Thesis: Outline

Oberon Swings

April 18, 2022

Contents

Chapter 1

Introduction

- Smartphones everywhere
  - Everyone has one
  - Interchangeable with PC
- Sensitive data
  - Lots of traffic
  - Personal data stored
- Comparable to IoT
  - Hardware similarities
  - Security features (lot less than PC)

Smartphones are everywhere, being used for more and more sensitive data. Hacking into these devices should be made as hard as hacking into someone's personal computer because for many those two have become interchangeable.

1.1 Problem statement

- IoT security
  - Minimize overhead
- PC functionality
  - Banking, e-Health and mails
- Missmatch
  - Sensitive data requires good security
  - Functionality is pushed but security lags behind

The hardware in smartphones is comparable to that of IoT devices, it is more powerfull in many occasions but the design principles are often the same. The problem with this is that IoT devices are not very secure, smartphones are in that sense lagging behind on security compared to how they are used (banking, health and identification applications).

1.2 Contributions

- Reproduction of paper
  - Replicated solution as closely as possible
  - Evaluation about results compared to original
- Open source code (proof of concept)
  - Enable easier reproduction/verification in the future
- Extra experiment measurements
  - Comparable experiments (to be able to compare)
  - More elaborate experiments (to allow better decision making)
- Comparison with similar solutions
  - Overview of comparable papers (pros and cons)
  - Weak points and strenghts of the reproduced paper compared to the others

Major producers of smartphone chips are adding hardware support for security (Intel SGX, ARM TrustZone). The focus of this thesis lies in using ARM TrustZone to achieve a secure open platform from a smartphone equiped with ARM TrustZone.

1.3 Outline

In the next chapter more background information about among other things ARM TrustZone and Remote Attestation will be given. In the third chapter the methods secure applications will be explained. In the fourth chapter the goal and outcome of the experiments will be made clear. The final chapter will conclude this thesis informing the reader about limitations of this work and possible future directions of research.

Chapter 2

Background

The smartphone that will be used is a PinePhone which is equiped with ARM TrustZone, it also comes with a component which can be used as Root of Trust to make secure boot possible.

2.1 Remote Attestation

- Goal
  - Verify integrity
- How it works
  - Verifier
  - Prover
  - Proof
- Assumptions
  - Trusted third party
  - Secure keys

Remote attestation allows a device to prove to an external verifier that the software running on it is not tampered with. This attestation can go a lot further than this by for instance also checking the data structures on the device to make sure these are logical.

2.2 Trusted Execution Environment

- Execution
  - Isolation

  - Authentic code
  - Runtime integrity
  - Strict interfaces
- Trust
  - Static
  - Dynamic
- Security
  - Data separation
  - Sanitization
  - Control of information flow
  - Fault isolation

A Trusted Execution Environment is a secure, integrity-protected processing environment, consisting of memory and storage capabilities.

2.3 ARM TrustZone

- Normal World
  - Rich OS
- Secure World
  - Trusted Kernel
  - NS-bit
  - Secure Configuration Register
- Peripherals
  - TZ Address Space Controller
  - TZ Protection Controller (interrupts)

ARM TrustZone is ARM's implementation of a TEE. This is achieved by having a secure and normal world in the System on Chip.

2.4 PinePhone

- Open source
  - –
- Linux
  - –
- ARM TrustZone
  - –

The PinePhone is an open source smartphone which supports Linux as operating system which adds to it's openness.

2.5 Secure boot, trusted boot and remote attestation for ARM TrustZone-based IoT Nodes

- Solution
  - Overview
- Trusted Boot
  - Trusted load phase
  - Attestation during boot
- Remote attestation
  - Trusted execution time
  - Pagebased approach

Chapter 3

Method

The main goal of this work is to achieve a secure open platform on the hardware.

3.1 Detailed Problem

- Lots of functionality
  - Sensitive data
  - Mobile computing
- Performance driven
  - All resources to functionality
  - Little room for security
  - Extensive security measures needed
- Hardware security solution
  - Security embedded in hardware
  - Better for performance
  - Correct implementation required

3.2 System Model

- Open platform
  - Multiple software providers
  - Platform owned by user
- Secure software execution

  - Software isolation
  - Secure data storage

The system model describes an open platform with no or minimal trust among stakeholders.

