



TrustICE: Hardware-assisted Isolated Computing Environments on Mobile Devices

Presented by Zhenyu Ning



Contents

1. Introduction
2. Motivation
3. Implementation
4. Evaluation
5. Conclusion



Contents

1. Introduction

2. Motivation

3. Implementation

4. Evaluation

5. Conclusion



ICE

- Isolated Computing Environments.
- To protect critical codes or perform some analysis.
- Virtualization, emulation or hardware-assisted isolation.

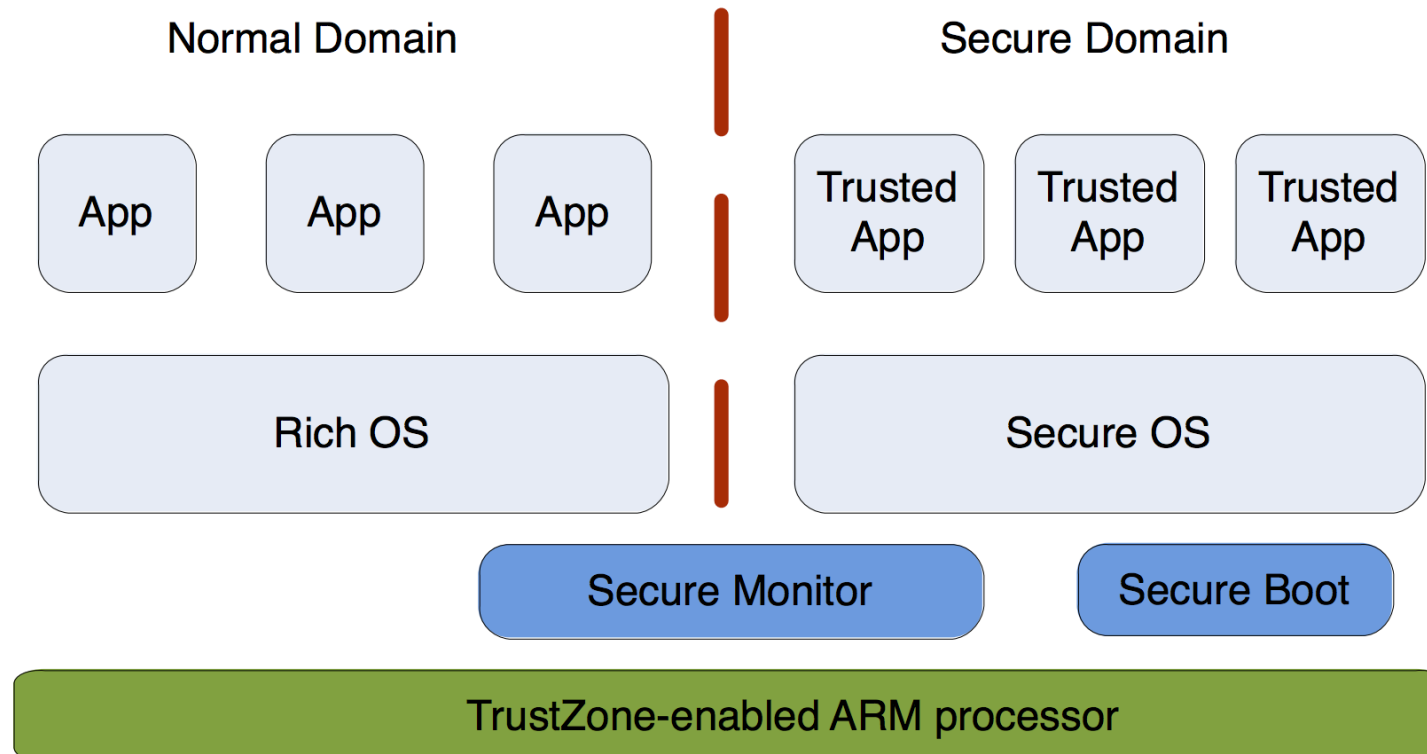


TrustZone

- Hardware security extension in ARM processors.
- Available in most nowadays Android mobile devices.
- Provide CPU state isolation, memory isolation and I/O device isolation.



TrustZone





CPU state isolation

- Normal state and secure state identified by NS bit in SCR.
- Traditional CPU modes in each state.
- A monitor mode as a gatekeeper managing state switching.
- SMC instruction to enter monitor mode.



