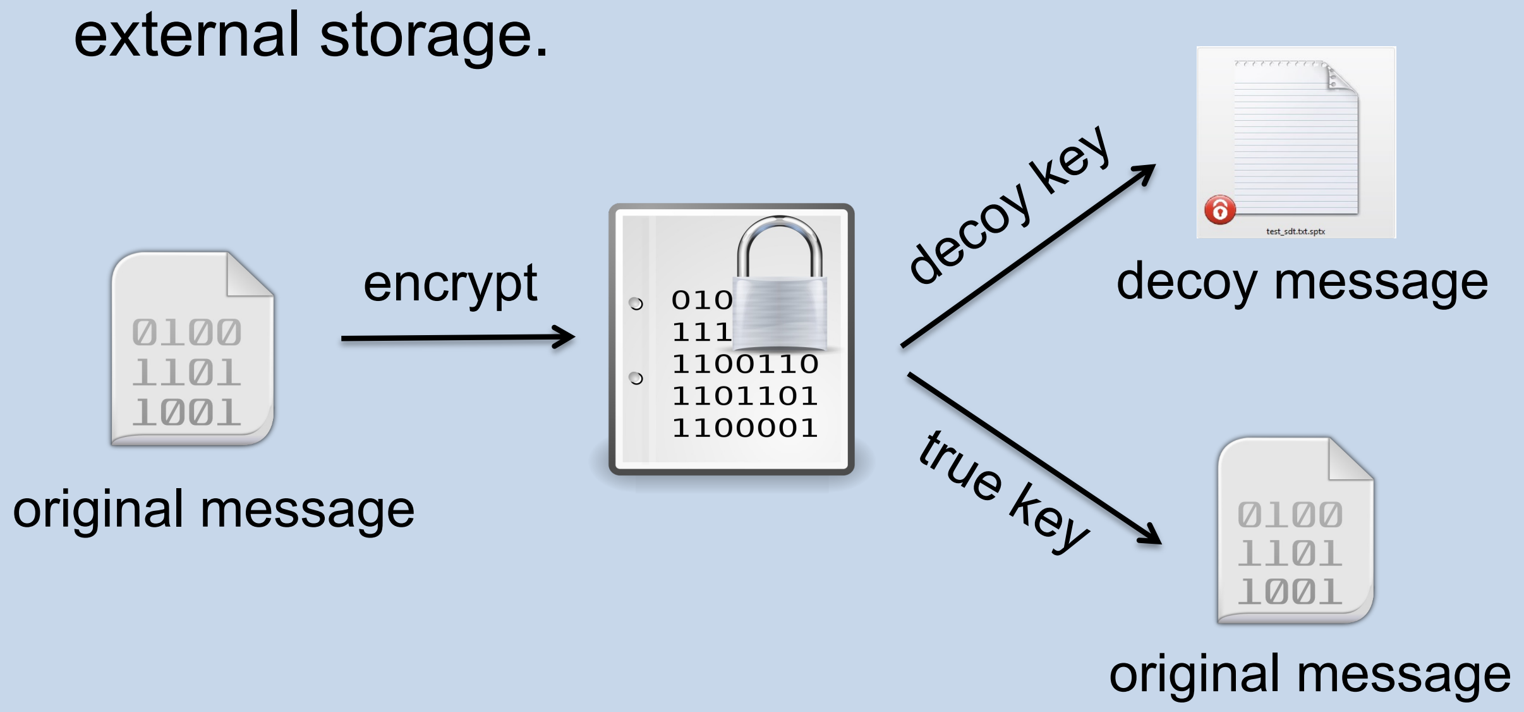
FSPDE: A Full Stack Plausibly Deniable Encryption System for Mobile Devices

Michigan Technological University 1885

Jinghui Liao, Niusen Chen, Lichen Xia, Bo Chen, and Weisong Shi

Introduction

Mobile computing devices have been used broadly to store, manage and process critical data. To protect confidentiality of stored data, major mobile operating systems provide full disk encryption, which relies on traditional encryption and requires keeping the decryption keys secret. This however, may not be true as an active attacker may coerce victims for decryption keys. Plausibly deniable encryption (**PDE**) can defend against this *coercive attacker* by disguising the secret keys with decoy keys. Leveraging the concept of PDE, we have developed a full-stack mobile PDE system which ensures plausibly deniability in both the internal memory and the external storage.

