CPEN 442 Project Proposal

Ben Henaghan, Student Number: 96671466
Scott Wang, Student Number: 72573322
Austine Yapp Student Number: 86705340
Mike Yue, Student Number: 24583156

***Abstract*—This project addressed the issue of keypad locks being highly vulnerable to a variety of simple attacks by implementing an internet connected keypad which utilised one-time passcodes. This retained the usefulness of a keypad (as opposed to a key lock) while greatly increasing security.**

I. Introduction

Keypad door locks are increasingly used in society today, with many more commercially-built homes and industrial buildings or factories switching from traditional key locks to keyless keypad door locks. The advent of such locks provides users with the much-desired convenience of keyless entry/exit. However, these systems are a relatively vulnerable security solution for preventing unauthourized access to a premises.

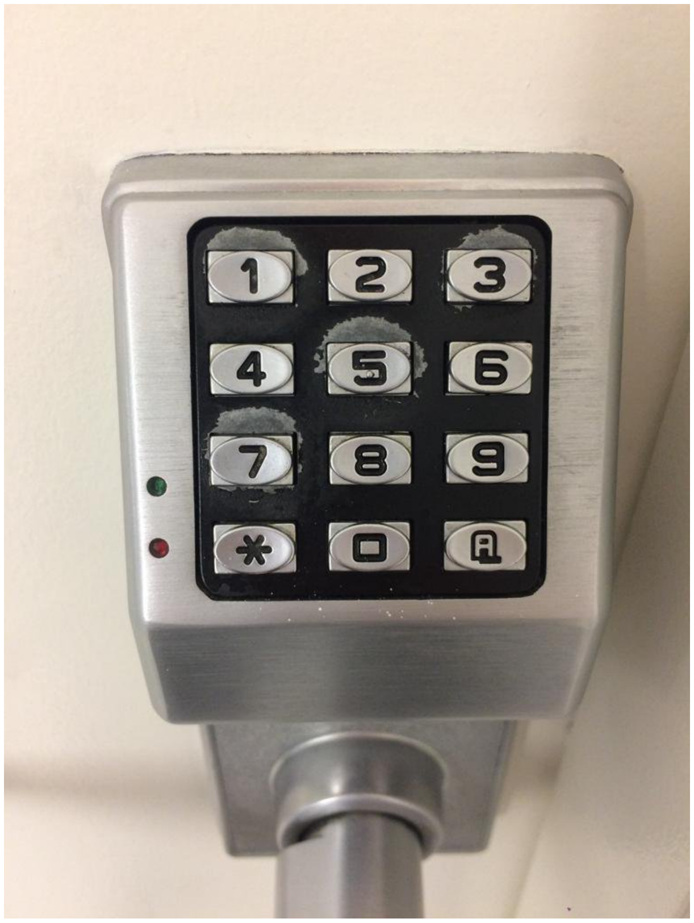

Fig. 1. A Keypad lock where button wear gives away the digits used in the code.

Upon further inspection, the vulnerabilities of a keypad door lock become increasingly apparent. One such vulnerability inherent to the design of the keypad door lock is the susceptibility of visual wear after prolonged use. Mechanical wear through direct and erosive contact with the keypad, as well as chemical wear through reactions with natural skin-oils, may cause keypads to quickly show signs of fatigue. In Figure 1, the keypad digits 1, 3, 5 and 7 are clearly worn out compared to the others. An adversary would thus only require a total of $4! = 24$ permutations of access codes in order to crack the users' access code.

Furthermore, the repetitive use of the static access codes may be vulnerable to shoulder surfing/pinhole camera recording attacks from adversaries. An adversary simply needs to identify the users' static access code in order to have full access to the compounds. One relatively easy method would be to simply install a pinhole camera in an inconspicuous position over the keypad door lock. With any unobstructed video capture of a users' entry, adversaries may quickly figure out the access code for that lock.

This project thus strives to introduce additional layers of security through the use of dynamically-generated One Time Access codes. Given the keypad door locks' extensive use in both commercial and industrial markets, the value of the assets that are secured by it is practically priceless — many users rely on one as their only access control for a given premises. Such a security analysis of the system is therefore of paramount importance, seeing as its vulnerabilities pose a threat to the incalculable worth of the users' assets.

II. Current Solutions

For security purposes, most keypad door locks in the market allow users to reset their door code. Periodic changing of the access code is encouraged to prevent visual wear as well as to prevent the code from being compromised for an extended period of time. However, there is typically no requirement for changing the access code regularly, which increases the vulnerability of the keypad door locks significantly. Therefore, the responsibility is on users to manually reset their door code on a regular basis, which is a mundane and tiresome task for most people. This means that many users will neglect to reset their code at all throughout the lifetime of the lock (years), ultimately causing an unsecure lock.

III. Our Implementation

The project focuses on the security improvement of a specific subset of digital locks: Smart locks. These are locks which can connect to remote servers over the internet. The design is split into 3 parts, a mobile application, a server, and the Smart lock itself.

Each user is assigned a static ID number, which is associated with their account. Every time access is required, the user simply requests a new temporary PIN number via the mobile application. The server registers the request, and randomly generates a new temporary PIN and associates it to the user's account. The user enters their static ID number followed by the temporary PIN, and the Smart lock will transmit the data securely to the server for verification. Upon successful verification, the lock opens for the user, and the server removes the temporary PIN from the user account.

IV. Comparison to Competing Solutions

By replacing the entry code every time it has been successfully used to enter the premises, pinhole camera/shoulder surfing recording attacks become next to useless, as the code they record will become useless the instant it's used. Furthermore, as a random code is being generated for each access, the wear and tear of all digits on the keypad will be effectively even. The automated generation of new entry codes also means users no longer have to worry about resetting passcodes themselves.

V. Plan for Project

A. Hardware Development

Our project required a prototype hardware system on which to test the software. This meant that it was imperative to have a functioning hardware component as early in the development cycle as possible. The project team agreed to finalise a design for the prototype hardware by October 25th in order to give sufficient lead time to order components and build the hardware by November 8th.

This finalisation of the prototype hardware did not signal the end of hardware development for the project, we decided to keep developing the physical lock system to make it a more compelling commercial product.

B. Software Development

Software development started with the establishment of a strict protocol between the client and server for authentication and sending the user their one-time key. Concurrently, the broad OOP software design for the server and client android application was defined.

After the software design was completed (which was planned to be finalized by October 21st) full software development was started. We chose to adopt an agile-like methodology where development was broken up into 1 week 'sprints' with key tasks for each team member to complete.

C. Development and Testing Methodology

We chose to make use of 'Test Driven Design' as a core part of our software development process. Unit testing was used for most back-end software and integration/system tests were created for critical or complex components. Towards the later stages of our project we employed user acceptance testing to help optimize the user experience.

Hardware testing comprised mostly of testing the functionality of our prototype — security testing of the prototype hardware was avoided as we felt it was less relevant to our prototype and would incur extra materials cost if any destructive testing was undertaken.

D. Documentation and Artifacts

Submission of the final project report and video constrained all other parts of the project plan. Among other sections, the introduction needed to be finalized by November 13th. This meant that it was crucial that the bulk of development was finished by this point, and thus we set the target of a **working minimum viable product by November 13th** meaning that we would have a few weeks to produce our video clip and fix any issues not found during our testing stage.