# On the Cost-Effectiveness of TrustZone Defense on ARM Platform

NAIWEI LIU, WANYU ZANG, MENG YU, RAVI SANDHU

UTSA ICS LAB; ROOSEVELT UNIVERSITY

# Contents

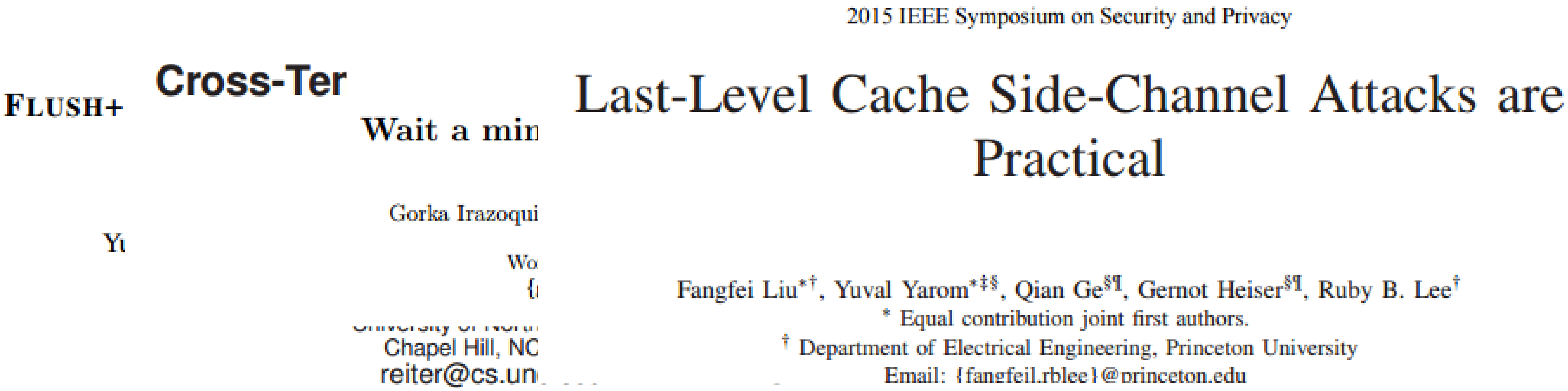
- Abstract and Introduction
- Related Work
- Cache-Based Security Threats and Attack
- Design and Implementations
- Experimental Results
- Evaluation
- Future Work
- Conclusion

# Abstract and Introduction

- Abstract
  - In Recent years, many research efforts had been made on secure and safe environment on ARM platform.
  - ARM structure and chips based on ARM had been taking up a lot of number of products in the market.
  - Security problems and potential risks had been discussed.
  - Cache and similar design brings in ‘trouble’ for security purposes.
  - Uniqueness on ARM-based products made things even tougher to solve.
  - What will we do?
    - Design defense framework for ARM
    - Evaluate by experiments
    - Optimization

# Abstract and Introduction

- Introduction
  - Last-Level Cache (LLC) is always the target of side-channel attack. On x86 structure, it is always L3 cache that is attacked.
  - Last-level cache side-channels are effective enough to extract user's private information.
  - Side-channel: collecting information like performance counters, timing, power consumption, etc. And process the information to derive information about the victim.
  - Most frequently used: access time based side-channels.



FLUSH+

Cross-Ter

Wait a min

Gorka Irazoqui

Yı

Wo

{

Chapel Hill, NC

reiter@cs.un

2015 IEEE Symposium on Security and Privacy

Last-Level Cache Side-Channel Attacks are Practical

Fangfei Liu*†, Yuval Yarom*‡§, Qian Ge§¶, Gernot Heiser§¶, Ruby B. Lee†

* Equal contribution joint first authors.

