NetBolt: A Remote Accessible Deadbolt Lock

Ryan Moe
University of Washington, Tacoma
Tacoma, WA, US
ryanm88@uw.edu

Nazim Zerrouki
University of Washington, Tacoma
Tacoma, WA, US
email address or ORCID

***Abstract*—With the integration of Internet of Things has led to more sophisticated pieces of technology. Devices can interact withe the internet to extract information from the web or relay information stored in the Cloud. To this end, we integrated a smart lock solution known as netbolt that will be used to reinforced mechanical locks to provide convenience, and ensure secure and authenticated entries. The smart lock was implemented using a Raspberry Pi. The Raspberry Pi will serve as the device to connect to a locally hosted webserver and relay user feedback and will invoke the use of an RFID reader so that users will be granted entry upon authentication of an RFID tag. Our smart lock solution connects to a MySQL database stored within the Azure Cloud service to store authenticated users with their respective RFID tags. It will also relay information using the Azure Cloud to convey critical information such as access events, access time, and previous user who accessed the lock. The locally hosted webserver facilitates creating user credentials and generating the RFID tag necessary for the locking mechanism. In addition to the Azure Cloud, netBolt has also been configured to send notifications upon unathorized access. This will be among the few smart lock solutions that will be implemented using a Raspberry Pi in an attempt to improve upon existing smart lock solutions. netBolt will provide a secure, convenient, and transparent system that may be feasible for AirBnB and Vrbo services with future enhancements.**

***Index Terms*—RFID, Internet of Things, transparency**

I. Introduction

Most US homes utilize some type of lock to prevent intruders from gaining entrance to the home, the most popular being mechanical pin tumbler locks. These and other mechanical locks all share a fundamental issue: they can be picked, raked, or bumped by criminals using only basic, inexpensive tools and little practice. Moreover, these types of attacks leave no evidence that a break-in even occurred. We aim to give property owners the option to replace or reinforce their mechanical lock by offering a smart deadbolt lock that can be opened remotely through a smartphone, via the network, or in person, using Radio Frequency Identification i.e an RFID tag. With remote access, we want to ensure that the status of our smartlock is transparent at all times. We call our system the netBolt, and believe it will make home security more reliable and convenient.

The netBolt will not be the first smart door lock. Other smart locks are combined with a mechanical lock or provide a keypad to offer multiple access options. By integrating remote access remote access and RFID, we aim to maximize convenience while minimizing the surface area. Imagine a scenario in which you had forgotten whether the door was locked as you head to work. Without the ability to access the lock remotely, your only option would be to return home to verify it. Or perhaps you would have multiple members in the household and it's difficult to properly manage whether the door was locked or not once you have left. By having access through a smartphone, you would receive notifications regarding the status of the lock and whoever is present at the home. In the event of an intruder, smart locks with the support of a camera would enable you to verify who is at the door and lock or unlock the door as needed.

Many solutions use other pieces of technology to create their smart locks but there's been a scarcity of solutions that have integrated the use of a Raspberry Pi. Therefore, netBolt will be one of the solutions that attempt to improve on existing smart lock solutions. The device will provide multiple features including login functionality, configurable notifications for lock/unlock and door open/close events, access sharing with guests, and potential remote access via a camera for improved system security. As a novelty for Raspberry Pi solutions, our system in particular will provide the ability to both share and revoke temporary access for outside guests which would be beneficial for both AirBnb and Vrbo services.

II. Related Works

There have been a few smart locks that were created using a Raspberry Pi. One of the most popular methods for user authentication and data storage is the invocation of RFID tags and SQL Databases to store user credentials and event logs [1, 2, 3, 4]. A smart lock was created using a raspberry pi among other hardware components and circuitry to fulfill multiple sophisticated components [1]. This smart lock was capable of relaying critical information pertaining to access logs using an LCD screen and invoked fingerprint scanners and a motion detector to authenticate user entry and detect motion outside of the door. This smart lock also used a MySQL Database to store user credentials and system information. Our smart lock adopted these principles to store and relay necessary information for the user. Another smart lock solution implemented a 6 digit-PIN in addition to the RFID tags to provide an alternative for user authentication [2]. This method used a MySQL Database that resided on a local Apache webserver to store authorized users and access logs conveying events of successful and unsuccessful unlock attempts.

