DeviceDisEnabler: a lightweight hypervisor which hides devices to protect cyber espionage and tampering

Kuniyasu Suzaki

National Institute of Advanced Industrial Science and Technology(AIST)

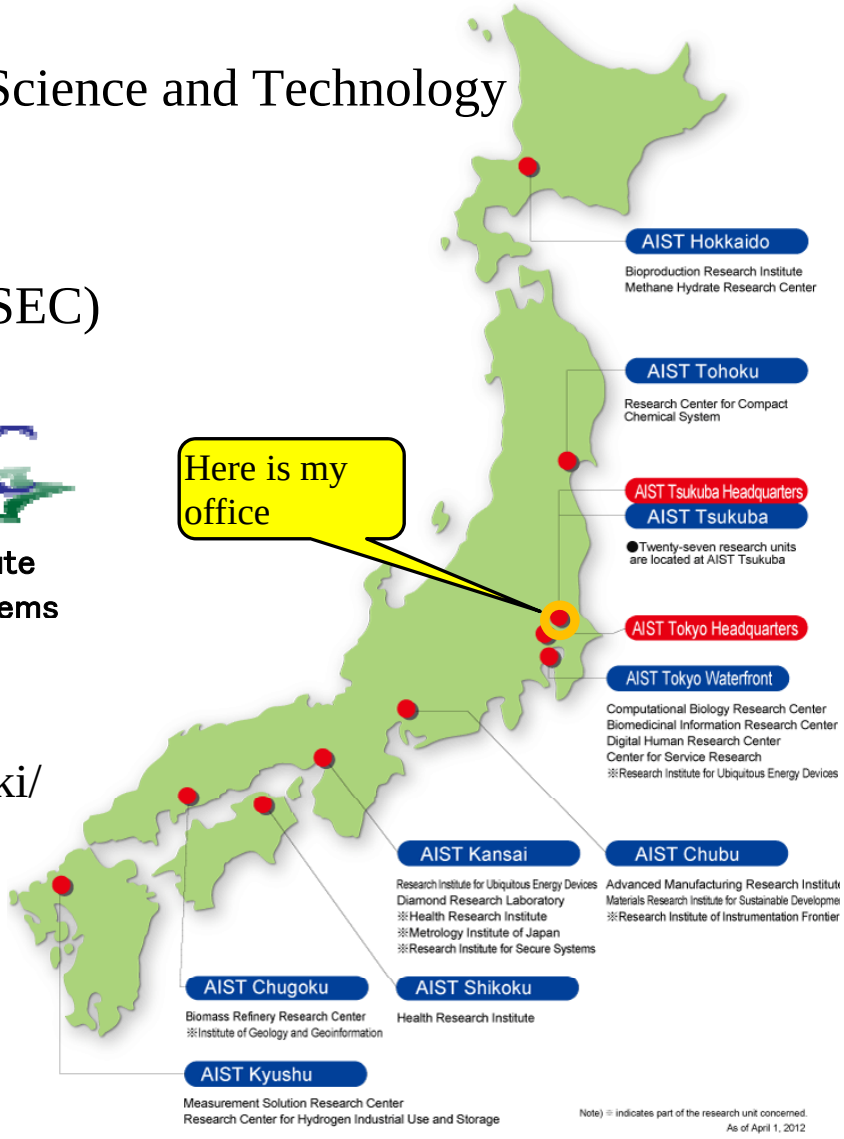
Black Hat Sao Paul, Brazil, 26/November/2014

Who am I?

- A researcher for computer security
  - National Institute of Advanced Industrial Science and Technology (AIST)
    - More than 3,000 researchers.
  - Research Institute for Secure Systems (RISEC)
    - About 40 researchers for security

National Institute of Advanced Industrial Science and Technology
AIST

RISEC
Research Institute for Secure Systems



  - My office is at Tsukuba headquarter.
- Current interests https://staff.aist.go.jp/k.suzaki/
  - Security on hypervisor
  - Security on control systems

Do you know how many devices included in a mobile gadget?

- Microphone, Speaker
- Digital Camera
- GPS
- Gyroscope
- etc. (Many sensors)

- Around 2000, PDA(e.g., Palm Pilot, Apple Newton) did not have such devices.
- CURRENT mobile gadgets are not traditional computers. They are an aggregation of sensor devices.

Do you know the resolution of these devices?

- Microphone, Speaker
  - More than CD quality (44.1 kHz).
- Digital camera
  - More than 100M pixel.
- GPS
  - Resolution is less than 10 m.
- Gyroscope
  - Sampling is more than 20 Hz.

Is there anything wrong with these high resolution devices?

- Yes!
- High-resolution devices can be used for cyber espionage.
  - There are many incidents and research results.

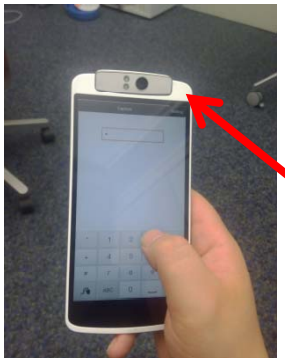
Eavesdropping caused by microphone

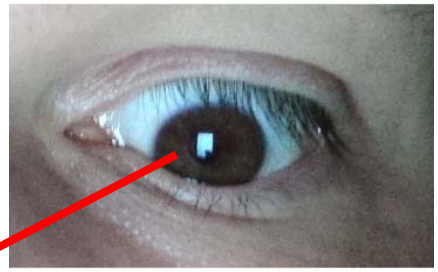
- “Bundestrojaner” (Federal Trojan) had high impact for society.
  - It is also named “R2D2” because the code has the string "C3PO-r2d2-POE".

- Allegedly, the malware R2D2 was installed by an officer at a German Airport.
  - R2D2 records Skype audio conversations and sends the data to a remote website.
  - R2D2 was discovered by Chaos Computer Club (CCC) in 2011.
- Finally, it was publicized that German authorities ordered the cyber espionage malware.

Facial Reflection Keylogger

[T.Fiebig, WOOT'14]



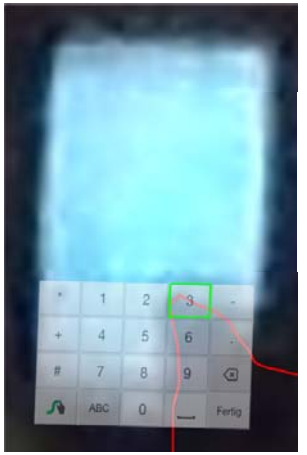


The front camera takes shot of user's face (eye).