† Department of Electrical Engineering, Princeton University

Email: {fangfeil,rblee}@princeton.edu

# Abstract and Introduction

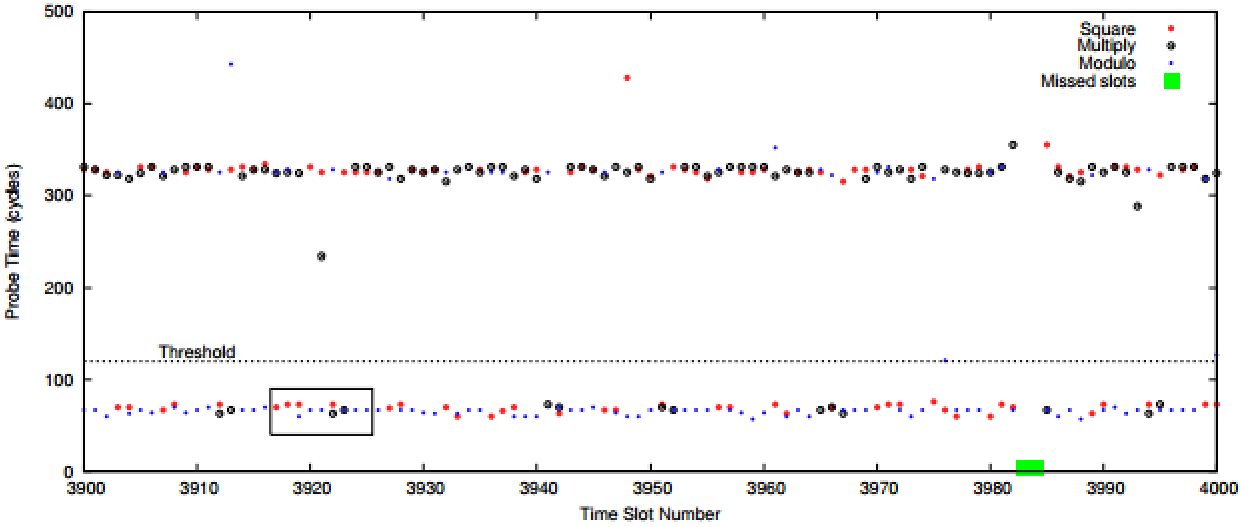
- Introduction (Continued)
  - Side-channel attack based via LLC can be dangerous, even without compromising OS.
  - Both on single OS machine and Virtual Machines (VMs) can be attacked.
  - Typical type: FLUSH+RELOAD
    - LLC is shared.
    - FLUSH+RELOAD can be practical using unprivileged instructions.
    - AES key of OpenSSL is recovered by this attack in lab test.
  - Threats to the Internet of Things (IoT) and devices
  - Modern TrustZone Design on ARM platform

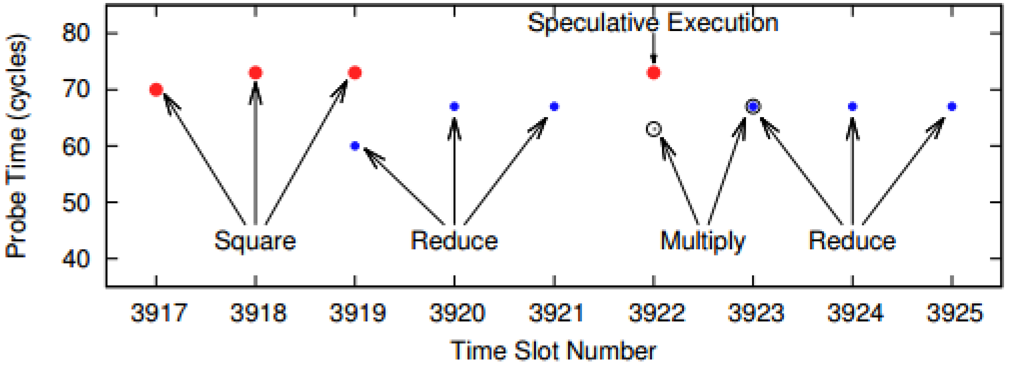
# Abstract and Introduction

- Introduction (Continued)
  - Contributions
    - Research on side-channel and covert-channel attack: bandwidth and effect.
    - Investigation on Flush operations on ARM platform and overhead.
    - Study of TrustZone technology and previous security design based on TrustZone.
    - Investigation on critical instructions related to TrustZone operations.
    - Test of cache flush operations: overhead and effect.
    - Different discussion based on ARMv8-A and ARMv8-M structures.