Ayyub created a smartlock through an Android or IOS application for user registration which was hosted on a Flask server

[3]. Within the smart lock presented by [4], it was deployed on a webserver facilitated by Node-Red. The smart lock used a MySQL database schema which contained multiple tables to store all of the access logs, events, users, and RFID tags. This was hosted on a MQTT server using a raspberry pi. The smart lock presented in [5], however, did not use RFID tags. Facial recognition and machine learning was used to execute user authentication. In the end, majority of the solutions we researched used RFID tags so that was our authentication scheme.

III. System Architecture

Within this section, we will discuss the design procedure of our smart lock as well as the hardware setup.

A. Design

The netBolt will provide an intuitive interface for users when interacting with the lock. Users will first interact with a local node-red dashboard hosted by the raspberry pi to register user credentials, store their respective RFID tags, and invoke the locking mechanisms of netBolt. The user information will be stored on a MySQL Database hosted on the Azure Cloud Server. This database will be prodded to authenticate users for unlocking the door through the RFID sensor. An integral aspect of this is enabling a connection between our locally executed program and the database within the Cloud. This connection will be invoked using the node-red-node-mysql native library so that the database will be prodded through an SQL query whenever a user needs to be authenticated when using netBolt. When the authentication is complete, the raspberry pi will invoke a response that indicates whether the door was unlocked or remains unlocked. If the user was rejected, then the red LED on grovepi will turn on using a digital output node to reflect the rejection. Otherwise, the green LED on grovepi turn on and the buzzer will be invoked using an analog command to provide clear user feedback. The lock status as well as the process of reading the RFID tag will be reflected on the dashboard.

Additionally, other proposed methods stored events pertaining to the type of interaction, recent access time, and recent user who interacted with the lock. As such, the event logs that were used to provide a historical chain of events of every interaction that occurred within the door was stored within an InfluxDB database. InfluxDB has an inherent mechanism that stores information in respect to the time the request was made. As such, each event will be logged in respect to when the event was initiated. This events will also be reflected on our dashboard hosted on the Azure Cloud to provide transparency of every interaction that occurred. The user will be notified of any authorized attempts and any attempted intrusions as well as the current lock status of the door so users can react appropriately.

To this end, when the authentication response is complete, a notification will be sent to the InfluxDB database. The notification will reflect the time of the event, the previous authorized user, and whether the event was authorized or intrusive. To ensure security, the RFID tags that are read are regularly updated at a specific time. These updates will be reflected in the MySQL database.

B. Methodology

This section will discuss the sequential process of netbolt. The data flow is as follows:

1) Users will interact with the node-red dashboard to register user credentials and retrieve an RFID tag for authentication.
2) Upon opening the door, netBolt will demand the user's RFID tag for authentication.
3) The authentication will be initiated by connecting to the MySQL database.
4) If the RFID tag was revoked, users will be able to sign in using their username and password as an alternative form of authentication.
5) Once complete, the record of the events will be stored within our event logs that will be stored in the InfluxDB database.
6) The events will be reflected on the Azure Cloud dashboard.
7) The raspberry pi will use grovepi digital sensors using the green and red LEDs to depict the lock status of the door. The buzzer will be triggered when the door is successfully unlocked.
8) An email notification will be invoked as a rule which will notify the user of any unathorized attempts.

C. Hardware Set-Up

The netBolt is built using a Raspberry Pi as a controller with GrovePi+ hat. Connected to the controller are a 125 kHz RFID scanner, servo motor, magnetic door switch, piezo buzzer, and LED lights.