Zooming

Detect thumb

Put on a keyboard



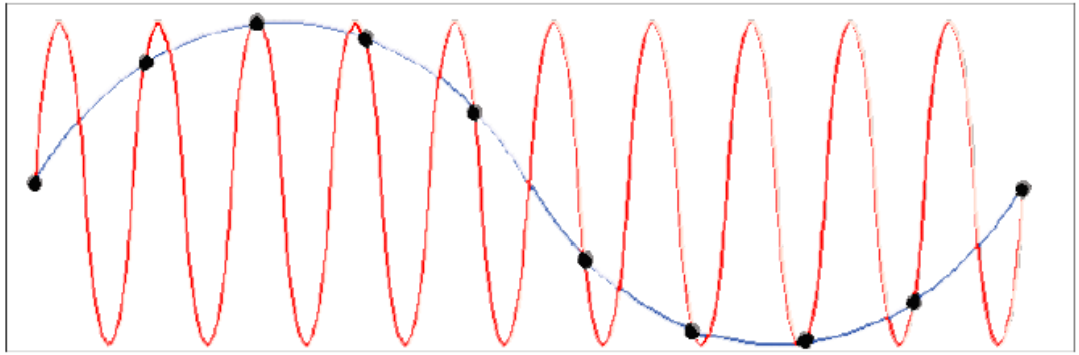
T.fiebig, j.krissler and r.hanesch, "Security Impact of High Resolution Smartphone Cameras" woot 2014.
https://www.usenix.org/conference/woot14/workshop-program/presentation/fiebig

Malicious location tracking by GPS

- “Cerberus” and “mSpy” are normal applications (anti-theft application), but they are used to track employee.
- Japanese application named “karelog” (Boyfriend Log) steals data of GPS without permission.
  - It was sold by the name of “GPS Control manager”.
  - It became the social problem in Japan and the company had to terminate the service.

Eavesdropping caused by Gyroscope

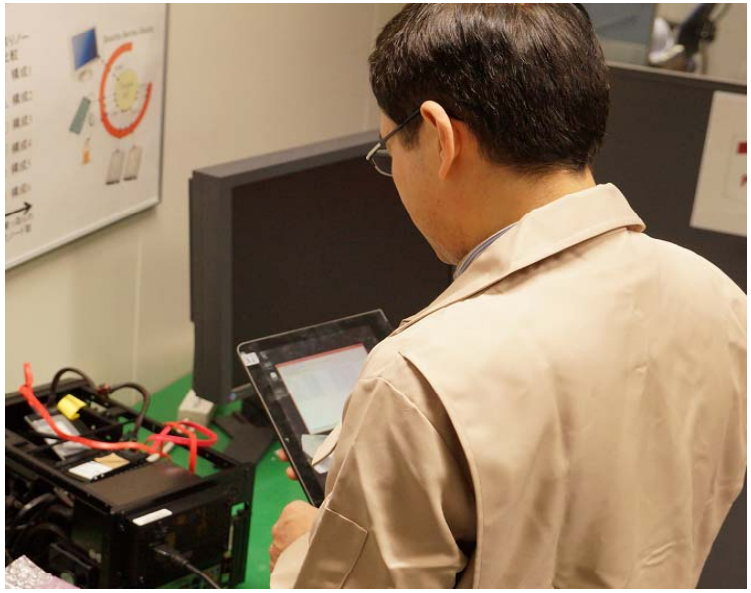
- Gyroscope is not a microphone, but it turns to be a speech logger.
- It is called Gyrophone [USENIX Security 14, BlackHat Europe 14].
  - Merit: Access to microphone requires permission, but access to gyroscope does not. It makes easy to use for cyber espionage.
  - The sampling rate of gyroscope (20-200Hz) does not fit speech (male 85 - 180 Hz, female 165 - 255 Hz), but ALIASING helps to understand speech.



Y.Michalevsky, D.Boneh, and Gabi Nakibly, “Gyrophone: Recognizing Speech from Gyroscope Signals”, https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/michalevsky

Mobile gadgets are used in a restricted area.

- Mobile gadgets are commonly used in factories, meeting rooms, hospitals, where treat important information.
- **The administrator wants to prohibit devices which are not used for work.**



Factory

Meeting

Extra Threat

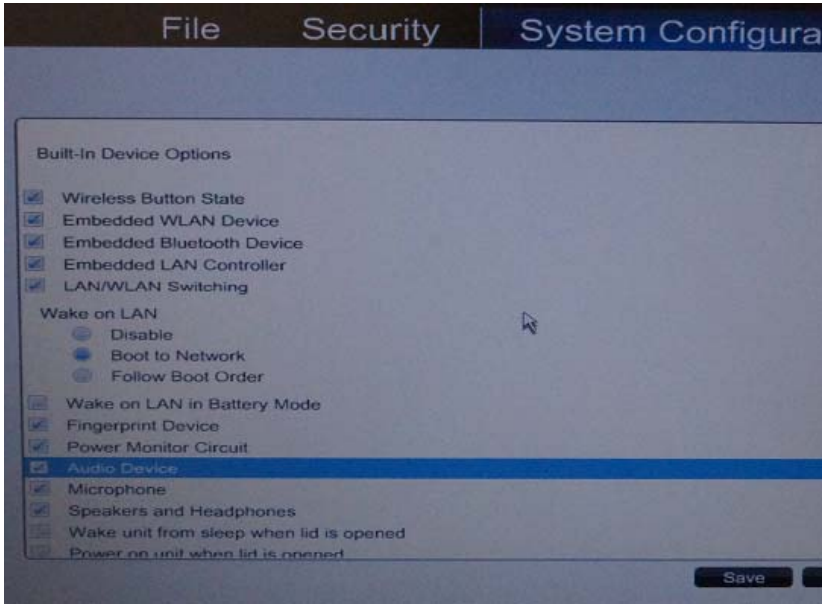
- Not only attackers but also users (workers) want to use the devices on mobile gadgets.
- **The users may circumvent countermeasures.**
- Administrators have to deal with attackers as well as workers.

Current Countermeasures

- Some BIOS/EFI can disenable devices.
  - It is useful, but all mobile gadgets do not have such function.
- Samsung KNOX disenables devices, but it runs on Samsung's Android only.
- Security goods

  Protect cap

  Security seal to hide camera



They depend on user's conscience.
Can you trust them?

My proposal

- ***"DeviceDisEnabler (DDE)":*** a lightweight hypervisor which hides devices to protect cyber espionage and tampering
- Features
  1. Lightweight and insertable to many mobile gadgets
  2. Hiding PCI devices from an OS
  3. Prevention of circumvention
     - The OS cannot boot without the DDE because a part of the disk is encrypted by the DDE.
     - The encryption key is hidden from the user.

Targets of DDE

- Mobile gadgets (Note PC, Tablet, etc.) with x86/AMD64 architecture CPU.
- DDE is developed on open source hypervisor “BitVisor”.
  - http://www.bitvisor.org/
- DDE disenables PCI devices which are not used for work.
  - Current implementation does not treat USB devices.

