

Industrial Microprocessor Design

Paper 2

Victoria Steele

Table of Contents

[1. Problem Statement 2](#_Toc404930424)

[2. Nomenclature 2](#_Toc404930425)

[3. AssumptionS 2](#_Toc404930426)

[4. Results and Analysis 2](#_Toc404930427)

[4.1 Background on Security for Microcontrollers 2](#_Toc404930428)

[4.2 Requirements for Appliances Pertaining to Security 2](#_Toc404930429)

[4.3 Conclusion 2](#_Toc404930430)

[5. References 2](#_Toc404930431)

# Problem Statement

*Is it worth paying more for a microcontroller with enhanced security features for consumer applications, such as appliances? If so, what enhancements are useful, and how much is security worth? If not, why not? Are there any liability concerns for the company that manufactures and sells consumer devices?*

# Nomenclature

|  |  |
| --- | --- |
| Abbreviation | Definition |

# 

# AssumptionS

* Knowledge of appliance controls does not need to be sited if information is inherent to developer.
* PC security is beyond the scope of this paper

# Results and Analysis

## Background on Security for Microcontrollers

Security within electronic applications has become an increasing concern over the past decades. This paper will specifically address microcontrollers which have a CPU, ROM, RAM, and peripherals all in one chip. PC security is beyond the scope of this paper as it incorporates many levels of malware topics. As technology becomes more advanced, so too, do invasive and noninvasive hacking techniques. The ways in which to breach each device depend upon a myriad of features from the chip packaging to the operating system. For example, devices which contain an embedded OS with a command prompt are vulnerable API attacks whereas those without an operating system which do not have a user interface generally are not. Historically speaking, microcontrollers are susceptible to attacks from two fronts: firmware and hardware. These techniques include microprobing, software attacks, eavesdropping, and fault generation (copy protection). These will be discussed in the following sections.

When addressing microprobing, the silicon technology (Fuse/Antifuse vs Floating Gate) used can leave the device more or less vulnerable to attacks (lecture). As described in “Protecting Your Electronic Wallet against Hackers”, there are several methods of invasive tampering including Device Cross Sectioning, Top Down Planar Inspection, and Voltage Contrast. Device Cross Section is more of a concern at an architectural level whereas Top Down Planar Inspection and Voltage Contrast can reveal the contents of the memory cells. This being said, the physical technology used can provide a measure of protection to these types of reverse engineering. Floating Gate technologies tend to be more resistant than Fuse/Anti-fuse due to the fact that there is no change to physical change to the silicon during the programming process and the contents can be easily disturbed by tampering. For example, Voltage Contrast can be conducted with antifuse whereas floating gate contents will be disrupted and the contents will be erased after one attempt which is desirable. Invasive attacks are, generally speaking, more difficult than noninvasive attacks and require specialized equipment; however, these items can be purchased through the second hand semiconductor market and should not be ruled out (copy protection).

Eavesdropping, software attacks, and fault generation will all be categorized as firmware attacks for this paper. As mentioned in “Intro to Embedded Reverse Engineering for PC Reversers” one of the primary means of breaking in is through or during firmware updates. In order to get into firmware updates, hackers can look for compression streams and messages decrypted on the PC side. Additionally, bootloader images can be distinguished and corrupted as described in “Heart of Darkness – exploring the uncharted backwaters of HID iCLASS security”. Of particular note for embedded devices similar to the one’s used at GE Appliances are attacks through the UART and JTAG interfaces. UARTs, I2C , and SPI ports are a common means of communication between boards and programmers. Also as described in the aforementioned paper, JTAG interfaces are typically used for reprogramming boards along with other simple connectors such as the 6 pin connector used to reprogram the PIC. On the same topic, most microcontroller manufacturers provide a software suite for reprogramming chips such as the PIC reprogramming and the Flash Development Tool by Renesas. As demonstrated in the HID iClass Security paper, these can be emulated and the embedded code can be dumped as the hacker sees fit. As noted in “Hacking the PIC”, the security bits can even be erased using reflected UV light. At this point, it is a matter of reading and reconstructing the disassembly, and hardware, assembly, and compiler manuals can be readily downloaded from the internet and can provide with all the information needed. Dev kits can also be purchased for a fairly inexpensive price, facilitating experimentation. Lastly, as addressed during class, fault generation and eavesdropping provide noninvasive means to obtain additional information. Monitoring the power supply provides an effective means of breaking security keys. Using the power supply and clock circuits can cause also cause a partial or incorrect initialization of the device (lecture).

Security breaches of both previously mentioned types have resulted in millions lost and numerous safety concerns. The products, companies, and organizations compromised include the Sony Reader, Casio Ex-Word, Kindle, NASA, European Space Agency, AT&T, Microsoft, Target, Play Station, and many more. There are even companies that will reverse engineer components including Mikatech, Russian Semi Research, Chip Works, and Circuit Engineering as provided in the lecture. There is no way to absolutely secure data from hackers, but there are ways to significantly improve security and discourage attacks.

* Measures that can be taken now to secure
  + Encryption keys
  + Erasure on chips
  + Shielding
  + Lock bits for programming
  + Use Floating Gate Technology at the physical layer
  + Hide and obfuscate programming connectors
  + Use SAM modules to store important information rather than the microcontroller
  + For communication, use authentication at both ends
  + Use on-time programmable microcontroller versions

## Requirements for Appliances Pertaining to Security

* History of micro hacking within appliances, reproduction of main boards, etc
* Current Vulnerabilities based on research
  + Supply voltage and clock signal tampering
  + Pull from above,
* What we do
  + Stacia (general)
    - Internet Security and typical techniques relating to database injection etc.
  + Steve (embedded)
  + Scott (hardware perspective)
  + Mark Brian (diagnostics)

## Conclusion

* Is it worth paying more
  + Focus is not primarily on security for appliances
  + Often it’s on features available for the microcontroller
  + As of now GE Appliances specifically does not pay more for additional security features – industry simply would support it since profit margins are so low
  + Investigate independent encryption/decryption engines. Use STM parts for inverters, consider using them for other applications as well.
  + Generally do what you can within the budget to make it more difficult
  + upside for us to the embedded portion is that there is no console, not OS, not even a scheduler, we do use uC/OS- III and Quantum Framework
* Yes/No what enhancements
  + Not much you can do related to invasive attacks, so focus on noninvasive, software security. This is probably the focus anyway since it’s mainly the reproduction of boards for resale purposes.
  + Appropriate measures should be taken in relation to the cost of the product and the cost of replacement products
  + How much is security worth – safety vs security, safety takes a priority – redundancy checks for safety are more critical, short production life, low budget, etc.
  + We do have an internal proprietary communication protocol
  + Might be worthwhile to obscure the chip name and type to add a layer of difficulty to disassembly
  + Don’t have typical programming connectors – it goes to an edge connector which further obscures it for most boards (iClass)
  + Renesas FDT tool offers similar options to the pic example (iClass) and we do have bootloaders
  + Vulnerability is that we have the GE Protocol connectors for Field service and diagnostics
  + Worth defending against as many attacks as economically feasible (Copy protection)
  + LVD for power supply attacks , current analysis, clock signal, data remnants in Volatile memory.
* Liability concerns for a company that manufactures and sells consumer devices
  + Consumer devices encompass a wide variety of products
  + Appliances fall in a certain category and there are liability concerns
  + List liability concerns relating to security
  + OTA updates provide a definite vulnerability (didn’t do this previously – no wifi connected appliances)
  + Currently we have Bootloader images (skochinsky)
  + Several papers read were dated and technology has improved dramatically

# References

[1] **There are no sources in the current document.**