3.3 Attacker Model

- Physical access
  - Threats
  - Vulnerabilities
- OS/Firmware attacks
  - Threats
  - Vulnerabilities
- Software attacks
  - Threats
  - Vulnerabilities

The attacker has physical access, can launch OS/firmware and software attacks. The Trusted Platform Module is assumed to be tamper resistent.

3.4 Solution

- Secure boot
  - Root of Trust
  - Chain of Trust
  - Secure starting point
- User attestation
  - Integrity (control flow, data structures, ...)
  - Authenticity (code, ...)
- Trust
  - Execution
  - Data protection

Ideally the device is started with secure boot, this makes sure the SW is started from a known secure state.

During operation the user should be able to attest whether their device is still in a secure state.

This can be done using a TA that makes measurements on their device and reports back to them.

These measurements are checking the integrity of the code section of the running applications and OS.

Chapter 4

Implementation

4.1 Attestation TA

- Trusted Application
  - Hash NW memory pages
  - Store reference values in secure memory
  - Make comparison and notify user
- NW OS interaction
  - Retrieve address of memory page
  - Translate into physical address
  - Provide datastructures to TA
- Extensions
  - Notify user using trusted IO
  - Decrease dependency on NW OS
  - Attest more than just code pages (data structures, invariants,...)

The attestation is mainly implemented in a Trusted Application. This is done to allow the Secure World to store measurements in the secure memory and have access to the Normal World memory. Ideally the TA doesn't need to rely on the NW OS (Linux) for the memory addresses but this is the starting point for the implementation.

Chapter 5

Experiments

These experiments are based on a proof of concept of the attestation application.

5.1 Performance

The TA responsible for the attestation of the NW will be executed in a variety of circumstances and the execution time for measuring and attesting will be taken into account.

5.2 Performance Evaluation

The performance should give an insight on how often these measurements can be executed to avoid too much overhead but still be able to have good security measures.

5.3 Security Properties

The secure boot process makes sure that TrustZone works as intended which should give confidence in the belief that secure execution of trusted applications is guaranteed.

With secure execution of TAs guaranteed the Secure World can give similar guarantees as a remote attestation server would give. This implies that the attestation can happen on the device itself while still having strong confidence about the validity.

5.4 Security Evaluation

Chapter 6

Conclusion

To achieve secure execution on the PinePhone some requirements need to be met. One of these requirements is that a chain of trust is achieved which is done using secure boot in this case.

6.1 Related work

- Secure boot, Trusted boot and remote attestation for ARM TrustZone-based IoT Nodes
- DAA-TZ: An Efficient DAA Scheme for Mobile Devices Using ARM TrustZone
- SecTEE: A Software-based Approach to Secure Enclave Architecture Using TEE
- TZ-MRAS: A Remote Attestation Scheme for the Mobile Terminal Based on ARM TrustZone
- TrustShadow: Secure Execution of Unmodified Applications with ARM TrustZone

6.2 Comparison of Approaches

- Effectiveness
  - Reached goal
  - Defends most variety of attacks
  - Stronges security guarantees
- Assumptions
  - Least assumptions
  - Most realistic

6.3 Future Improvements

- Best approach
  - Overview
  - Reasoning
- Weaknesses
- Possible improvements
  - inspiration from Lightweight and Flexible Trust Assessment Modules for the Internet of Things
- Different solutions

To allow the user to attest their device it is important that Trusted I/O is used to inform the user about the outcome of the attestation process.

The attestation application can be seen as one module that can be accompanied with a variety of different modules to increase the amount of checks that can be executed to check more possible attacks/ vulnerabilities.

Chapter 7

Discussion

- Solution overview
  - Reproduction
  - Thorough understanding paper solution
  - Extensions
- Shortcomings
  - Unaccounted for attacker possibilities
  - Unaccomplished desired guarantees
- Positives
  - Open source code
  - Thorough comparison

Currently only the code in the textsection is being attested which is far from enough to trust the execution of a device.

The datastructures of the applications running in the NW or the OS should also be taken into account.

The state of the memory could be restricted to always comply with certain invariants that can be checked.

Check whether the method applied is sufficient to protect against the attacker model and if not explain why.