# Related Work

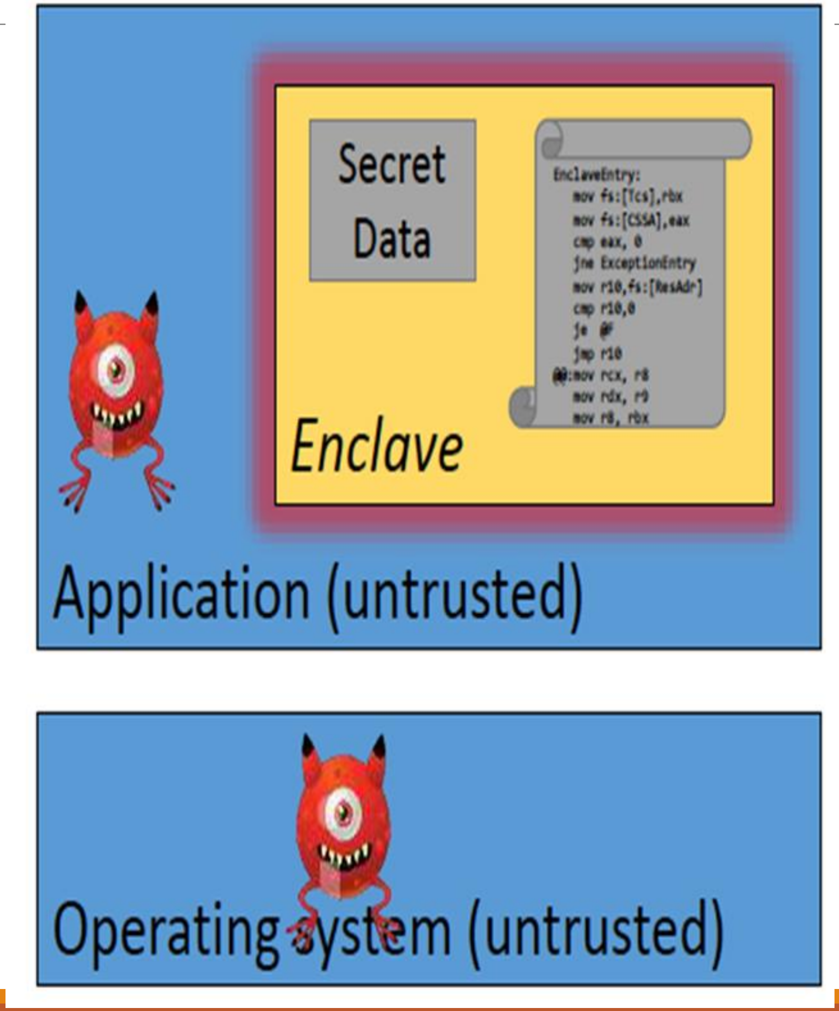
- Side Channel Attacks
  - LLC based side-channel attacks: Flush+Reload, Prime+Probe
  - Effectiveness of LLC based side-channels





# Related Work

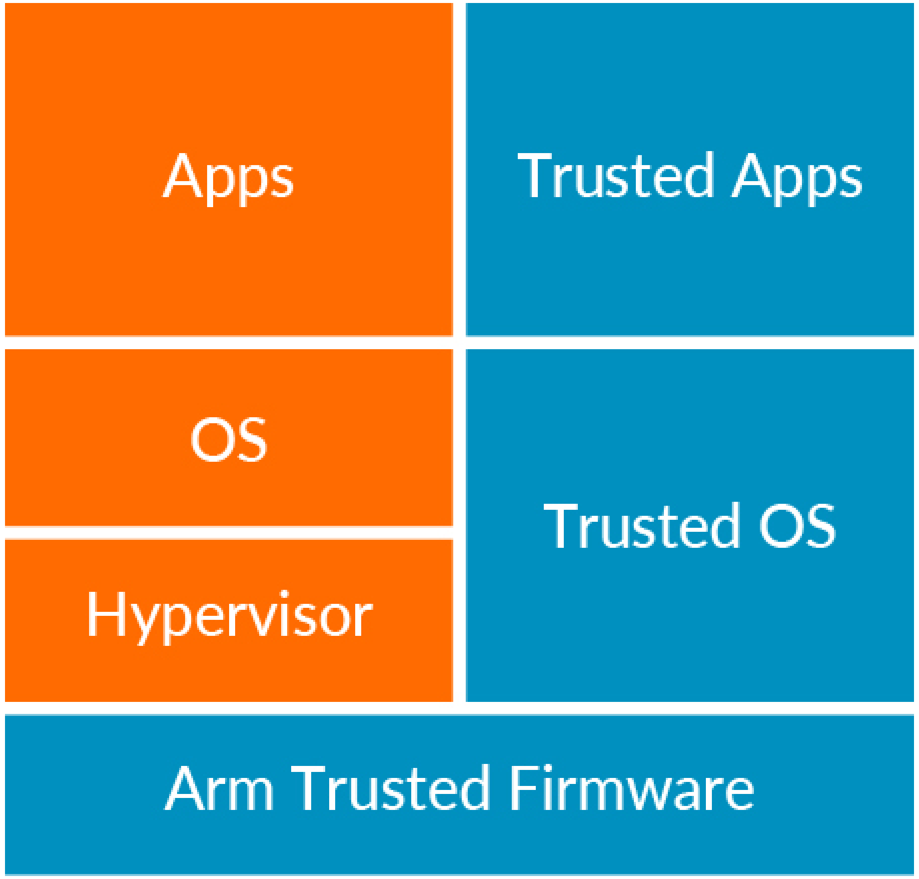
- Security Design and Protections
  - Hardware Solution: Intel SGX, ARM TrustZone
    - Hardware isolation for an enclave
    - New instructions to establish, protect
    - Call gate to enter
  - Remote attestation
    - Processor manufacturer is the root of the trust
  - Prime+Probe Attack: March, 2017
    - Target to DRAM



```
EnclaveEntry:
   mov fs:[Tcs],rbx
   mov fs:[CSSA],eax
   cmp eax, 0
   jne ExceptionEntry
   mov r10,fs:[ResAdr]
   cmp r10,0
   je  @F
   jmp r10
@@:mov rcx, r8
   mov rdx, r9
   mov r8, rbx
```