Laptop PC used for presentation outside of a office

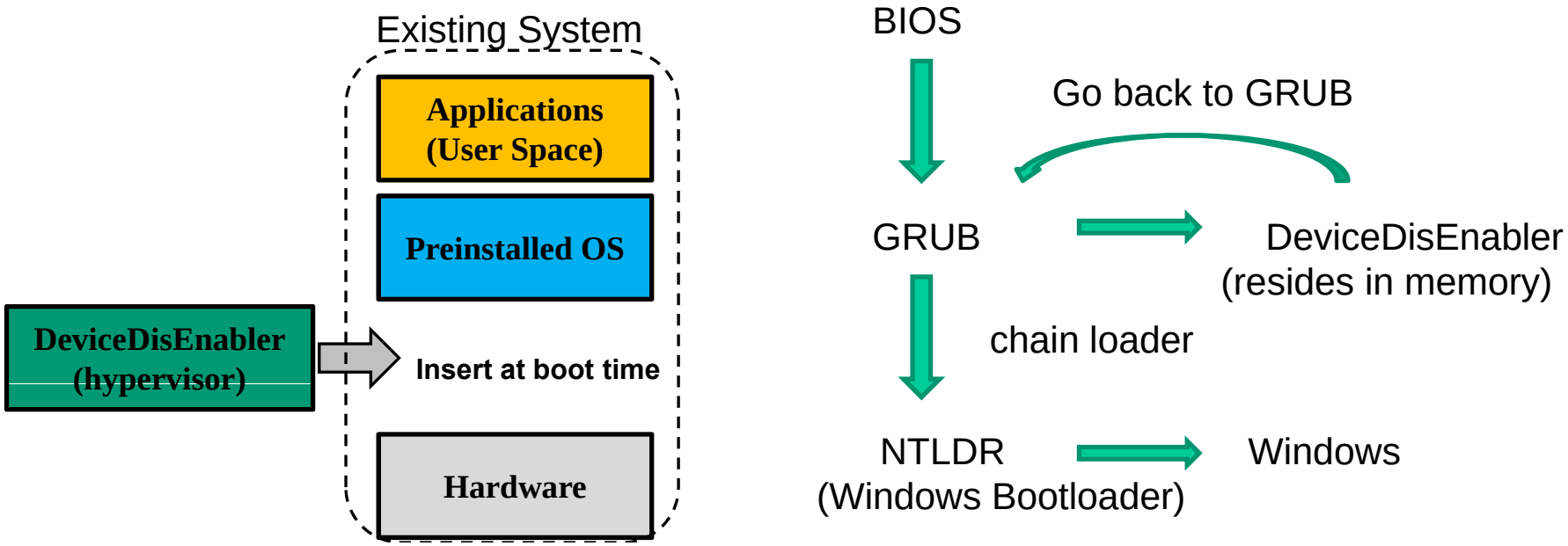
Tablet used in hospital

Division of roles between DDE and OS

- DDE manages devices.
  - The DDE is independent of the OS and hides some devices from the OS.
- OS has responsibility for the user account.
- DDE's Disk encryption is independent of the OS's encryption.
  - The DDE's Disk encryption can coexist with OS's disk encryption (e.g., Windows's BotLocker).

(1) Insertable Hypervisor on an existing OS

- Thin type-I (bare-metal) hypervisor
  - Para-passthrough architecture (BitVisor[VEE'09])
    - No Device Model. Guest OS can access devices directly.
  - Small Trusted Computing Base (TCB)
    - Type-I hypervisor has no Host OS.
- DDE is inserted using chainload function of boot-loader.



(2) Hiding PCI devices from an OS

- A mobile gadget has many devices on PCI.
- Tool: PCI-Z
  - http://www.pci-z.com/

(ThinkPad Helix)

PCI-Z 1.3 - PCI devices information utility

File Database Report Help

System information
CPU name: Intel(R) Core(TM) i5-3427U CPU @ 1.80GHz, 4 logical CPUs
Computer name: LENOVO-PC
User name: knoppix
Operating system: Microsoft Windows 8
Available memory (MB): 3914

Type	Vendor	Device	Subsystem	PCI device
SMBus	Intel Corporation	7 Series/C210 Series Chipset Family SMBus Controller		VEN_8086&DEV_1E22&SUBSYS_220717AA
Serial controller	Intel Corporation	7 Series/C210 Series Chipset Family KT Controller		VEN_8086&DEV_1E3D&SUBSYS_220717AA
PCI bridge	Intel Corporation	7 Series/C210 Series Chipset Family PCI Express Root Port 2		VEN_8086&DEV_1E12&SUBSYS_220717AA
ISA bridge	Intel Corporation	QS77 Express Chipset LPC Controller		VEN_8086&DEV_1E56&SUBSYS_220717AA
USB controller	Intel Corporation	7 Series/C210 Series Chipset Family USB Enhanced Host Controller #1		VEN_8086&DEV_1E26&SUBSYS_220717AA
Host bridge	Intel Corporation	3rd Gen Core processor DRAM Controller		VEN_8086&DEV_0154&SUBSYS_220717AA
VGA compatible con...	Intel Corporation	3rd Gen Core processor Graphics Controller		VEN_8086&DEV_0166&SUBSYS_220717AA
Audio device	Intel Corporation	7 Series/C210 Series Chipset Family High Definition Audio Controller		VEN_8086&DEV_1E20&SUBSYS_220717AA
SATA controller	Intel Corporation	7 Series Chipset Family 6-port SATA Controller [AHCI mode]		VEN_8086&DEV_1E03&SUBSYS_220717AA
PCI bridge	Intel Corporation	7 Series/C210 Series Chipset Family PCI Express Root Port 1		VEN_8086&DEV_1E10&SUBSYS_220717AA
USB controller	Intel Corporation	7 Series/C210 Series Chipset Family USB xHCI Host Controller		VEN_8086&DEV_1E31&SUBSYS_220717AA
USB controller	Intel Corporation	7 Series/C210 Series Chipset Family USB Enhanced Host Controller #2		VEN_8086&DEV_1E2D&SUBSYS_220717AA
Communication cont...	Intel Corporation	7 Series/C210 Series Chipset Family MEI Controller #1		VEN_8086&DEV_1E3A&SUBSYS_220717AA
Network controller	Intel Corporation	Centrino Advanced-N 6205 [Taylor Peak]		VEN_8086&DEV_0085&SUBSYS_C2208086

Database: The PCI ID Repository | Version: 2014.11.15 | htt

Right click on the list for options.

How an OS recognizes a device on PCI

- An OS gets the information of devices on PCI from "**PCI configuration space"**.
  - The information includes Vendor ID, Device ID, and Device Class Code, etc.
    - Vendor ID and Device Class code are defined by PCI-SIG.

PCI configuration space

- PCI configuration space is the underlying way that the Conventional PCI, PCI-X, and PCI Express perform auto configuration of the devices.

- PCI configuration space has 2 registers (I/O ports).
  1. PCI Address Register I/O port: 0x0cf8

31	30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11	10 09 08	07 06 05 04 03 02	01	00	
EN	Reserved	Bus No	Dev No	Fun No	Register Address	0	0	0x00