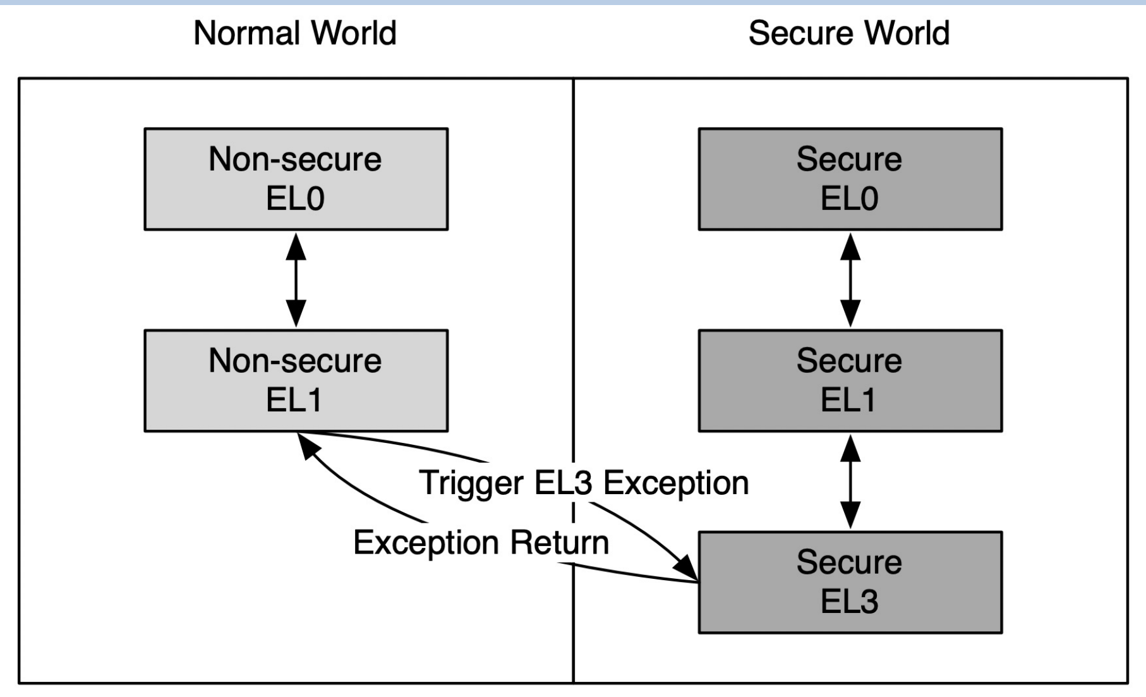


Plausibly Deniable Encryption (PDE)

Background

Mobile devices are typically equipped with Arm processors and NAND flash memory. The Arm processor commonly supports TrustZone, an Arm implementation of Trusted Execution Environment (TEE). The TrustZone can provide a secure environment within a system-on-chip for running trusted applications.