# Related Work

- ARM TrustZone
  - Based on ARM Cortex-A and Cortex-M series
    - Privileged instructions to call entry/exit
    - Light-weighted comparing with other protection
  - ARM helps in creating Trusted Execution Environments (TEE)
  - Cache Problems
    - ARM Cortex-A series
    - ARM Cortex-M series (ARMv8-M)



# Related Work

- Previous Defense Strategy against Side-Channels
  - LLC-level Protection (memory access control)
  - Cache enclaves (Trusted vs. Untrusted)
  - Scheduler-based solutions
  - Others
- Cache Flush against Side-Channels
  - Benefits: easy to implement, ensure safety
  - Problems: high overhead, not adaptive to every situation

# Cache-Based Security Threats and Attack

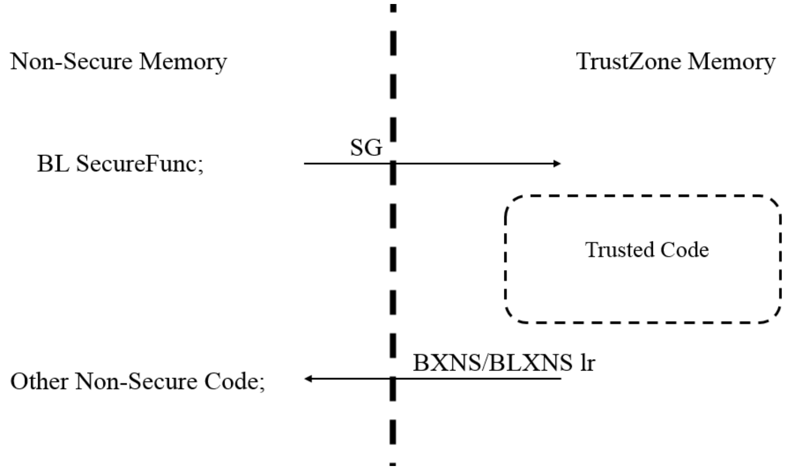
- Overview
- Users’ memory access are not protected by TrustZone – Covert Channel (Sharing resources)
- TrustZone Entry/Exit without Flushing cache – Side-Channel (Malicious collecting access time)
  - Flush+Reload Attack
  - Prime+Probe Attack
- Malicious eavesdropping

# Cache-Based Security Threats and Attack

- Side-Channel Attack Experiment
- Flush+Reload Attack
  - step 0: attacker maps shared library → shared memory, shared in cache
  - step 1: attacker flushes the shared line
  - step 2: victim loads data while performing encryption
  - step 3: attacker reloads data → fast access if the victim loaded the line
- Prime+Probe Attack
  - step 0: attacker fills the cache (prime)
  - step 1: victim evicts cache lines while performing encryption
  - step 2: attacker probes data to determine if the set was accessed