  2. PCI Configuration Register I/O port: 0x0cfc

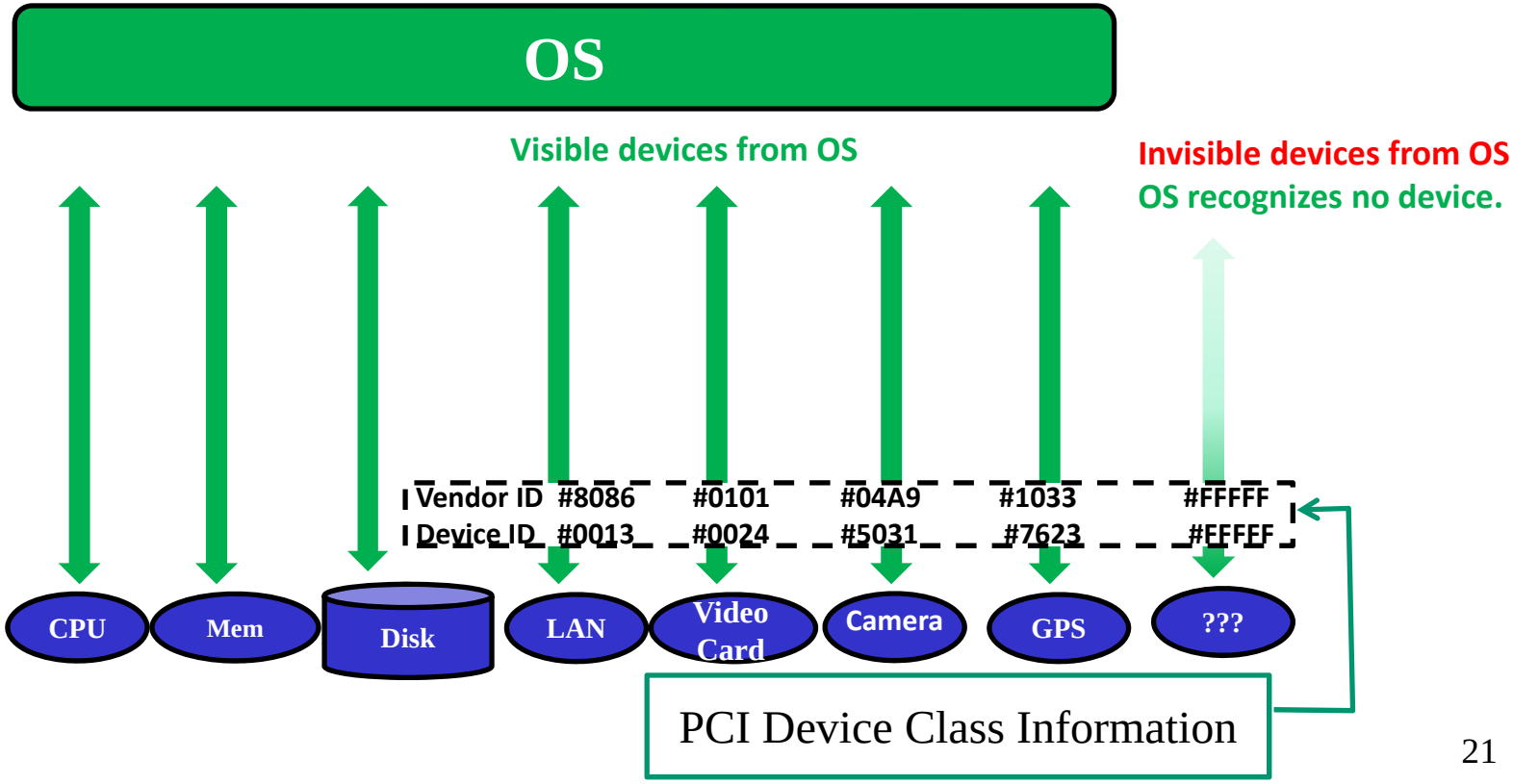
PCI Configuration Register

- I/O port: 0x0cfc

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	
Device ID	Vendor ID	0x00
Device Status	Device Control	0x04
Class Code	Revision ID	0x08
Header Type		0x0c
Base Address 0		0x10
Base Address 1		0x14
Base Address 2		0x18
Base Address 3		0x1c
Base Address 4		0x20
Base Address 5		0x24
		0x28
Subsystem ID	Subsystem Vendor ID	0x2c
		0x30
Reserved		0x34
Reserved		0x38
	Interrupt Pin / Interrupt Line	0x3c
Undefined		0x40 ~ 0xfc

Normal OS

- In order to get device information on the PCI, the OS accesses to the I/O ports (PCI configuration space).
- The OS running on Intel CPU uses I/O instructions (i.e., IN and OUT) to access the I/O ports.

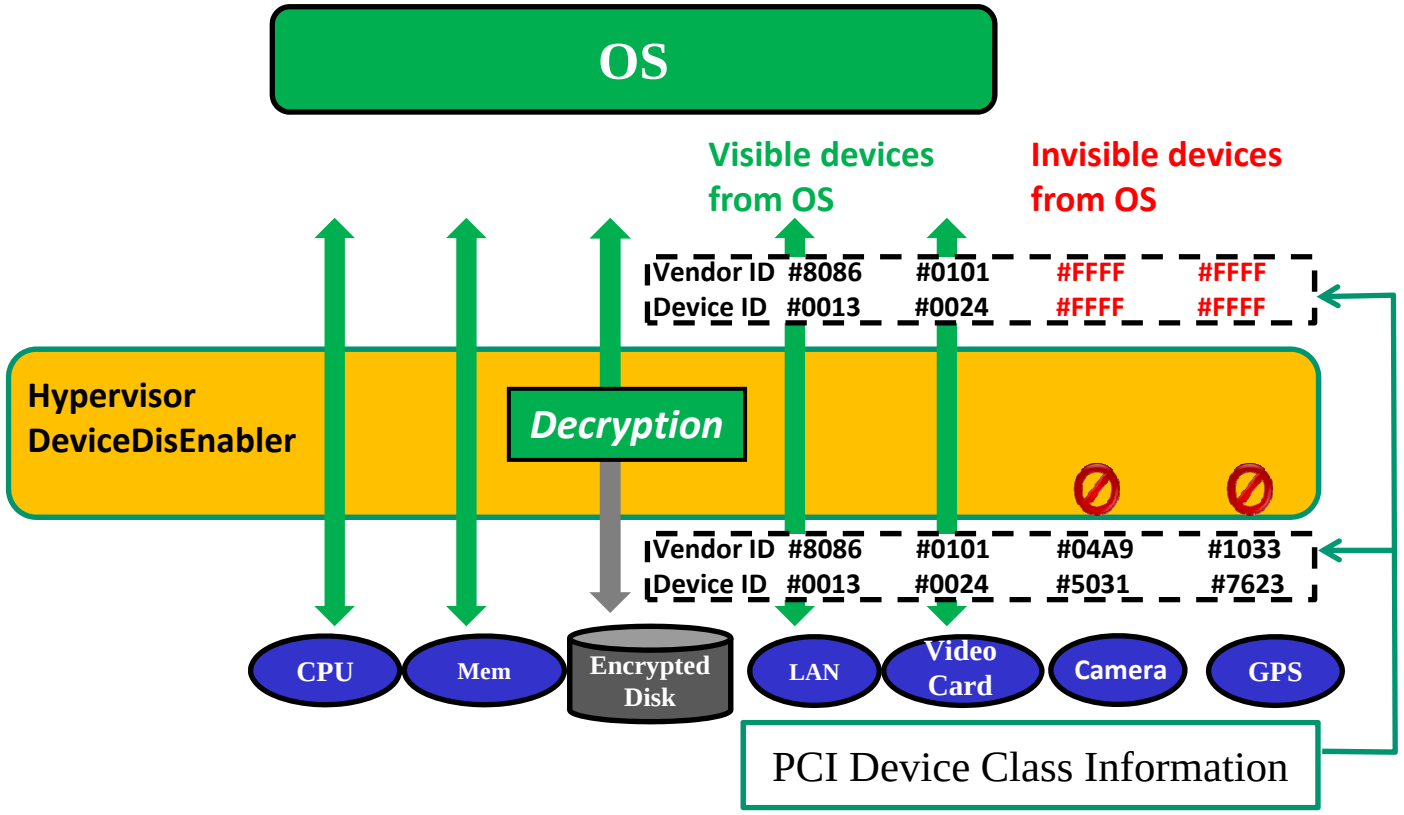


DDE hides devices

- The DDE intervenes in the I/O operations.
  - The I/O instructions (i.e., IN and OUT) issued by the OS are **trapped by Intel&AMD virtualization architecture**. Then the control is transferred to the hypervisor(DDE).
- When an I/O instruction is issued to PCI configuration space, the DDE checks the contents.

