Chapter 1

Introduction

Smartphone functionality. Smartphones have become an essential part of our daily lives and everyone is assumed to have one of their own. These devices can be spotted everywhere, people use them at home, on the bus or even at work. In most cases these phones are only occasionally used for text messaging or calling but very often for reading e-mails, surfing on the web or even for services like e-banking. Because of this wide range of functionality some people may even replace their personal computer by a smartphone entirely. The success of smartphones lies in their ease of use and always being accessible, people just carry them in their pocket. Besides the user having access to their phone all the time, a smartphone also has or could have access to the internet all the time. It is known that the internet is the gate to lots of services which are perceived as necessities lately.

IoT security. Devices that make lots of connections on the go while also utilizing online services for which sensitive data is required may introduce security vulnerabilities. The fact that people are using their smartphones for services like online banking or even consulting health related reports implies that some sensitive data must be stored on these devices or at least present while they are interacting with it. While this data should be protected very well it is present on an Internet of Things (IoT) device for which security solutions and standards present today are not adequate in terms of protection against the existing threats. The hardware similarities between smartphones and IoT devices are more prominent than one might expect, this is because the architecture of smartphones stems from that of IoT devices. The main issue here is that IoT devices are designed for performance, they only have a small number of tasks but these need to be executed as fast or as energy efficient as possible. This also applies to smartphones because while the functionality of a smartphone is close to that of a personal computer the hardware is not. This weak link in the hardware gives rise to multiple different attack strategies that adversaries can utilize to steal sensitive data from smartphone users. Lately improvements have been made in this area by

extending the processors of these devices with features that make it possible to setup a Trusted Execution Environment (TEE) on them. A TEE can increase the security of an IoT device, this is often achieved by utilizing core security services for critical operations. Examples of these critical operations are cryptographic operations, storing data in secure memory or accessing Input and Output (I/O) through trusted paths.

Our contributions. This thesis is based on the reproduction of existing work namely [**?**], the solution of this paper is replicated as closely as possible. The proposed system attempts to increase the security of a smartphone by having the TEE attest the code of the user and Operating System (OS) space. To allow for a direct comparison between the performance achieved in the paper and our implementation some experiments in the paper are redone. After this comparison more elaborate experiments are executed to give a better view on the trade offs between performance and increased security. To allow others to easily reproduce or review the work that has been done all code and experiment setups are made available in open source. The performance is of course only a small aspect of the analysis of the solution, to do the security analysis, this work (and the solution of the initial paper) are compared to similar work. In this comparison it is discussed whether the solution is the most effective out of the existing ones, which other alternatives may achieve more in terms of defended attacks or achieved security guarantees. Finally it is evaluated which type of solution is the most promising as the direction for future work based on the comparison with the similar alternatives.

Outline. In the next chapter more background information about among other things Remote Attestation and ARM TrustZone will be given. In the third chapter the methods to solve the problem will be explained. In chapter four and five the implementation of the attestation program are elaborated upon and the outcome of the experiments will be made clear respectively. The sixth chapter will conclude this thesis informing the reader about related work and future improvements. The final chapter will give a discussion on the presented work.