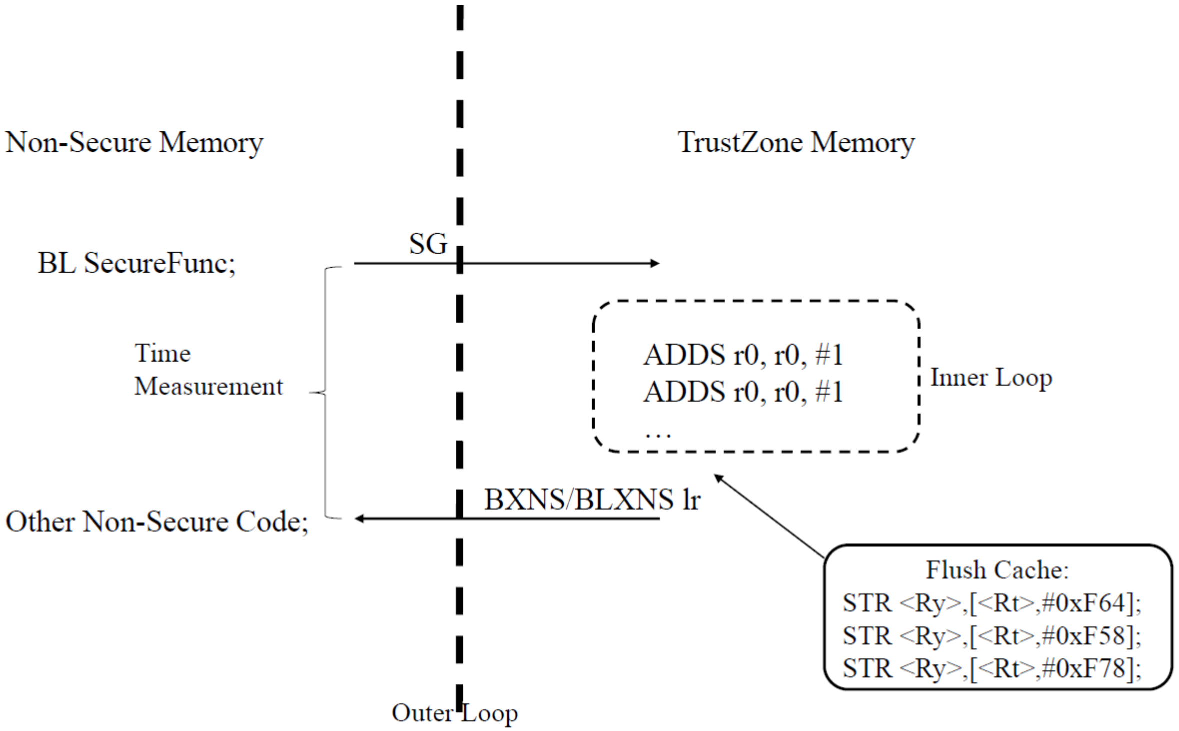
# Design and Implementations

- TrustZone-Related Instructions
  - ARMv8-A
    - Test Environment: ARM Juno r1 Board, with A57 and A53 chips; QEMU as testing benchmark.
  - ARMv8-M
    - Test Environment: ARM Development Kits with Cortex-M4



# Design and Implementations

- Experiments on TrustZone Instructions
- ARMv8-M
- Our experiments on ARMv8-M are using ARM Versatile V2M-MPS2 Motherboard with an ARM Cortex-M4 chip. It offers 8Mb of single cycle SRAM, and 16Mb of PSRAM. It supports the application of different ARM Cortex-M classes, from Cortex-M0, to M3, M4, and M7.



