A Secure Embedded Student Attendance Monitoring System for Arduino

Department of Computer Science, University of Puerto Rico, Rio Piedras Campus

Abstract

There has been a recent trend to connect home automation devices to the internet, many of which could be vulnerable to cyberattacks. Our goal was to design an embedded system while considering security measures for the system. We present S.A.M.S., a Student Attendance Monitoring System, whose design process allowed us to explore some of the challenges faced when designing a secure embedded systems [1]. We have implemented a S.A.M.S. prototype using an Arduino UNO and several inexpensive peripherals. Despite the performance and memory limitations of the chosen platform, the system is capable of registering and web-serving student attendance in encrypted form.

The Software

S.A.M.S. main algorithm:
1. do forever:
2. Serve web page (if client available)
3. if 1 is pressed:
4. Read a student id number
5. Get time (using NTP)
6. Encrypt number with time(using RC5)
7. Save to microSD
8. else:
9. do nothing

Due to the small program storage capacity on the Arduino UNO (32K), all parts of the program had to be stripped to their bare minimums. The RC5 encryption algorithm was chosen because of its simplicity and small memory requirements [3]. A tool that helped us to cut down on storage and memory usage was the MemoryFree library [4]. We also had to sacrifice functionality like having a card reader, or displaying the time on the screen due to the lack of space.

What is S.A.M.S.

The main function of the system is to register classroom attendance (student ID) while storing the data securely. The student enters his ID using a keypad, then S.A.M.S. stores the ID plus a timestamp encrypted using the RC5 encryption algorithm. Later, the professor can access the encrypted attendance list through a webserver hosted by the embedded system.

The Hardware

Our original design was intended for the Intel Galileo. However, Galileo’s input/output rate was too low to even offer adequate keypad input (226 Hz)[2]. We completed our prototype using the Arduino UNO board, a simpler and more economic board that is software-compatible with the Galileo.

Component list:
1. Arduino UNO (16MHz ATMega328P, 32KB user memory)
2. Ethernet shield (enables networking with the arduino)
3. MicroSD card (for storage of encrypted attendance list)
4. Numeric keypad
5. 2 x 16 LCD
6. Keypad/LCD shield (our own circuit)
7. Power supply for the arduino

Future Work

- Optimize code and program size
- Create other “stripped down” or “bare bone” versions of the circuit
- Include more functionality (NFC, magnetic card reader)
- Port the project to other non-arduino boards (Raspberry Pi, Beaglebone Black, etc…)
- Make our source code and circuit design available publicly

References


Acknowledgments

Presentation of this work has been supported in part by NSF Grant CNS-1042341.