Memory isolation

- Different memory translation map in the two states.
- TZASC partition the memory into secure region and non-secure region.
- Watermark regions in i.MX53 QSB.
 - Two Watermark regions.
 - Continuous memory region not exceed 256MB for each Watermark region.



I/O device isolation

- **Hardware interrupt isolation**
 - TrustZone Interrupt Controller(TZIC)
 - IRQ and FIQ
- **DMA isolation**
 - Direct Memory Access Controller(DMAC)



Contents

1. Introduction

2. Motivation

3. Implementation

4. Evaluation

5. Conclusion



Motivation

- **Software-based hypervisor and emulator**
 - Easy to compromise
- **Hardware-based hypervisor**
 - Large Trust Computing Base(TCB)
- **Trusted application based on TrustZone**
 - Increasing TCB
 - Tough OEMs



Contents

1. Introduction

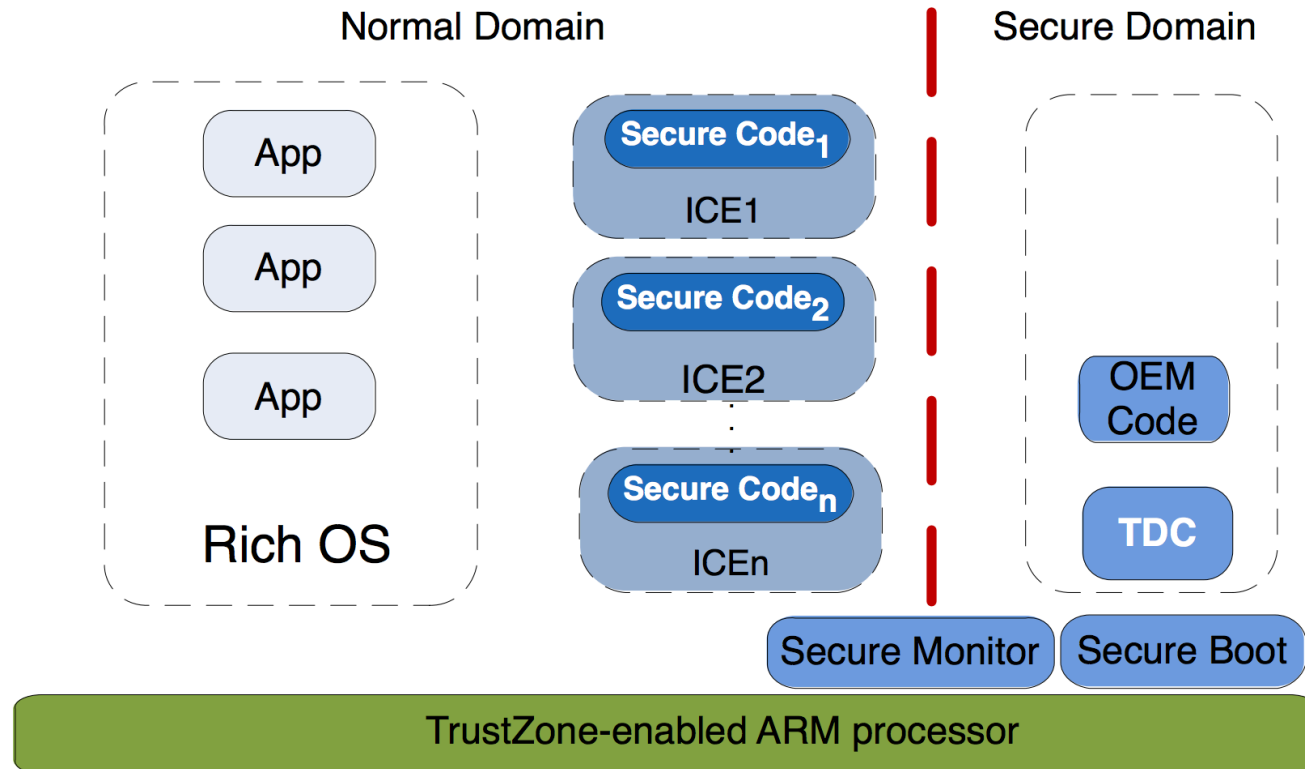
2. Motivation

3. Implementation

4. Evaluation

5. Conclusion

Architecture





Architecture

- TDC codes, ICE codes and secure codes.
- Dynamically load secure code to ICE.
- Secure switching between Rich OS and ICEs.
- Isolation between Rich OS and ICEs.