- The RFID scanner is attached via the Raspberry Pi serial port built into the hat. It is made up of two parts: the scanning controller proper, and a rectangular antenna.
- The servo is attached to a linear actuator that serves as the bolt assembly, and receives signals from the Pi via GPIO/BCM pin 18 (physical pin 12), with 5V power and ground on physical pins 2 and 9 respectively.
- The magnetic door switch is a sensor that reports "true" when affected by a magnetic field, and "false" when it is not. Paired with a magnet attached to a door frame, it can be used to detect whether the door is open or closed.
- The Piezo buzzer, Red LED, and Green LED are used to signal to the user when a change has taken place, such as when a tag has been accepted, and the door has unlocked.

IV. Discussion

Within this section, we discuss the challenges encountered and any changes made while implementing netBolt. Initially, we considered using Grafana to illustrate the event history for the access logs. One of the challenges encountered was successfully invoking the connection between the Grafana container hosted on the raspberry pi. Multiple attempts were

conducted to properly establish the connection. We successfully managed to connect to the Grafana container by configuring digital certificates that were provided by Microsoft. The Grafana dashboard was not implemented in the final prototype because it was redundant. The implementation was already covered by the Azure IoT Dashboard which provided the functionality we needed in regards to conveying event information. Azure also provided a very intuitive method in integrating email notifications when an attempted intrusion occurred. This was a necessary use-case in providing an extra layer of security and transparency for the user. Instead, we chose to use node-red to provide a GUI to enter user credentials and generate the RFID tag. The lock and unlock mechanism was also a challenge to integrate so this was also added to the node-red dashboard to provide the necessary functionality. Another challenge encountered was configured using the RFID reader and testing it to ensure that it worked proprly. This was not covered within the scope of the class so extra research and testing was conducted to ensure that it was working properly.

V. Conclusion

The proposed solution, netBolt, achieves security, transparency and convenience for the users. The raspberry pi was used to host the node-red server which integrates the front-end and the front-end to register the user's credentials and allow a conventional login system as an alternative. It was also used to provide user feedback through the LEDs and the buzzer to illustrate the lock mechanism and user revocation (if applicable) in the real-world. As a part of Internet of Things, we integrated a cloud service using Microsoft Azure which houses the MySQL Database to store user credentials and their respective IoT tags. It was also used to illustrate the access logs in real-time so the user is informed of every authorized and unauthorized use of netBolt.

As part of our contributions, Nazim Zerrouki worked on the report, instantiating and hosting the databases, and the cloud dashboard. Ryan Moe worked on the hardware component with designing the smart lock, configuring the RFID reader, and the video. We worked collaboratively on creating our flow on node-red.

VI. Future Work

netBolt only serves as a finalized prototype for a smart lock system that integrates RFID authentication through the Raspberry Pi. As part of our future works, we would enhance netBolt by adding additional functionality such as facial recognition or a fingerprint scanner, a camera that acts an additional component to the security system to relay information outside of the door, and displaying the visual feed onto the dashboard in real-time. Instead of a conventional login system as an alternative for user authentication, a 6-digit PIN code would be preferable but it would require a more sophisticated design which is outside of our expertise. The additional functionality would still be implemented on node-red which would require further additions to our flow. This would be the future plan going forward.

References

[1] I. (2019, June 21). SafetyLock: a Smart Lock Made With Raspberry Pi (Fingerprint and RFID). Instructables. https://www.instructables.com/SafetyLock-a-Smartlock-Made-With-Raspberry-Pi/

[2] Hackster.io. (2018, September 25). The ULTIMATE Raspberry Pi Smart Home Door Lock. https://www.hackster.io/paulfp/the-ultimate-raspberry-pi-smart-home-door-lock-3c55a0

[3] Ayyub, I. (2018, November 5). Smart Lock. Raspberry PI Projects. https://projects-raspberry.com/smart-lock/

[4] RFID Door Lock User & Access Management (flow) - Node-RED. (n.d.). https://flows.nodered.org/flow/b9860e96b4df2c1a94a864b6343b067b/in/9RgOMM3J9o

[5] Face Recognition Door Lock System using Raspberry Pi. (n.d.). https://iotdesignpro.com/projects/face-recognition-door-lock-system-using-raspberry-pi