DDE hides devices

- If the DDE found the device that must be hidden, the DDE replaces the Vendor ID and Device ID with "#FFFF".
- The OS recognizes that there is no device, and the device is not used.
  - This effect is same to the hiding by BIOS.



Hidden device by DDE

- DDE has 2 types to hide the device.
  - For a certain device (Vendor ID and Device ID)
  - For a category (defined by PCI device class code)

Vendor ID	Vendor name
0x05ac	Apple, Inc.
0x04B3	IBM
0x1010	Video Logic Ltd.
0x104D	Sony Corporation
0x1061	8x8 Inc.
0x106B	Apple Inc.
0x13B5	ARM Ltd
0x12E1	Nintendo Co. Ltd.
0x13B5	ARM Ltd
0x15AD	VMware Inc.
0x15C6	Technical University Of Budapest
0x8086	Intel Corporation
0x8087	Intel
0xA304	Sony
0xF5F5	F5 Networks Inc.

Class code	Class Name
0x00	Unclassified device
0x01	Mass storage controller
0x02	Network controller
0x03	Display controller
0x04	Multimedia controller
0x05	Memory controller
0x06	Bridge
0x07	Communication controller
0x08	Generic system peripheral
0x09	Input device controller
0x0a	Docking station
0x0b	Processor
0x0c	Serial bus controller
0x0d	Wireless controller
0x0e	Intelligent controller
0x0f	Satellite communications controller
0x10	Encryption controller
0x11	Signal processing controller
0x12	Processing accelerators
0x13	Non-Essential Instrumentation
0xff	Unassigned class

(3) Prevention of circumvention

- Unfortunately, we can't rule out the possibility that users try to bypass the DDE because they want to use the devices.

- DDE’s countermeasure
  - The DDE encrypts a part of the disk and tries to make impossible to boot the OS without the DDE.

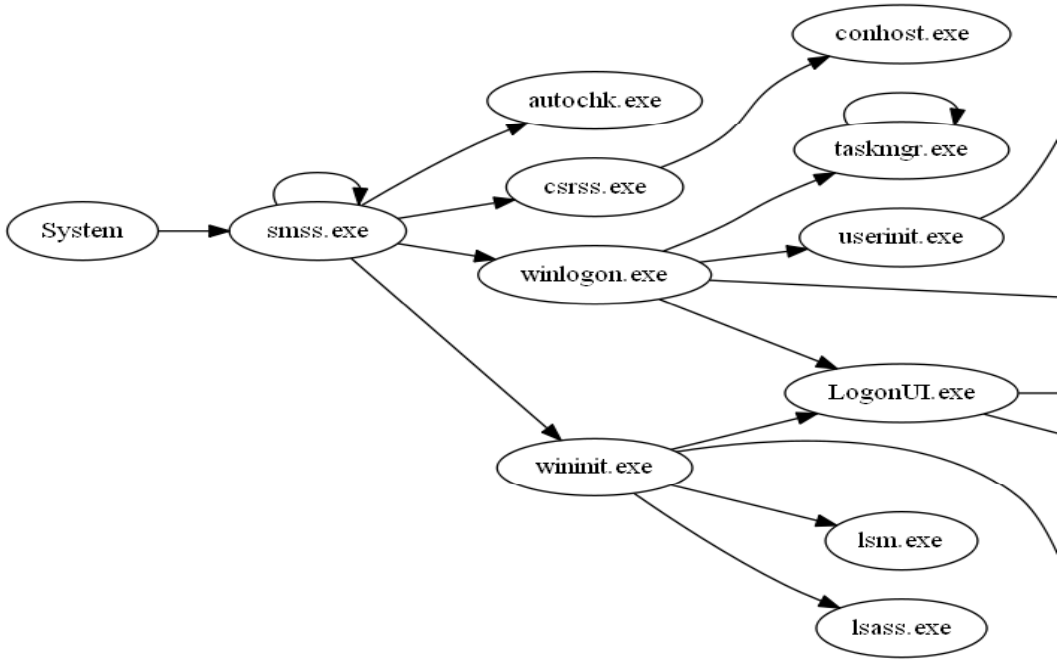
- Problem
  - However, it is not easy to stop booting OS (Windows) using simple disk-block encryption.

Difficulty to stop booting OS

- BitVisor (the base of DDE) has a function to encrypt a region (blocks) of hard-disk.
  - It is useful to protect the data when the disk is stolen.
- Unfortunately, BitVisor's encryption is not applied on a whole partition of Windows because the boot sequence can access the disk without a hypervisor.
  - Maybe, the booting of a kernel uses BIOS to access the disk. BitVisor cannot intercept the BIOS's disk access.
  - Even if the DDE decrypts the partition correctly, OS cannot boot.
- (Note) If I can use Linux, I can separate the disk image into 2 partitions: miniroot and rootFS. The miniroot is used for booting Linux kernel and rootFS is used for mounting root file system. The DDE encrypts the partition of rootFS and stops the booting of the Linux properly.

Stop Windows booting

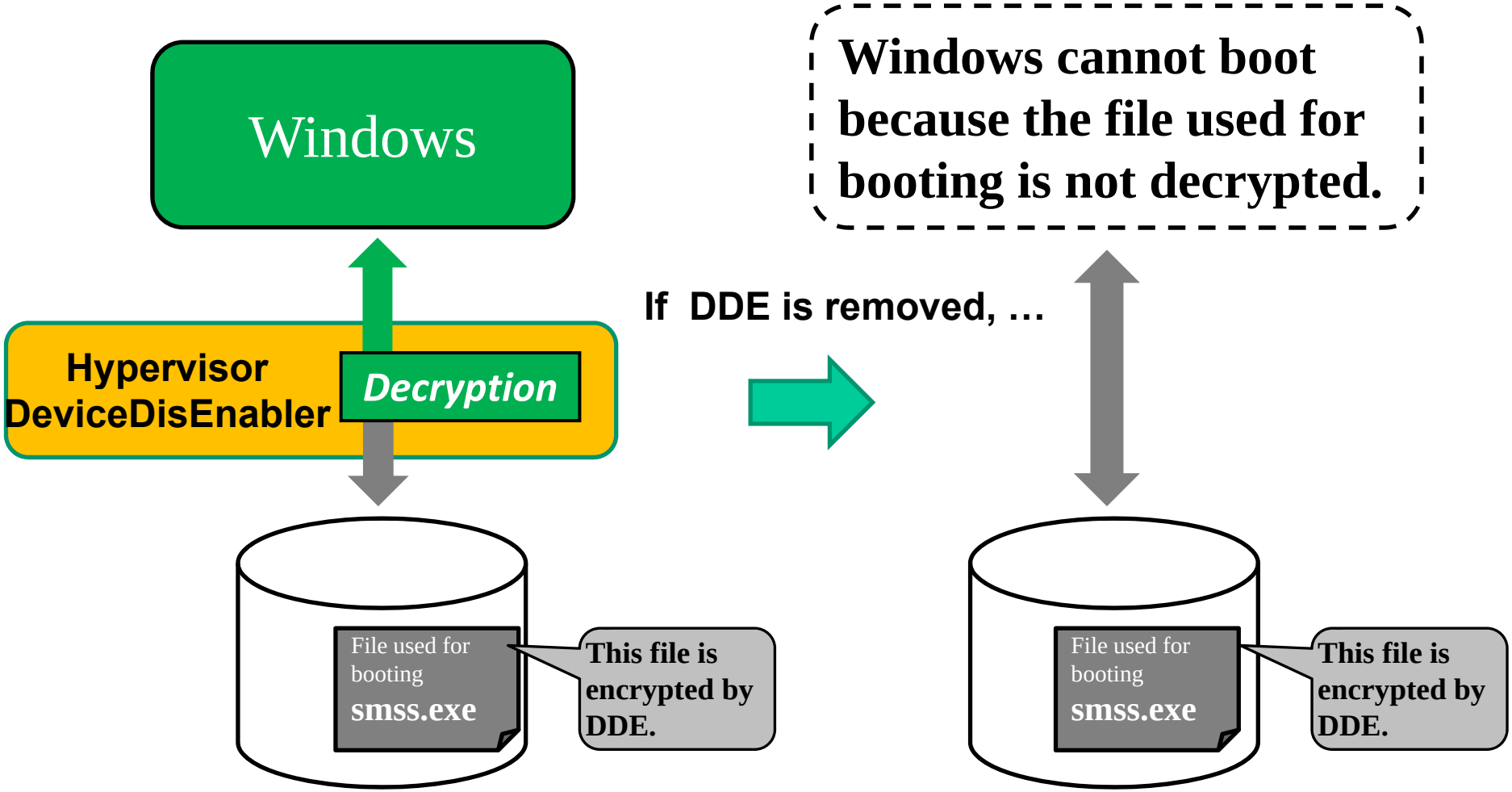
- I give up stopping kernel booting. I tried to stop the boot sequence on user space.
- I analyzed the boot sequence on user space of Windows, and try to encrypt a file which is needed to boot Windows.
- I chose “smss.exe” file to be encrypted by the DDE.
  - If the file is not decrypted by the DDE, Windows don’t not boot properly.