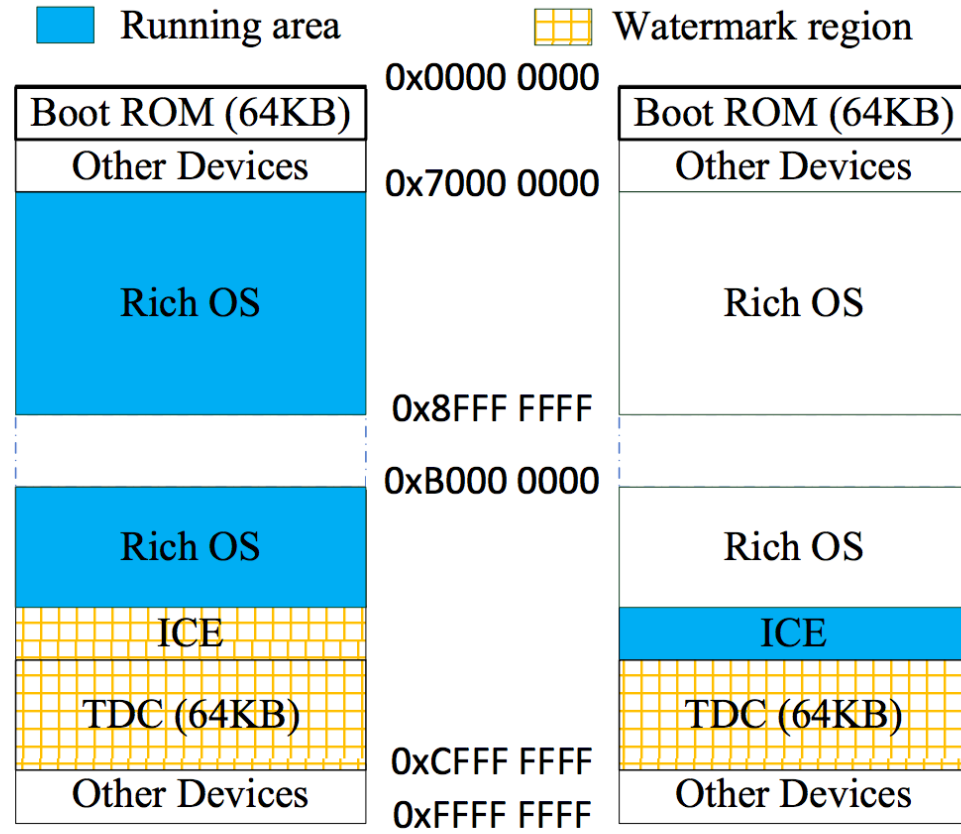
Implementation



How to protect ICE image?