TrustZone

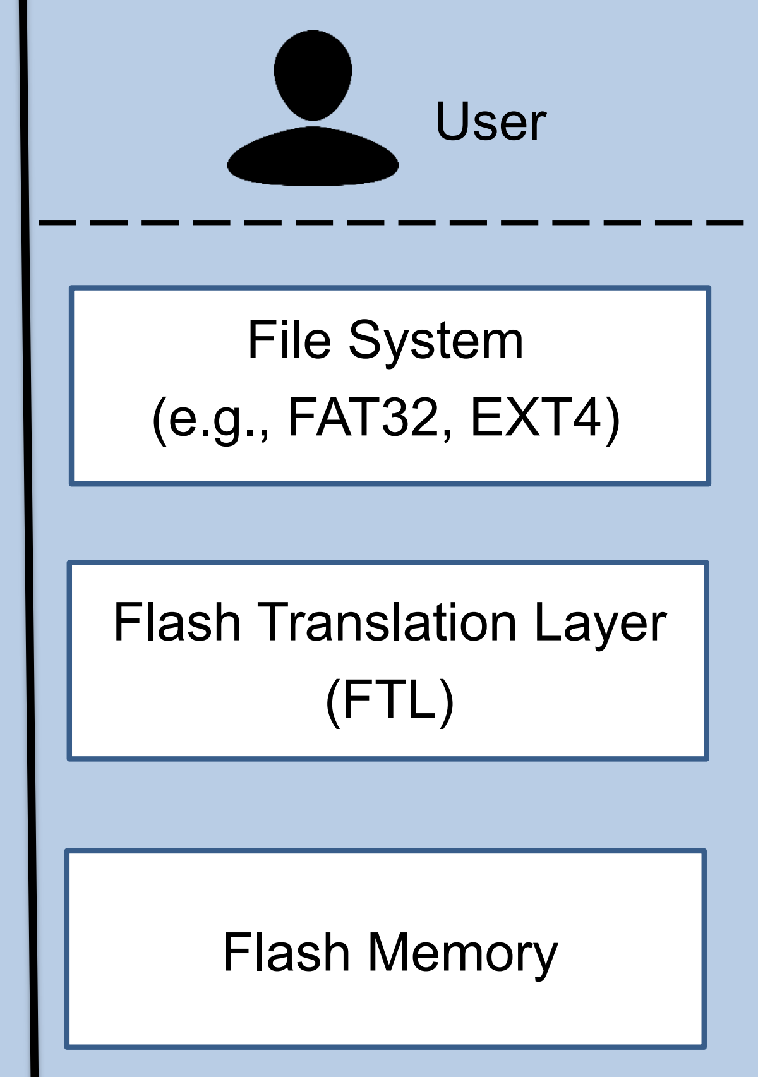


Special Characteristics of NAND Flash

- Over-writing data stored on a page requires first erasing the entire encompassing block
- Writing is performed on a page basis, but erasure is performed on a block basis
- Each flash block can only be programmed/erased for a limited number of times (e.g., 10K)

How to use NAND Flash

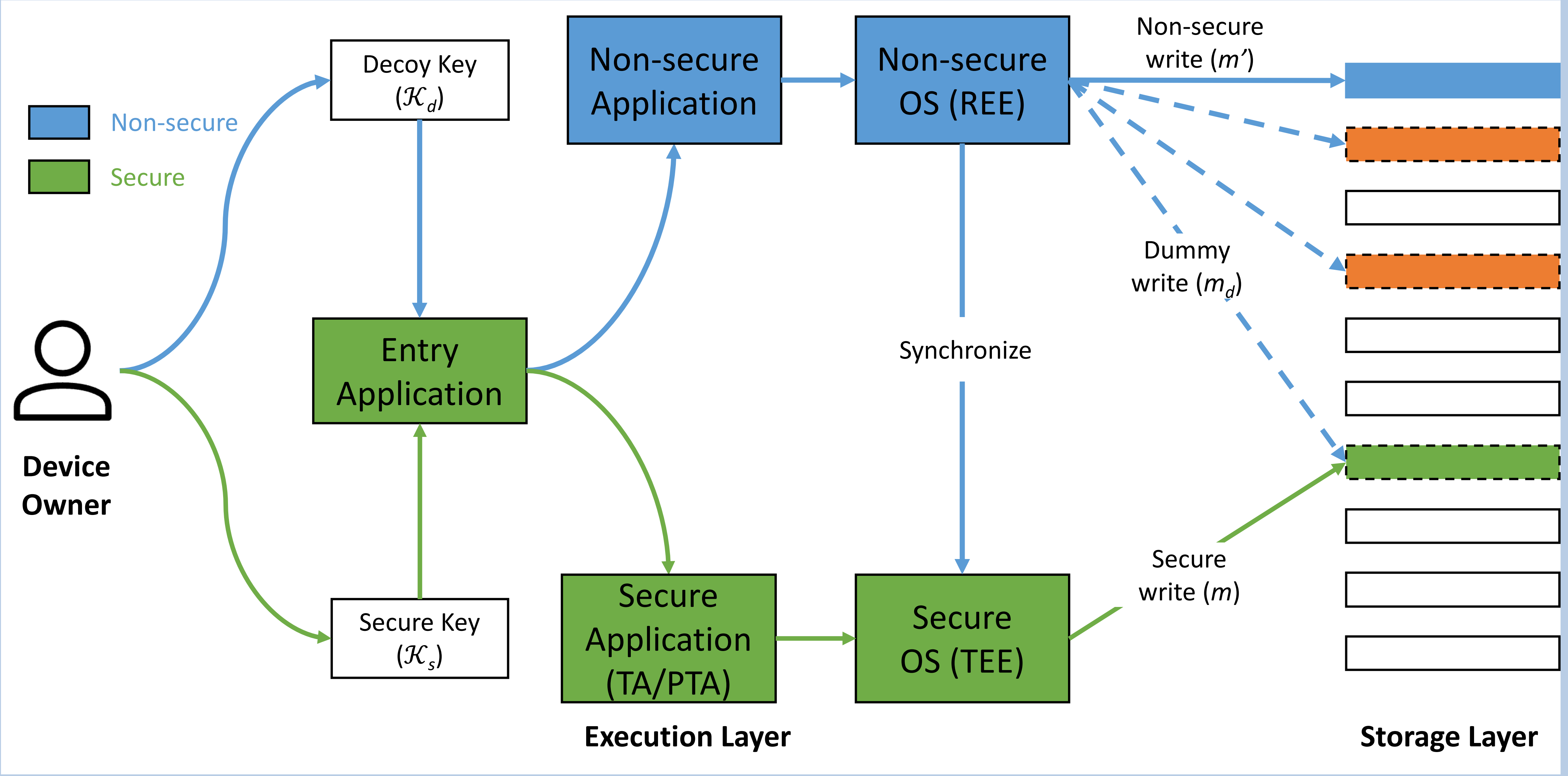
- The most popular form of using flash memory is to emulate it as a block device
- This architecture is commonly found in flash memory cards like eMMC cards, SD cards
- Flash Translation Layer, FTL, is introduced between the block device and the raw flash
- FTL transparently handles unique nature of flash
- FTL implements special functions like garbage collection, wear leveling, and bad block management



