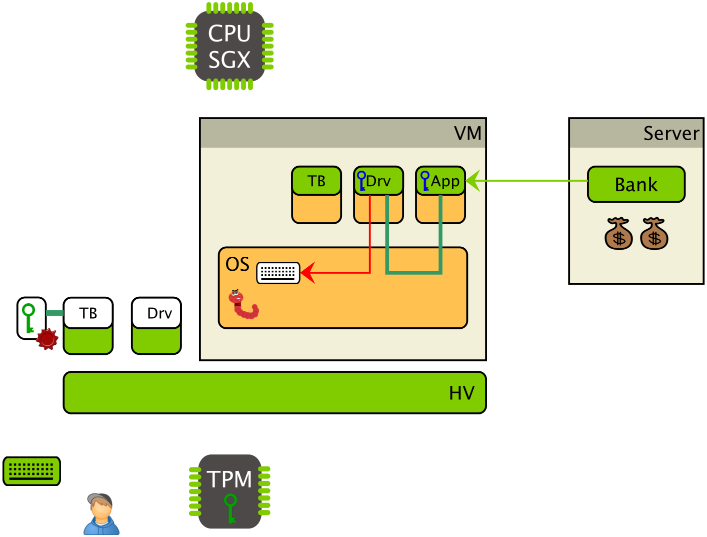


TB Enclave Virtualization Attack



Everything fine!

TB Enclave Virtualization Attack



Making enclaves context-aware

Problem:

- Enclaves do not know their execution context
- Driver/TB Enclave cannot detect virtualization

Making enclaves context-aware

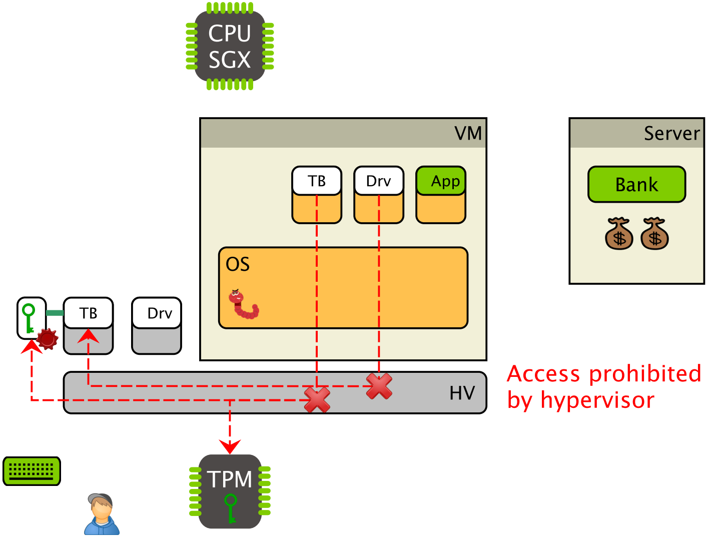
Problem:

- Enclaves do not know their execution context
- Driver/TB Enclave cannot detect virtualization

Solution:

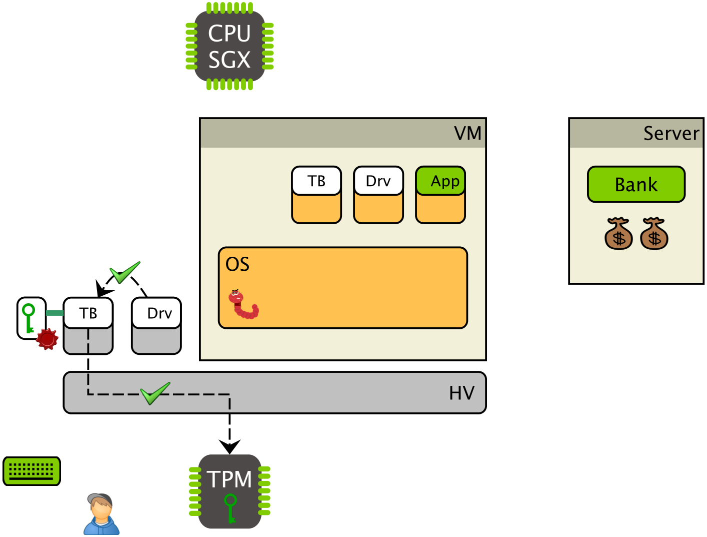
- Hypervisor knows enclave context
- Hypervisor isolates legitimate TB enclave and TPM from OS

Making enclaves context-aware



Access prohibited by hypervisor

Making enclaves context-aware



Summary: SGXIO Requirements

- App and untrusted OS inside a VM

Summary: SGXIO Requirements

- App and untrusted OS inside a VM
- Driver outside this VM

Summary: SGXIO Requirements

- App and untrusted OS inside a VM
- Driver outside this VM
- Hypervisor isolating driver and user device from VM

Summary: SGXIO Requirements

- App and untrusted OS inside a VM
- Driver outside this VM
- Hypervisor isolating driver and user device from VM
- TPM for trusted boot

Summary: SGXIO Requirements

- App and untrusted OS inside a VM
- Driver outside this VM
- Hypervisor isolating driver and user device from VM
- TPM for trusted boot
- Strong binding between TPM and TB Enclave

Summary: SGXIO Requirements

- App and untrusted OS inside a VM
- Driver outside this VM
- Hypervisor isolating driver and user device from VM
- TPM for trusted boot
- Strong binding between TPM and TB Enclave
    - Certificate installation policy

Summary: SGXIO Requirements

- App and untrusted OS inside a VM
- Driver outside this VM
- Hypervisor isolating driver and user device from VM
- TPM for trusted boot
- Strong binding between TPM and TB Enclave
  - Certificate installation policy
- Hypervisor isolating TB enclave and TPM from VM

More Topics

- User verification
- Choice of hypervisor
- Driver and app design
- Intel PAVP, Intel Insider
- Fast & lightweight key exchange with SGX local attestation

→ See paper [1, 2]

More Topics

- User verification
- Choice of hypervisor
- Driver and app design
- Intel PAVP, Intel Insider
- Fast & lightweight key exchange with SGX local attestation

→ See paper [1, 2]

- PCI device isolation [3]
- Hardware I/O support for enclaves [4]

SCIENCE ▪ TECHNOLOGY ▪ PASSION

SGXIO: Generic Trusted I/O Path for Intel SGX

Samuel Weiser, Mario Werner,
Graz University of Technology - IAIK

March 23rd, 2017

References

[1] Samuel Weiser and Mario Werner.
SGXIO: Generic Trusted I/O Path for Intel SGX.
arXiv:1701.01061, January 2017.

[2] Samuel Weiser and Mario Werner.
SGXIO: Generic Trusted I/O Path for Intel SGX.
In *CODASPY'17*, 2017.

[3] Z. Zhou, V. D. Gligor, J. Newsome, and J. M. McCune.
Building Verifiable Trusted Path on Commodity x86 Computers.
In *SP'12*, pages 616–630, May 2012.

[4] Samuel Weiser.
Secure I/O with Intel SGX.
Master's thesis, Graz University of Technology, 2016.
`https://pure.tugraz.at/portal/files/7516934/2016_Weiser_Thesis_SecureIO_SGX.pdf`.