Dynamic Watermark region

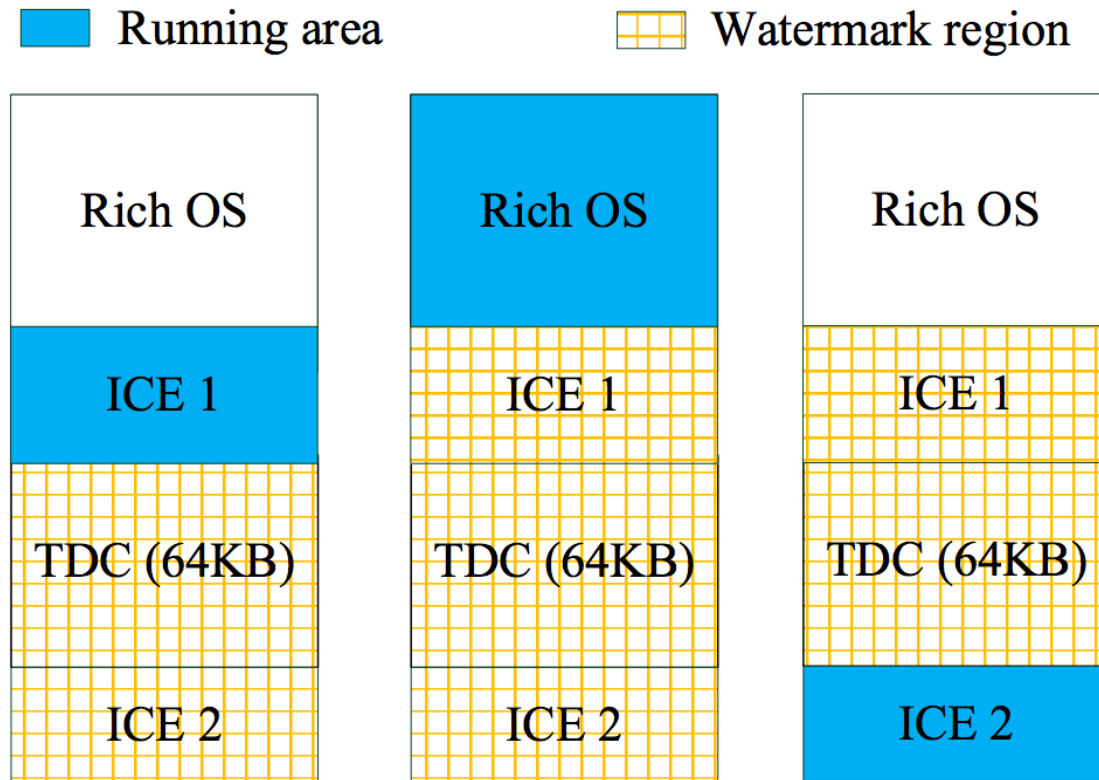


(a) Rich OS is running

(b) ICE is running




Dynamic Watermark region




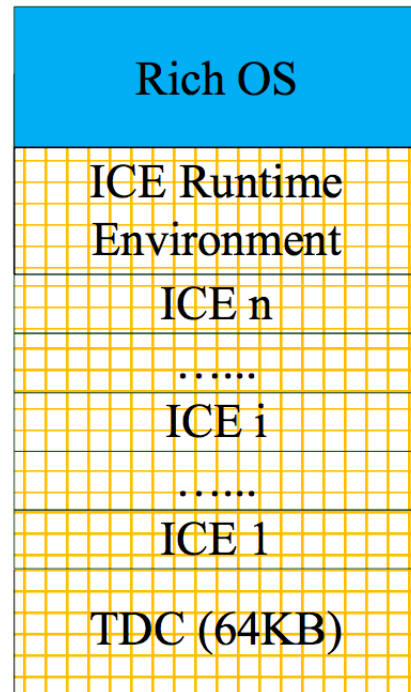
(a) ICE 1 is running (b) Rich OS is running (c) ICE 2 is running



