Embedded devices are ubiquitous nowadays and interact with us on a daily basis. They are not only in charge of low impact actions like controlling household appliances or tracking your fitness level but also part of critical infrastructure such as traffic lights, water level control of dams or delivering electricity to your home. The large scale of their adoption, connectivity, and complexity are several reasons why securing these devices is challenging and remains an unsolved problem.

Reverse engineering is the process of understanding how a device or system works internally without accessing design documents or source code. Through reverse engineering, defenders and testers can discover vulnerabilities in these systems. This allows developers to fix the vulnerabilities before an attacker has the opportunity to exploit them, thus improving the overall security of these systems.

In the first part of the dissertation we explore reverse engineering aimed at firmware. Namely, first we improve reverse engineering by proposing a lightweight static analysis and show how to detect parsing and parser-like code inside binary code. Because this type of code deals with untrusted user input, a vulnerability located here has high changes of being exploitable and cripples the security of the device. Second, we propose a heavyweight analysis designed to solve intricate indirect control flow transfers inside firmware code. In firmware code, one source of these intricacies is C switch statements that are compiled to a jump table-based implementation in binary code. This leads to data and code interleaving which is a hurdle for reverse engineering tools such as disassemblers and decompilers, but also for binary security deployment tools.

In the second part of the thesis, we focus on a form of reverse engineering that is tightly coupled with the hardware. We show that software defenses such as instruction duplication are ineffective and even harmful in the presence of hardware attacks such as power glitching. Furthermore, the ineffectiveness of these defenses depends on the hardware runtime configuration which can be revealed by reverse engineering. Lastly, we reverse engineer the error correcting codes (ECC) embedded inside various popu-
lar memory controllers by extending fault injection attacks and leveraging cold boot
attacks. Initially an ECC equipped system was thought to be immune to Rowhammer
attack. With the knowledge gained through reverse engineering we show that this is
not the case, therefore exposing the dangerous false sense of security. In addition,
our result sprouts a new research direction focused on defenses against this class of
attacks.

In conclusion, in this dissertation we show not only how reverse engineering at the
boundary between hardware and software may help improve the security of computer
systems, but also advance the state of the art of hardware reverse engineering.