Adversarial Model

- The adversary is capable of capturing a victim user, along with his/her computing device, at various intervals, i.e., a *multi-snapshot* adversary
- The adversary can have access to both its external storage and internal memory
- The access to the external storage can cover different storage sub-layers including application, block device, and raw flash memory
- The adversary may perform reverse engineering over the binary files in the victim device
- The adversary may coerce the victim user to reveal the decryption key for data
- The adversary is computationally bounded

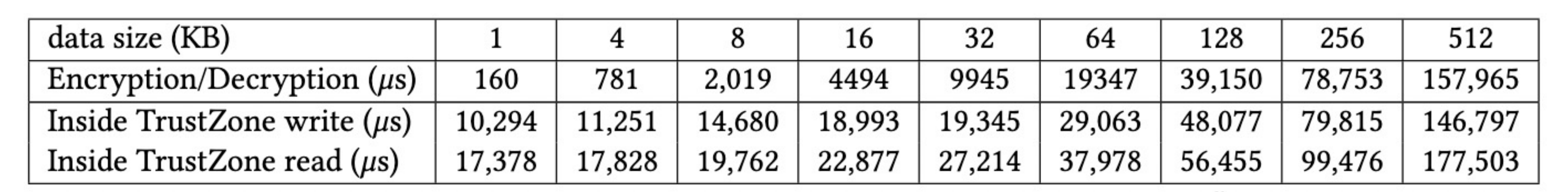
Design of FSPDE



Implementation and Evaluation

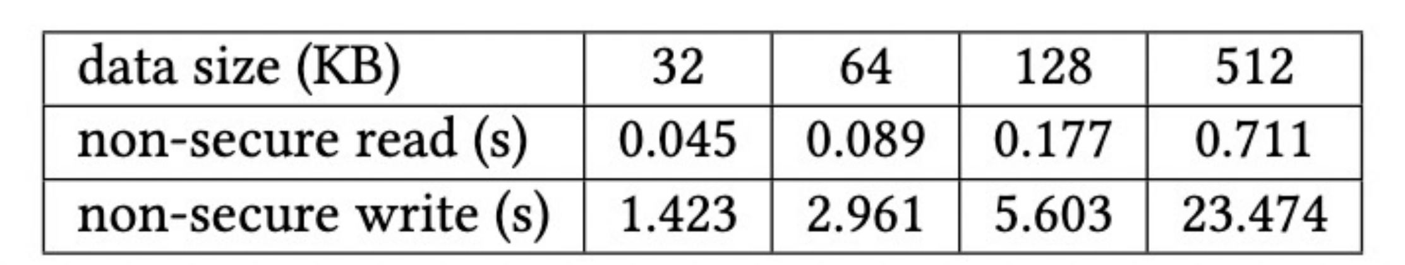
- We evaluated FSPDE using a Raspberry Pi 3 Model B development board. This SoC has built- in support for TrustZone technology. On the software front, we have implemented OP-TEE, version 3.19
- The non-secure software realm was anchored in the Linux kernel with version rpi3-optee-5.17
- For external flash memory storage, we used another electronic board LPC-H3131. The open-sourced flash controller OpenNFM was modified and ported to the LPC-H3131, converting it to a regular flash storage device

secure read/write in the TrustZone secure world:



data size (KB)	1	4	8	16	32	64	128	256	512
Encryption/Decryption (μs)	160	781	2,019	4494	9945	19347	39,150	78,753	157,965
Inside TrustZone write (μs)	10,294	11,251	14,680	18,993	19,345	29,063	48,077	79,815	146,797
Inside TrustZone read (μs)	17,378	17,828	19,762	22,877	27,214	37,978	56,455	99,476	177,503

non-secure read/write in the normal world:



data size (KB)	32	64	128	512
non-secure read (s)	0.045	0.089	0.177	0.711
non-secure write (s)	1.423	2.961	5.603	23.474

Acknowledgments

This work was supported by National Science Foundation under grant number CNS-2313139, CNS-1928349, and CNS-2225424.