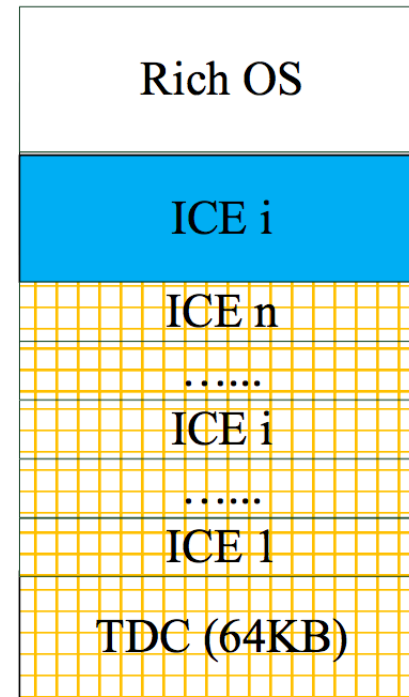
Dynamic Watermark region

 Running area

 Watermark region



(a) Rich OS is running



(b) ICE i is running

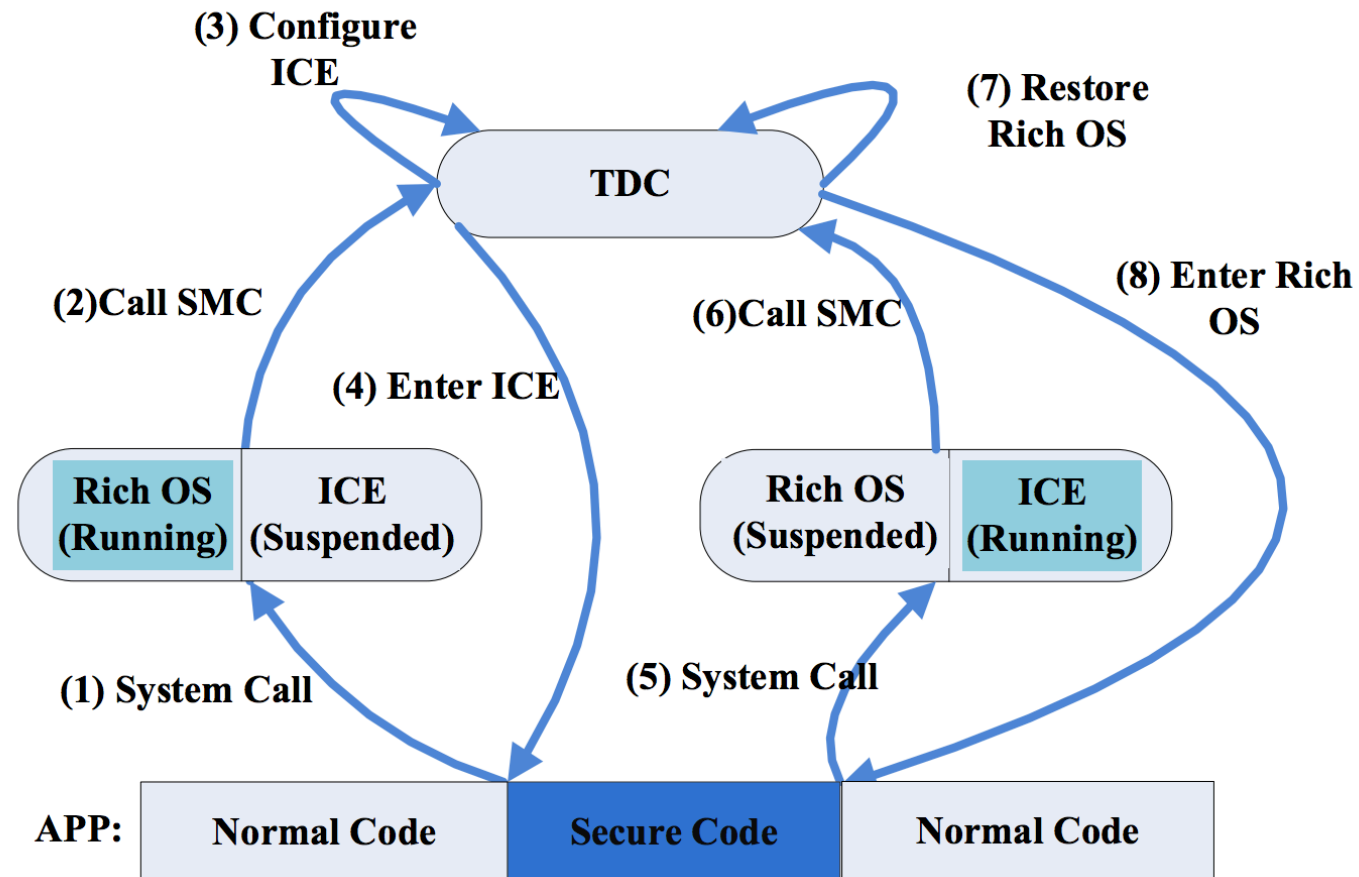
Implementation



How to protect ICE when running?



System State Switching





Implementation

- ICE code is running in non-secure Supervisor mode and secure code runs in non-secure user mode.
- ICE code provides secure system calls.
- Both the head and the tail of secure code should be SMC system call.
- Secure code can not rely on Rich OS.



Secure Isolation

- **CPU isolation**
 - Save all CPU state information before enter ICE.
 - Clean up foot print and recover the CPU state information before enter Rich OS.
- **Memory isolation**
 - Dynamically change Watermark region.
- **I/O device isolation**
 - Enable a minimal set of required interrupts and disable all the other interrupts.



Trusted Path

- Verify secure bootloader image using RSA public key stored in eFuse.
- Secure bootloader is responsible for ensuring the secure load of the ICs.
- Use some signal that only be controlled by TDC to indicate a successful switching.