Finding blocks of a file

- (Problem) It is not easy to find disk-blocks for a file on NTFS.
  - I used a tool offered by Mark Roddy.
    - getFileExtents.exe
    - http://www.wd-3.com/archive/luserland.htm
  - The getFileExtents worked well on Windows7. However, Windows8 has a harder security mechanism and the getFileExtents does not work well.
    - Handler “initFileTranslation” is not available on Windows 8.
  - For windows8, I make a disk copy with “dd” command and mount the disk image on Window7. It makes possible to find disk blocks for a file using getFileExtents.

Stop Window boot by DDE



- I could make tamper resistance for DDE, but …

Struggle with Recovery Mechanism

- Current OS has ***automatic recovery mechanism***.
  - Automatic recovery mechanism can fix a broken file.
    - e.g., Windows RE (Recovery Environment)
- On current implementation of DDE, administrator must halt the recovery mechanism on Windows 8.

- This problem does not solve yet. However, the situation is same to re-install attack.
  - When a user tries to re-install the OS on the target machine, most countermeasure mechanisms cannot prevent it.

Hiding an encryption key

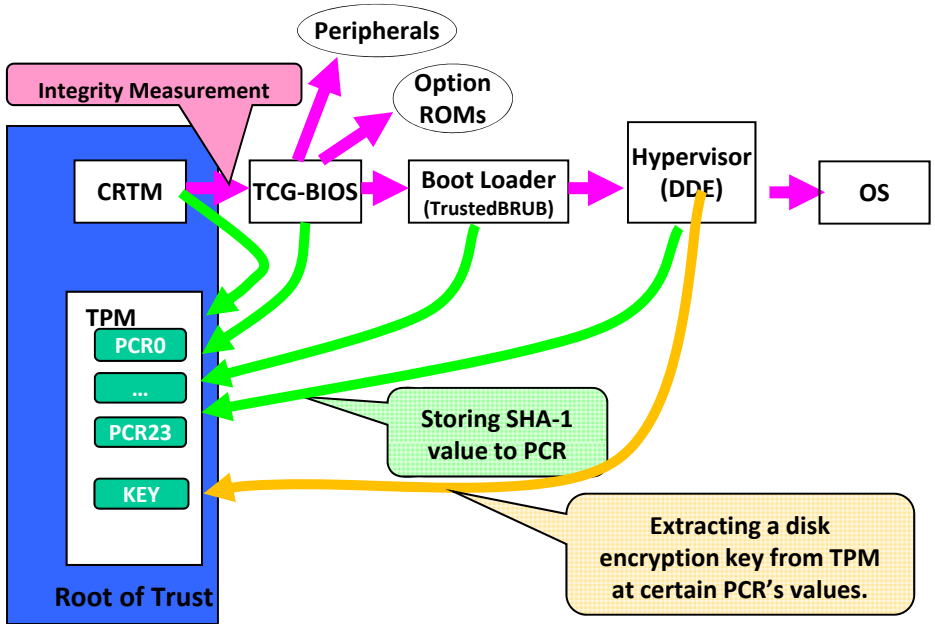
- The encryption key of DDE must be unknown to the user.
- Original BitVisor only includes the key in the binary.
  - Attacker can get the key by comparing the binaries of DDE.

- DDE's solution
  - The encryption key is hidden in a secure chip TPM (Trusted Platform Module).

Hiding encryption key in a TPM (1/3)

- TPM offers a mechanism of ***Trusted Boot***. Trusted Boot measures boot sequence and keeps the log. It makes possible to certify the integrity of the boot sequence (i.e., ***Chain of Trust***).
  - The SHA-1 of each sequence (e.g., BIOS, peripherals, bootloader, etc.) is stored to a ***PCR (Platform Configuration Register***) in a TPM with "extend" operation.
    - *PCR=SHA-1(PCR + SHA-1(Component))*
  - It means that PCR shows the stage of the boot sequence.

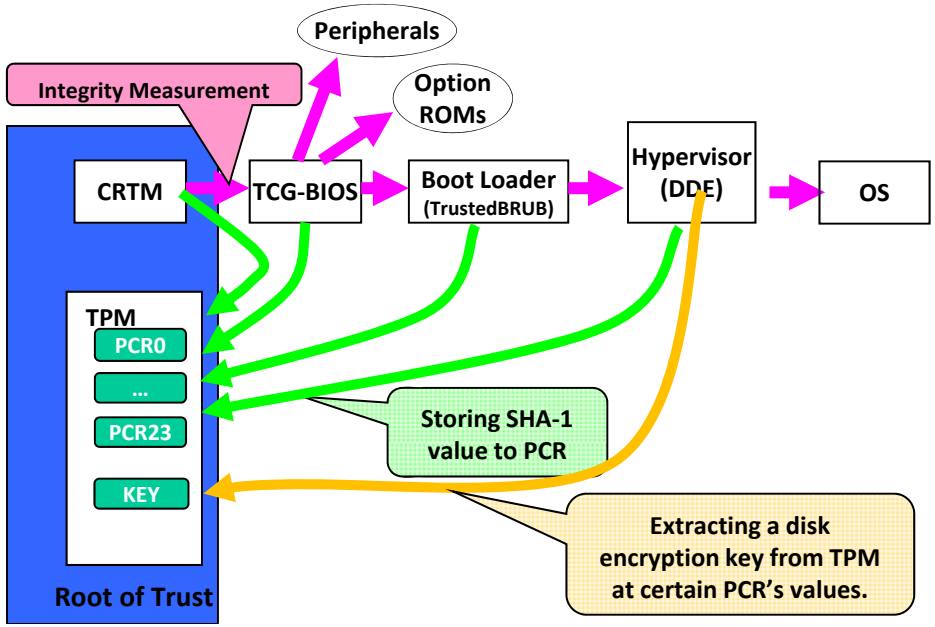
Each PCR represents category of action.