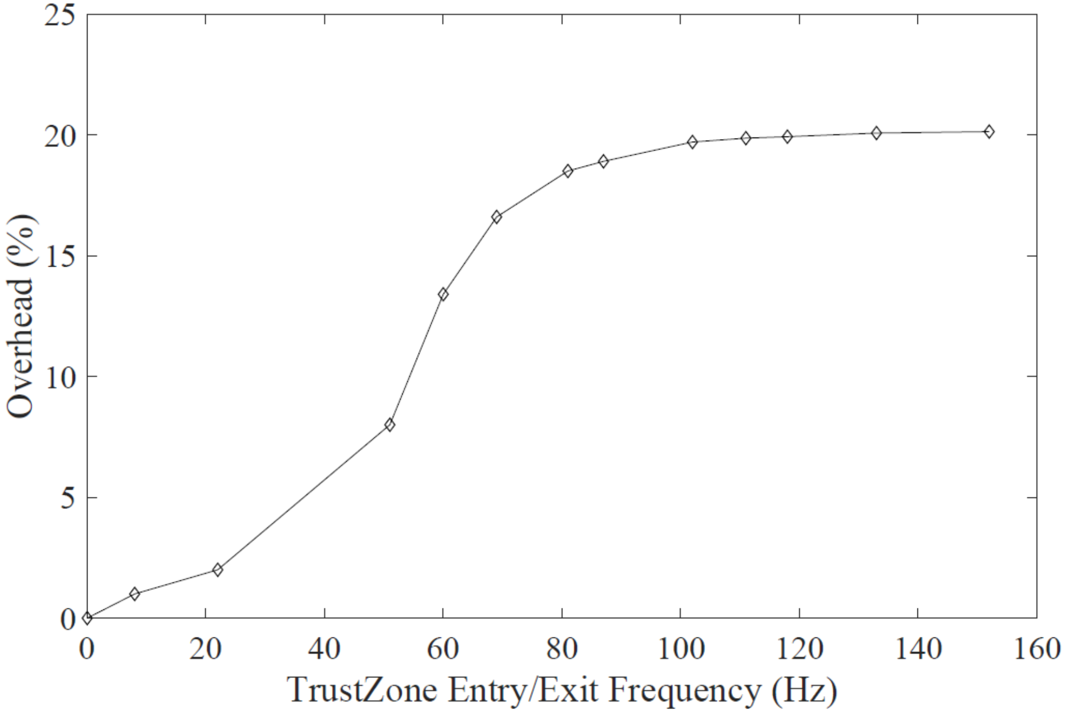
# Experimental Results

- Experiments on TrustZone Instructions
- ARMv8-A
  - We use Ubuntu 16.10 as the normal world OS, with 26 processes running on background, including the workload we use for testing. We count the smc-related instructions that belongs to TrustZone-related operations, and analyze the attributions of them.

| Type | Percentage |
|---|---|
| Non-secure to Secure Test R/W | 2.87% |
| Secure to Non-secure Test R/W | 2.91% |
| Others (Access from Background) | 0.01% |

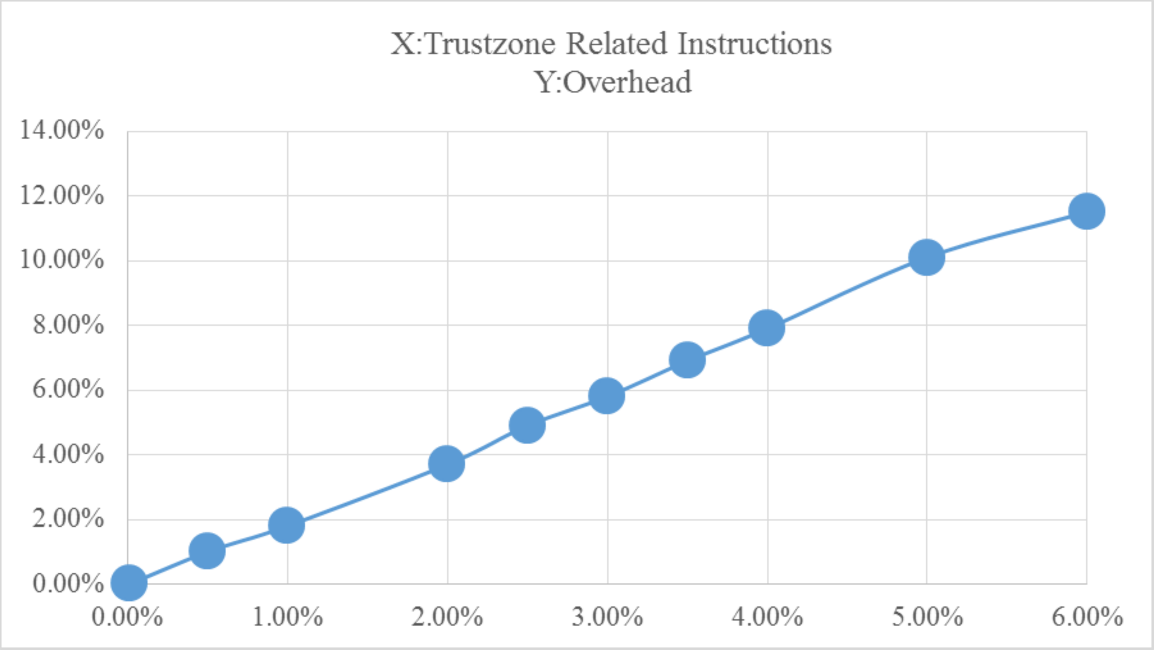
# Experimental Results

- Experiments on TrustZone Instructions
- ARMv8-A
- With every smc-related instruction, we operate Flush on cache.