Contents

1. Introduction
2. Motivation
3. Implementation
- 4. Evaluation**
5. Conclusion



Switching time

Operation	Encryption ICE (<i>us</i>)	Interface ICE (<i>us</i>)
exiting the Rich OS	5.84	5.93
verifying secure code	9.76	9.75
verifying the ICE	475.85	10559.37
configuring the ICE	35.05	34.89
entering ICE	1.27	1.27
total	527.77	10611.21

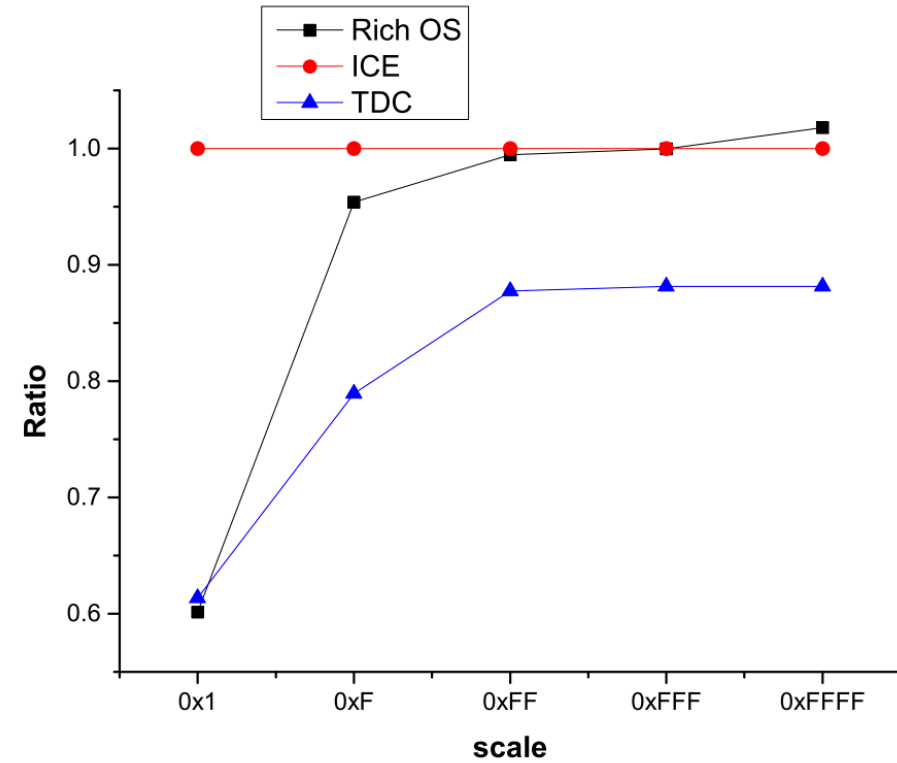
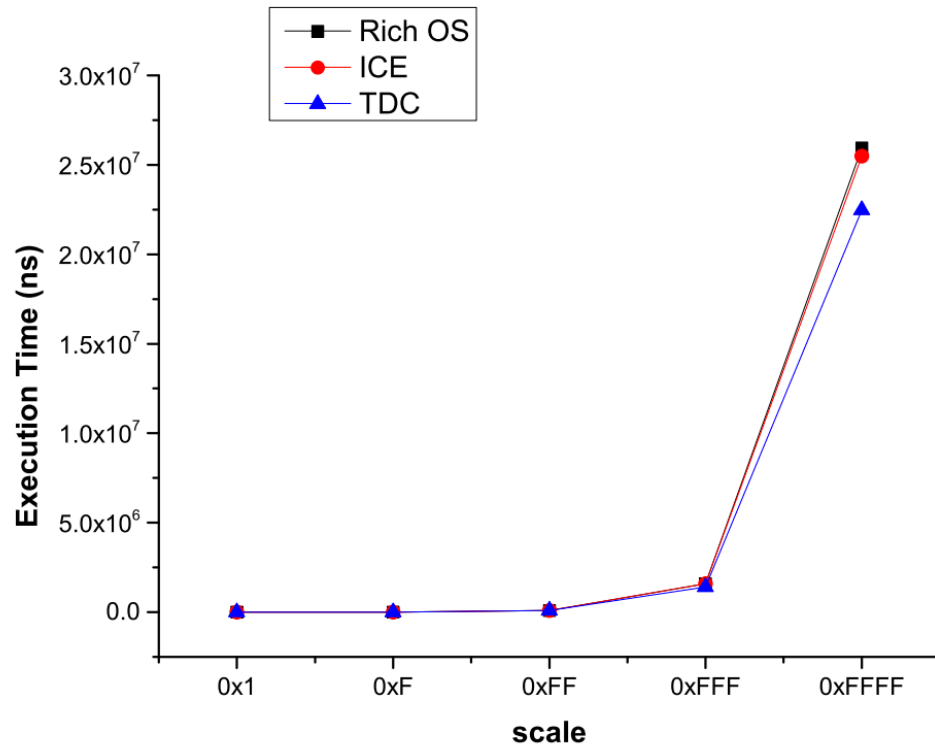


