SMART LOCKS: EXPLORING SECURITY BREACHES AND ACCESS EXTENSIONS

By

PALLE, SAIPRASANNA

Bachelor of Technology in Computer Science

Jawaharlal Nehru Technological University

Hyderabad, Telangana

2012

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2017

SMART LOCKS: EXPLORING SECURITY BREACHES AND ACCESS EXTENSIONS

Thesis Approved:

Dr. Eric David Chan-Tin

Thesis Adviser

Dr. David Cline

Dr. Yanmin Gong

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Eric David Chan-Tin for the continuous support of my research, for his valuable suggestions and patience. His guidance and feedback helped me in carrying out the research smoothly and in writing of this thesis. He has also been a mentor in the whole process.

I would like to thank the thesis committee: Dr. David Cline and Dr. Yanmin Gong, for their suggestions and involvement in the research.

I thank my fellow mates: Ashwini Jinka and Rohit Nutalapati, for their stimulating discussions on the topic and helping me carrying out the experiments. I would like to thank my seniors for sharing their experiences of their thesis, which helped me in approach to my thesis.

Last but not the least, I would like to thank my family: my parents, sisters and brother-in-laws for their unfailing moral and spiritual support during my thesis.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: PALLE SAIPRASANNA

Date of Degree: JULY, 2017

Title of Study: SMART LOCKS: EXPLORING SECURITY BREACHES AND ACCESS EXTENSIONS

Major Field: COMPUTER SCIENCE

Abstract:

The Internet of Things (IoT) has rapidly become one of the most popular devices across businesses and technology. Referred to as the next industrial revolution, it has transformed the way devices interact with each other, with home automation being the one of the popular fields of IoT. Though IoT can make the devices smarter, it also has a potential to expose these same devices to an attacker, to exploit their vulnerabilities, thus raising a concern for security.

This paper deals with the security aspects of one of the popular subsets of home automation systems: smart locks. Smart locks replace the traditional door locks; they can be electronically controlled by mobile devices or the lock manufacturer's remote servers. The current state of art, design, implementation, vulnerabilities and the attacks that can be exploited on these devices are discussed. A solution that addresses the design and architecture of IoT smart lock in providing a better security mechanism is proposed for achieving smarter and secure smart homes. It is shown through experimental analysis that the proposed smart lock system is secure against unauthorized access, replay, and relay attacks.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

The widespread adoption of Internet of Things (IoT) is significantly increasing with the emerging technology trends. It has been estimated at almost 30 billion devices will be connected to the IoT by 2020 [18]. Smart home automation is one of the most popular areas of IoT wherein different electronic devices connected on a home network can be operated from a smart phone and can internally communicate with other IoT devices. Smart locks are smart home automation devices wherein, the deadbolts are replaced by a lock operating on Bluetooth Low Energy (BLE) or other short-range network signals, which in turn can communicate with the user’s phone wirelessly. Smart locks do not just replace a door bolt, but can also track who is entering and leaving the home, and can also lock and unlock doors automatically without manual intervention by sensing when a user’s smartphone is nearby.

Various installations of smart locks are already in use but it has been shown recently at DEFCON hacking conference in Las Vegas in 2015, that roughly 75% of Bluetooth-powered Low Energy smart locks are susceptible to hacks [19]. The design of the smart lock and sometimes the implementation mechanism behind a smart lock can act as a security breach. For example, a smart lock transmitting the digital keys associated with it in plain text can make them vulnerable to wireless sniffers. An attacker can also replay the authorization message captured from the legitimate user and trick the smart lock to give the attacker access. These hacks can be exploited further to trigger a false alarm, breaking into a house, and so on. This calls for a secure architecture design and implementation for secure smart locks for smart homes.

Smart locks use Bluetooth Low Energy (BLE) technology and/or Wi-Fi technology, thus allowing remote locking/unlocking and access key extension to guests for a specific period of time. This paper discusses the design of existing smart locks, the attacks that can be carried out by exploiting their vulnerabilities, the solutions that have been already proposed for counteracting them and the feasibility of these attacks.

The contribution of this thesis is the implementation of a smart lock system that allows for remote automatic locking and unlocking using a smartphone and granting access to guests for specific period of time. The system is also shown to prevent unauthorized access, replay attacks, and relay attacks.

Section 2 gives a background overview of the architecture of the current existing smart locks and the attacks that are possible on them. Section 3 proposes a design solution for achieving secure smart lock and the implementation details of the proposed secure smart lock system along with security measures for the identified attacks on smart locks. The analysis of the attacks is given in Section 4 and choosing a time threshold for defeating these attacks is discussed. Section 5 concludes by discussing the trade-offs in achieving a secure system

CHAPTER II

BACKGROUND

Smart locks have evolved from “touch based unlocking” to “automatic unlocking” devices [1,5,14,15]. Some of the smart locks today are also equipped with Wi-Fi modems, which have the ability to connect to a home network to interact with manufacturer’s servers and log access records, push access tokens, etc.

