Secure BLE-Based Authentication using Sensor Augmentation

Austin Freel, Andrew Titus

Massachusetts Institute of Technology
[afreel, atitus]@mit.edu

I. Introduction

Many multi-factor authentication schemes using external hardware exist, but few utilize the existing Bluetooth Low Energy (BLE) and sensor hardware on people's mobile devices. This lack of ubiquity can lead to significant "friction" in users - that is, users experience a level of frustration every time they need to use a separate device that can often lead to users opting for less secure but easier to use methods. Fortunately, most mobile devices manufactured in the past few years contain hardware that enables BLE and various sensors to be utilized for wireless communications and data collection. This leads us to propose that a "low-friction" system can be built for existing mobile hardware that guarantees the same level of security for multi-factor authentication.

II. Methodology

Many Bluetooth-enabled peripherals (including mobile devices) have the ability to support encrypted communication. We will build our system on iOS, as the operating system requires that all remote devices connected to it support encrypted pairing connections (see here). We will start out by developing a pair of iOS 9 applications, one of which will function as a normal user and one of which will function as the BLE peripheral to which to send encrypted credentials. This will enable us to test the pairing and authentication processes without using external hardware for peripherals during testing. Then, we plan to replace this BLE peripheral iPhone with an external hardware device, such as an Arduino-based device or a custom-built Bluetooth peripheral (perhaps a 6.115 project by Andrew).

The setup of the pairing connection will be done by standard Apple guidelines for connecting Bluetooth devices. Authentication methods, however, are much more flexible and wide-ranging. We will initially start with a simple one-factor authentication system, such as a username/password combination or certificate-based system. Then, we will move into multi-factor authentication schemes which are augmented by internal sensor data, such as gestures. We also hope to make the system more robust to spurious authentications by using sensor data as well. For example, in the use case of the BLE peripheral being a door lock, this would prevent users from unlocking a door if they are simply walking by it.

III. Plan and Schedule

Assuming a due date of May 9th, we will have exactly two months to complete this project. We present here a list of deliverables, with our current plan and due date estimates:

- *Simple communication between user and peripheral iOS applications*: We will begin design work and begin constructing the basic wireframes of the test applications by **March 18th**, the start of Spring Break. We will then build the applications and establish this simple communication by **April 1st**.
- *One-factor authentication between user and peripheral iOS applications*: This will likely not need much additional work to the simple communication applications, so we have the goal of establishing a one-factor authentication scheme (likely with

simple dummy passwords) by **April 8th**.

- *Multi-factor authentication between user and peripheral iOS applications*: This is a somewhat heftier goal that will require a great deal of research, design decisions regarding what factors to use, and testing, so we are setting the due date for this to be **April 29th**.
- *Optimization for spurious authentications*: This is a somewhat unbounded goal, so we will aim to have some degree of completion on this by the end of the project on **May 9th**.

As the 6.115 Final Project Proposal is due April 5th, Andrew will have decided by this point what type of project he will be pursuing for this class. If it is possible to build a BLE peripheral for this project, we will focus concurrently on integrating this hardware using the same steps (simple communication, one-factor, multi-factor, optimization). Otherwise, we will begin similar work and research on an Arduino-based alternative. Our main deliverable is a proof-of-concept, so if the external hardware is not working as well as we hope, we will focus on making the phone-phone BLE multi-factor authentication system work.

IV. Resources and Requirements

In order to accomplish this work, we will need xCode 7 and at least two iOS 9+ iPhones, which we currently own. Additionally, we will need at least one external peripheral as mentioned, which will consist of either Arduino-based devices similar to the Anthill or a custom-built BLE device that Andrew will construct in 6.115 (Microprocessor Project Laboratory). We may also need laboratory equipment or specialized computers to be able to program these external peripherals. If Arduino-based devices are chosen, this should likely not pose much of an issue, but if the device is custom-built, Andrew will seek the proper permissions from the 6.115 laboratory staff to build such a device in accordance with our project.