Hiding encryption key in a TPM (2/3)

- In order to keep ***"Chain of Trust"***, each component must have a function to measure next component.
  - The mobile gadget must have TCG-BIOS as well as TPM.
  - The boot loader must support measurement function.
    - Trusted GRUB http://sourceforge.net/projects/trustedgrub

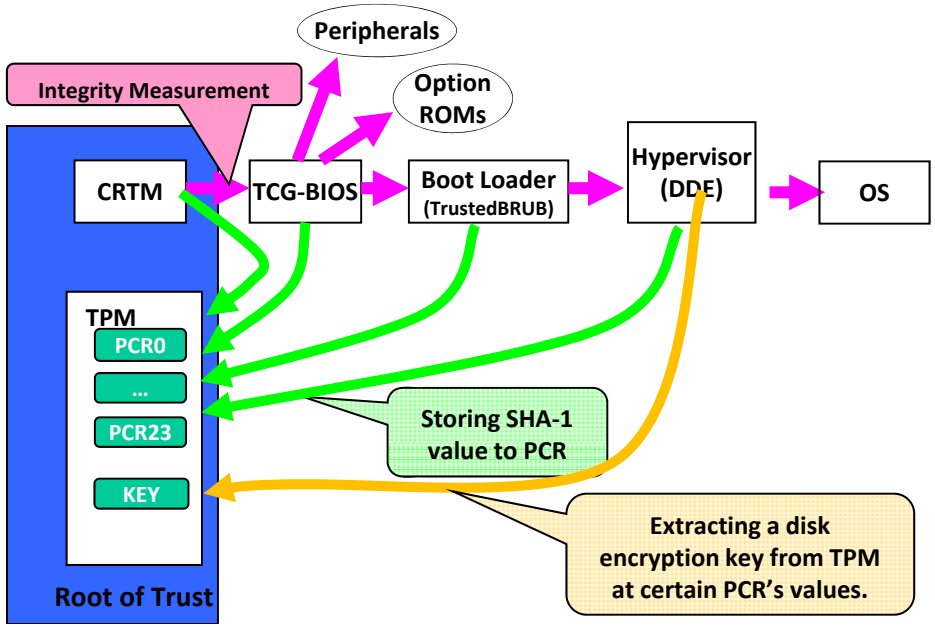
Each PCR represents category of action.



Hiding encryption key in a TPM (3/3)

- The encryption key is stored to a TPM. It can be set to extract at certain PCR values.
  - If the binary of DDE is customized, PCR values are changed, the key is not extracted.
- It means that **the users MUST use the valid DDE**.

Each PCR represents category of action.



Chain of Trust

- Boot sequence on ThinkPad Helix
  - Each Sequence is measured on a TPM.
    - *PCR=SHA-1(PCR + SHA-1(Component))*

Each PCR represents category of action.

```
PCR    SHA1                                     Event
↓      ↓                                        ↓
 0 4b81c044c1472a34c73da87d7ad3a64ba62e9047 08 [S-CRTM Version]
 6 fcad787f7771637d659638d92b5eee9385b3d7b9 05 [Wake Event 6]
 0 8841e9e7d8eb4c753d2ef7dc9f89a07c756cb30b 07 [S-CRTM Contents]
 0 3d9766e45814d6374d9a85aa519071dc82574017 01 [POST CODE]
 1 b83f6c64a1727add477a94874f3f11f29d531c47 09 [CPU Microcode]
 4 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 2 199804c152f10535cd88f8f5d607ae55e9e2f3ef 06 [Option ROM]
 5 cd0fdb4531a6ec41be2753ba042637d6e5f7f256 80000007 []
 0 afbf30b554a35d0ba6a469934d35cf9f58eec6af 80000009 []
 1 8de522ea7b732f0bf261ed931245c5c7e75fedbb 80000009 []
 0 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 1 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 2 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 3 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 5 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 6 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 7 9069ca78e7450a285173431b3e52c5c25299e473 04 []
 1 1f3c97f0b6d45a46ec1aa91e5868322dea94d76c 80000002 []
 4 c1e25c3f6b0dc78d57296aa2870ca6f782ccf80f 05 [Calling INT 19h]
 4 d564bb707b030e193fdd3ddae8818703225c49c3 05 [Booting BCV Hard
Disk]
 4 f2e7a20ef1397308f937841b55040905ff7cabca 0d [IPL]
 5 c358aaa78d400ad539f90d542e5519aa4e403714 0e [IPL Partition Data]
 4 e479a239ff8d17b2391782a86e19ca873ec6536c 0d [IPL]
```

TPM non-volatile storage

- TPM has storage system named ***"TPM non-volatile storage",*** which allows access when PCRs has certain values.
- The disk encryption key of DDE is stored on the storage, which prevents the circumvention of DDE.
  - Because the PCR values are changed when the binary of DDE is customized.

- Reference
  - TPM Main Part 3 Commands, Specification Version 1.2, Level 2 Revision 116, 1 March 2011

http://www.trustedcomputinggroup.org/files/static_page_files/72C33D71-1A4B-B294-D02C7DF86630BE7C/TPM_Main-Part_3_Commands_v1.2_rev116_01032011.pdf

Register encryption key to TPM non-volatile storage

- The “TPM non-volatile storage” is accessed by the API offered by TCG-BIOS.

API of TCG BIOS	Description
TPM_NV_DefineSpace	•API to reserve a region of TPM non-volatile storage. •The region has “index” number to access. •The access can be limited by certain vales of PCRs.
TPM_NV_WriteValue	•API to write data to the TPM non-volatile storage. •The region is accessed when PCRs are same to registered values.
TPM_NV_ReadValue	•API to read data from the TPM non-volatile storage. •The region is accessed when PCRs are same to registered values.

TPM non-volatile storage for DDE

- A region of TPM non-volatile storage has an index to access.
- The region can be read/written when the hash of PCR[0-7,12-14] is the registered hash value.

On ThinkPad Helix

```
# tpm_nvinfo
      NVRAM index   : 0x00010016 (65558)
      PCR read  selection:
      PCRs   : 0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14
      Localities   : 0x7
      Hash   : bcea2524269cafd359d69caa850e209481feeec4
      PCR write selection:
      PCRs   : 0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14
      Localities   : 0x7
      Hash   : bcea2524269cafd359d69caa850e209481feeec4
      Permissions   : 0x00000000 ()
      bReadSTClear  : FALSE
      bWriteSTClear : FALSE
      bWriteDefine  : FALSE
     Size          : 32 (0x20)
```

PCRs to verify

Hash of values of PCRs

PCRs to verify

Hash of values of PCRs

PCRs on TPM

On ThinkPad Helix

Trusted GRUB uses PCR[12-14]

PCR[0-7, 12-14] are used to get the encryption key from the TPM non-volatile storage.

PCR[0-7] are used to certify the true boot sequence before Trusted GRUB.

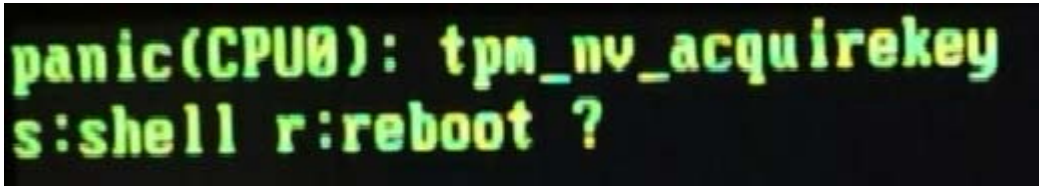
PCR[12-14] are changed when the DDE is customized.