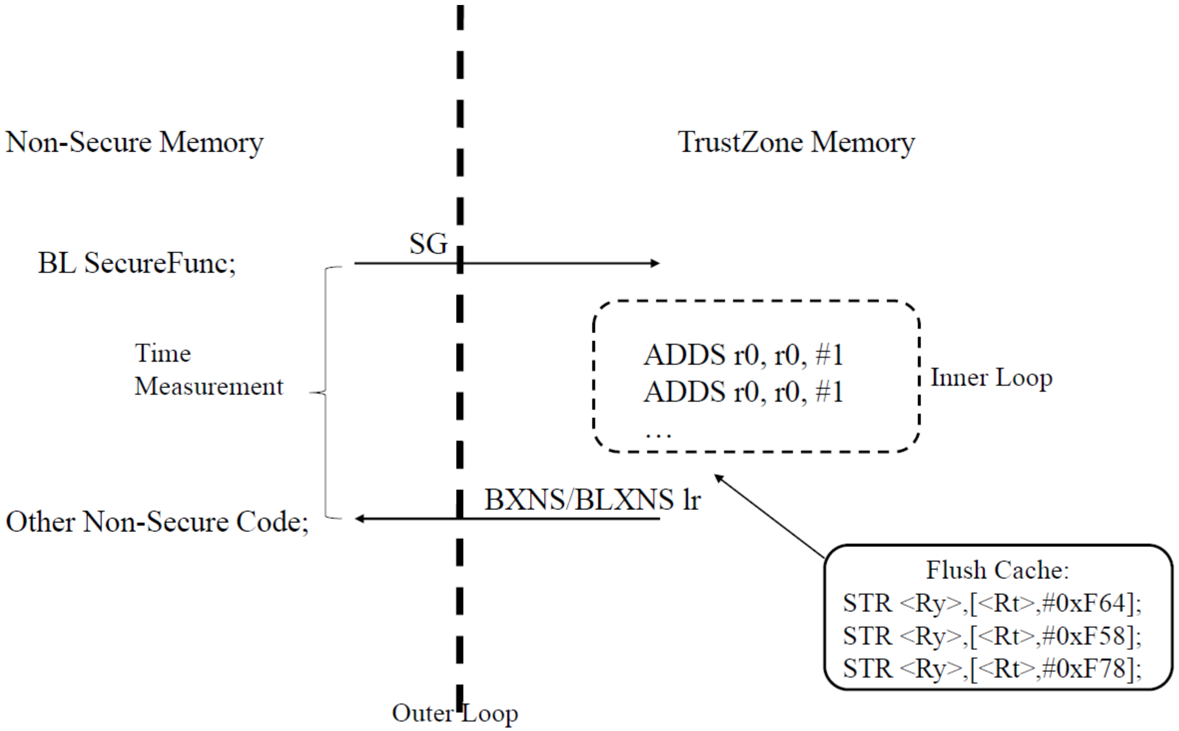
# Experimental Results

- Experiments on TrustZone Instructions
- ARMv8-A
- We change the overall percentage of smc instructions and see the overhead difference.



# Experimental Results

- Experiments on TrustZone Instructions
- Cortex-M
- Our experiments on ARMv8-M are using ARM Versatile V2M-MPS2 Motherboard with an ARM Cortex-M4 chip. It offers 8Mb of single cycle SRAM, and 16Mb of PSRAM. It supports the application of different ARM Cortex-M classes, from Cortex-M0, to M3, M4, and M7.