Current smart locks have three components associated with them: the smart lock itself, which is equipped with Bluetooth Low Energy (BLE)/Wi-Fi capability, the user’s mobile device, and the smart lock manufacturer’s servers. The smart lock can be controlled by the app installed on the user’s mobile device. For initial access, an account is created on the manufacturer’s servers and then the token or digital key is received from the server with the lock on a wireless channel like BLE.

2.1 ARCHITECTURE

Smart locks follow one of two network designs. In the first architecture, shown in Figure 1.1, Smart locks themselves do not have any Wi-Fi capability. Instead they rely on users’ mobile phones to interact with the manufacturer's servers. The communication between the user's mobile phone and the lock occurs on BLE, whereas the lock pushes any state updates or logs to the server through the Internet capability of the user's mobile phone. Hence, the user's mobile phone acts as the "gateway" here and the server as the cloud and this architecture is called Device-Gateway-Cloud (DGC) model [1,5,14].

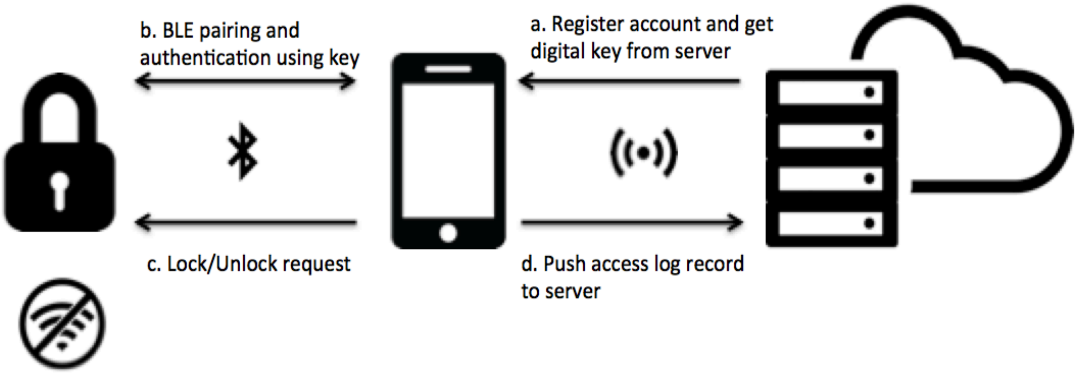
b. BLE pairing and authentication using key
a. Register account and get digital key from server
c. Lock/Unlock request
d. Push access log record to server

Figure 2.1: Device-Gateway-Cloud architecture

In the second architecture, the smart lock is embedded with a Wi-Fi modem. Hence, the lock can directly interact with the manufacturer's servers through the Internet by connecting to the home network. The communication between mobile device and the lock occurs through either BLE or Wi-Fi. But any state updates or access logs recording or exchange of digital keys, is carried out by the lock through its Wi-Fi capability. This architecture is thus called Direct-Connectivity [15] referring to the direct connectivity between the lock and the manufacturer's servers.

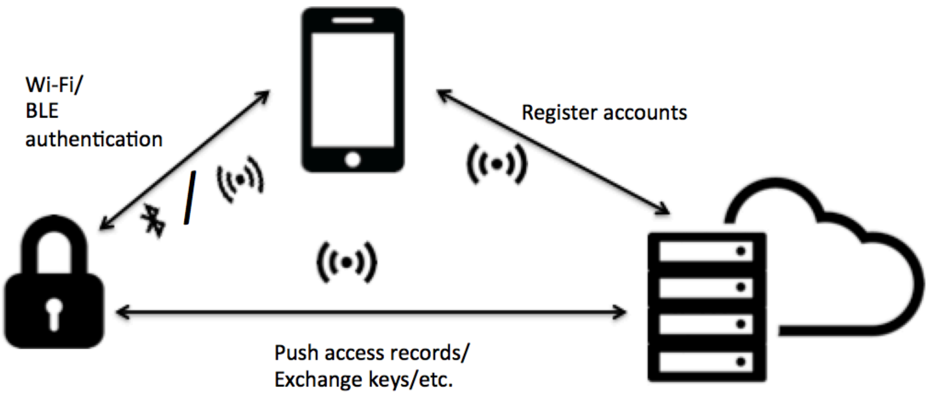

Figure 2.2: Direct Connectivity architecture

In both architectures, the authentication is done by the exchange of digital key between the lock and the user mobile app. Keys are further classified into owner, resident, guest, etc., levels. The "owner" status is given to the first device that pairs with the lock and this status is maintained throughout unless the lock is reset. The "owner" can enter and leave the home any time and also has administrative capabilities like granting or revoking access, etc., with respect to the features provided by the lock. A resident also can enter and leave the home at any time but doesn't have the administrative capabilities. A guest has temporary access to the home.