Original DDE

```
PCR-00: 27 CD 64 2F DA 95 EA 09 3B 8C AE BC 68 9F FA C7
2A 59 76 01
PCR-01: E2 60 C4 57 A9 DC 8B C1 3C 5D E8 23 9F 2B 6B 71
86 19 72 19
PCR-02: F2 E5 65 2A DC 7F 57 8A F0 89 9D F1 0F 6B AE A1
13 08 19 E2
PCR-03: B2 A8 3B 0E BF 2F 83 74 29 9A 5B 2B DF C3 1E A9
55 AD 72 36
PCR-04: AA C6 8F 43 8F 5C 23 4E BD 70 F7 46 7D 51 18 4E
BD A3 CA 55
PCR-05: 01 C2 F5 26 13 11 B9 6F 4B BF A4 39 14 AC CA 6B
CD A2 65 41
PCR-06: EE 1B 0F 99 7D 75 17 B2 86 BC 9D 73 A4 CF 74 2C
65 A7 69 BE
PCR-07: B2 A8 3B 0E BF 2F 83 74 29 9A 5B 2B DF C3 1E A9
55 AD 72 36
PCR-08: 93 41 C4 1A 6D EA 42 08 65 16 B8 4B AF AF 48 3C
CD 96 36 91
PCR-09: 1B 60 78 EA 42 8E FA 3A 2A D2 A9 7E 22 04 90 7C
1A E6 33 A9
PCR-10: 3D C7 DF C4 CB B0 EC D3 9F B2 75 14 4B 41 E0 42
52 AF C1 17
PCR-11: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
PCR-12: 98 CB C3 5A 43 22 54 CB CB DD E6 04 30 B1 89 D9
54 E4 E7 F8
PCR-13: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
PCR-14: FB 17 F0 8C C8 F0 1F D6 8B 96 63 14 63 54 70 A4
```

Failing Booting

- If the DDE is customized, it fails to get the encryption key from TPM non-volatile storage.



Current Implementation

- Current DDE is applied on laptop PC and tablet which satisfy the following requirements.
  - x86/AMD64 architecture CPU
  - TPM 1.2
  - TCG BIOS (Current DDE does not support EFI.)
  - Only PCI devices are controlled.
  - OS independent (I have tried Windows 7,8, and Linux)

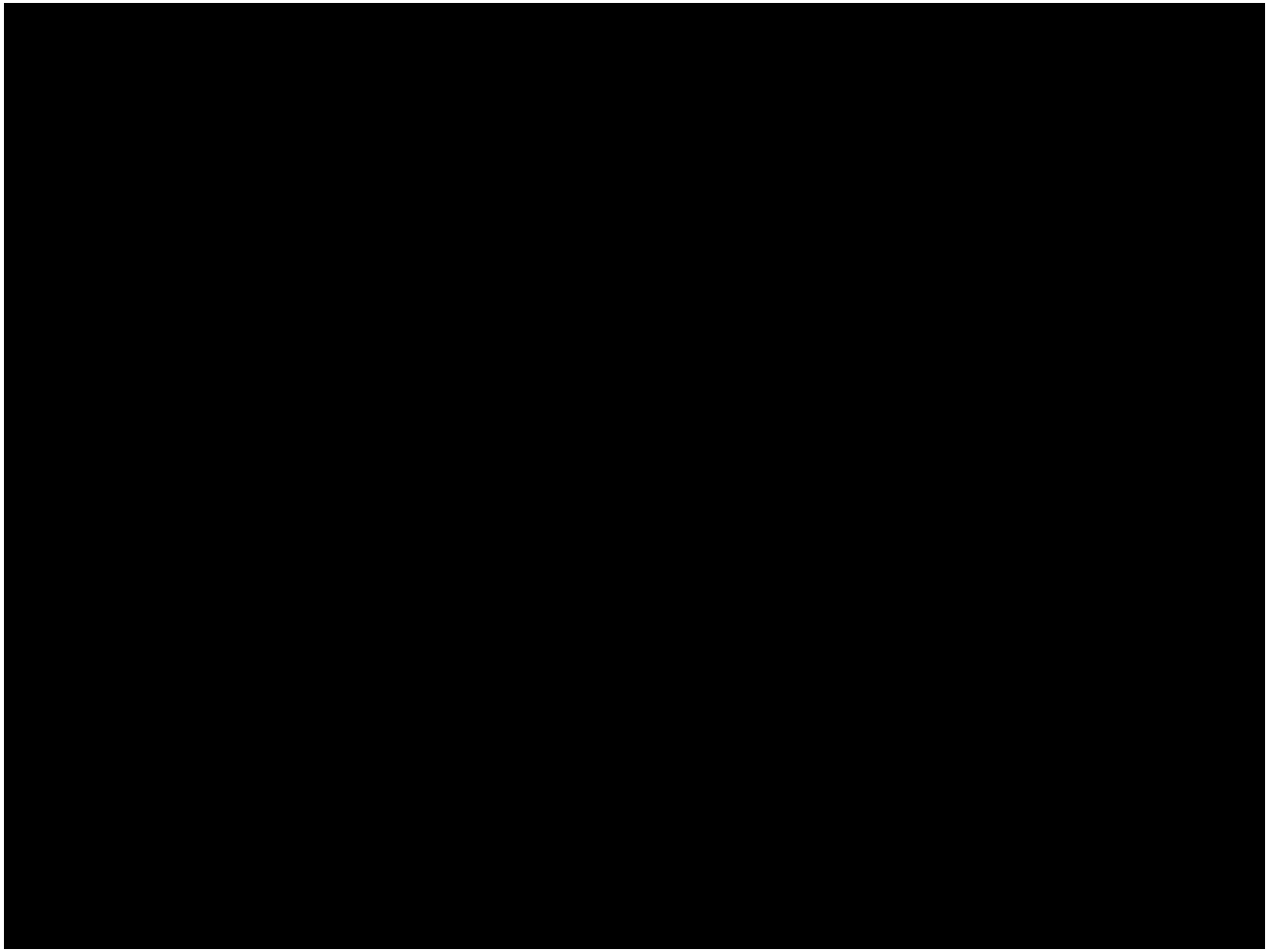
Demo Video

- Three kinds of booting
  - Standalone boot of Windows8
    - smss.exe is encrypted by the DDE and it fails to boot.
  - Customized DDE
    - It cannot get the encryption key and fails to boot.
  - DDE and Windows8
    - It works well.

Just Fun!

Trusted GRUB has 3 boot options.

- Windows 8
- DDE
- noDDE (Customized DDE)



Conclusion

- High-resolution devices on mobile gadgets may be used for cyber espionage.
  - Administrators want to disenable unnecessary devices on their working place.
- I proposed thin hypervisor “DeviceDisEnabler” which hides devices from an OS.
- DeviceDisEnabler has a tamper resistance mechanism to prevent the circumvention caused by users.

- As future work
  - I will develop DeviceDisEnabler for EFI boot and USB device hiding.

Special Thanks

- Toshiki Yagi, AIST
- Michitaka Yoshimoto, AIST
- Kazukuni Kobara, AIST

- Developers for BitVisor
  - http://www.bitvisor.org/