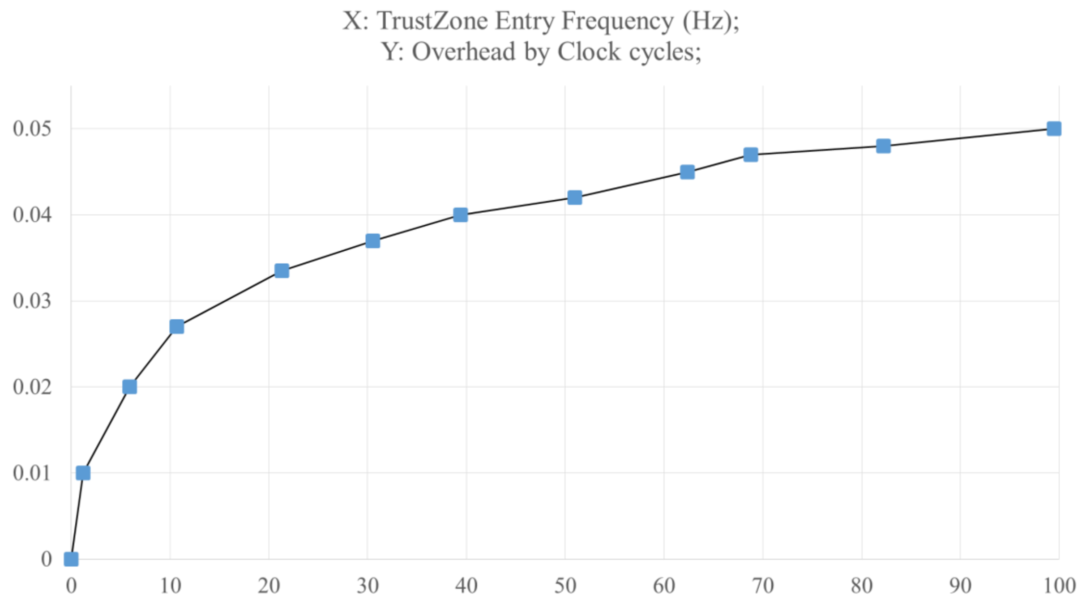
# Experimental Results

- Experiments on TrustZone Instructions
- Cortex-M
- Using Testing Program as shown above.

| Operation | Direction | Cost on Average (Clock Cycles) |
|---|---|---|
| **SG** | Non-Secure to Secure | 3.5 |
| **BXNS/BLXNS** | Secure to Non-Secure | 5.2 |

# Experimental Results

- Experiments on TrustZone Instructions
- ARMv8-M
- We change TrustZone entry/exit frequency by setting different parameters in inner and outer loop. The overhead can be limited to less than 5%.



# Evaluation

- On the cost-effectiveness balance of defending by Flush operations
  - Flush operations are necessary, but they cost much;
  - We can never wipe out the risk, but can cut down bandwidth;
  - Adaptive strategy can be used to keep the balance of performance and effectiveness;
  - Even better on ARMv8-M chips.

# Evaluation

- On TrustZone related instructions
  - Most of the apps and users are not ‘making use of’ TrustZone features;
  - On IoT devices, TrustZone is not costing much resources;
  - It is possible to move some of the hardware/software security design into TrustZone surface;
  - Cortex-M series chips perform better than Cortex-A series chips.
  - On Cortex-A series chips or x86 chips, cache flush operations are just some instructions with privileges. However, the case are different on ARMv8-M. The allocation of a memory address to a cache address is defined by the designers of the applications.
  - Because of the special structure of ARMv8-M, the cache Flush operations are sets of DSB (Data Synchronization Barrier) operations, with address-related instructions.

# Future Work

- Implementations and Experiments
  - Design and implement a defense framework based on ARMv8-M.
  - Test the performance of defense framework using some benchmarks, and optimize the framework to good effectiveness and lower overhead.
  - Port defense framework to new ARMv8-M boards: M23 and M33 series chips.
- Theory Work
  - Study adaptive control method in theory to match the experimental results, and predict the optimal solution of best adaptive control in defense.
  - Investigate entropy theory based on experimental results, predictions and related theory.
  - Discuss performance of implemented defense framework in theory, and try to have theoretical conclusion on defense against cache-based attack.

# Conclusion

- Cache-based attack are new focal point on security design, with risks of leaking information through side-channel and covert channels.
- Flushing cache is effective to cut down the risk, but with high performance overhead, and sometimes not affordable.
- On IoT devices, the performance of connecting with TrustZone can be better, which brings the possibility to making use of TrustZone.

# Thank you!

Naiwei Liu, UTSA ICS Lab, Naiwei.liu@utsa.edu

Ravi Sandhu, UTSA ICS Lab, ravi.sandhu@utsa.edu

Meng Yu, Roosevelt University, myu04@Roosevelt.edu