Switching time

Operation	Encryption ICE (<i>us</i>)	Interface ICE (<i>us</i>)
exiting ICE	0.49	0.48
restoring the Rich OS	19.26	19.41
entering the Rich OS	763.72	763.07
total	783.47	782.96

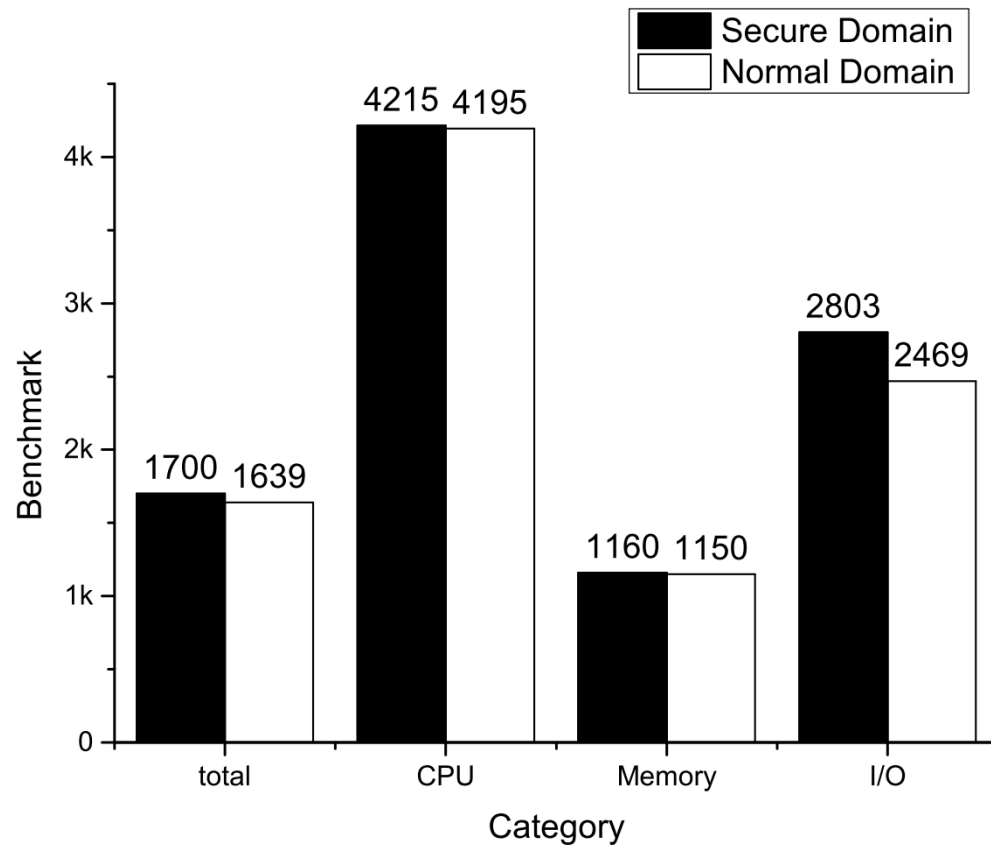


Execution time





Other evaluation



System	Power (W)
The Rich OS	2.49
TDC	2.47
ICE	2.47



More than two ICEs

- Additional time to copy the ICE into ICE runtime environment.
 - 2.85ms for the encryption ICE and 68.44ms for the interface ICE.
- Maybe hardware platform can provide a flexible Watermark solution.



Contents

1. Introduction
2. Motivation
3. Implementation
4. Evaluation
5. Conclusion



Conclusion

- TrustICE: Hardware-assisted Isolated Computing Environments on Mobile Devices.
 - Security
 - Flexibility
- Small TCB and low overhead.
 - TDC and ICE are relative small.
 - Low performance overhead while amount of ICE is below 2.



Reference

- H. Sun, K. Sun, Y. Wang, J. Jing, and H. Wang, “TrustICE: Hardware-assisted Isolated Computing Environments on Mobile Devices,” in Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN’15), June 22-25, 2015.



Thank you!