2.1.2 Wireless technologies for Smart lock – Device communication

Different modes of wireless communication, which can be used for communication between smart lock and the user's device, are explained in the following sections.

2.1.2.1 Bluetooth Low Energy

One of the cheapest, easiest and the most common modes of the communication used between a smart lock and user's device is Bluetooth Low Energy (BLE).

There are two key players in a Bluetooth low energy communication: the central and the peripheral. Each of the players: peripheral or central, has its own set of functionalities to perform. A peripheral typically has data associated with it, which is of interest to the central and is used by the central to accomplish some task. For example: an automatic light switching system can have a sensor equipped with BLE to provide the light intensity of a room to an mobile app which then controls the home electric lights or take the necessary action. The sensor acts as the peripheral here and the mobile app as the central.

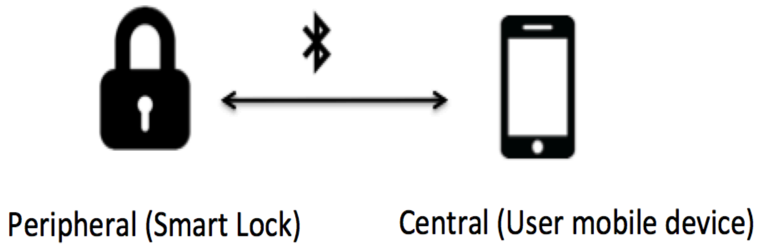

Figure 2.3: Peripheral and Central roles in a smart lock

Figure 2.3 shows how a device can act as a central using the BLE capabilities.

Peripherals make their presence known by advertising their manufacturer data and the data they have, over the BLE communication. Centrals, which keep scanning for a particular peripheral or particular data, try to connect to a peripheral when they find the peripheral or data of their interest. If the connection is successful, the central interacts with the peripheral data and the peripheral responds in an appropriate way to the requests it receives.

2.1.2.2 Other Wireless technologies

The smart lock and the device can also communicate on NFC (Near Field Communication) and Wi-Fi wireless technologies. NFC is designed for very short-range communication, which might range from a few cm to no more to 10 cm. With NFC, data can be transmitted between a tag and a mobile device. The tag can act as a smart lock and the mobile device as the user's device.

The communication between lock and the user's device can also go over Wi-Fi, though it is used less because BLE dominates over the other wireless technologies.

2.2 CAPABILITIES

2.2.1 Unlocking Mechanism

Different locks implement unlocking mechanisms in different ways. Some locks need an explicit pressing of a button in the mobile app [15]. Some provide an automatic unlocking feature by removing the users' need to interact with the mobile device explicitly. The lock unlocks if the mobile device is within the wireless range of the lock and locks itself once the device goes out of wireless range [1,5]. Some other locks have a built-in touch sensor and the user has to touch it to unlock the door [14].

2.2.2 Recording Access Logs

Recording access logs will enable the owner to track who is coming and leaving out of the house. In the case of DGC architecture, the logs are pushed to a server by the lock through the users' mobile app whereas in direct connectivity the lock pushes the logs by itself. A record is logged at the lock with the user’s name along with time stamp whenever an entry is made. Also, any change in access/digital keys in case of granting new keys or revoking the existing ones is also recorded.

2.2.3 Exchanging digital keys

Apart from the owner, guest or resident also can access the lock by downloading the key from the server and passing it to the lock over the BLE communication in case of DGC architecture and BLE/Wi-Fi in case of direct connectivity. An owner can remotely grant these access tokens or digital keys to any other user by pushing these digital keys to the manufacturer’s servers. Thus a smart lock can provide remote granting and revoking of access keys.

2.3 SECURITY THREATS

The architecture design or the implementation mechanisms chosen for designing a smart lock plays a critical role in determining how secure a smart lock is. In an attempt to provide more ease of usability over traditional unlocking mechanisms, the current smart lock systems can be exploited in various ways to break into a home illicitly. The attacks can vary from manipulating the log access records to impersonating the legitimate users and smart locks to gain unauthorized access into the home, which is explained below.

2.3.1 Relay Attacks

Relay attack is a type of man-in-the-middle attacks where an attacker gains illegal access by replaying captured signals between authenticated entities. An attacker is the one who initiates the communication between the parties in case of a relay attack. He then merely relays the packets between the two legitimate parties without manipulating the packets.

Numerous prior papers have demonstrated the practicality of relay attacks against analogous auto-unlock protocols in cars [6,9,10,11] and Bluetooth authentication protocols [12]. In smart locks scenario, the relay attack is carried out between the lock and the legitimate user's device, when the user's device is not in proximity of the lock. The below figure explains the Relay attack scenario in case of Smart Locks.

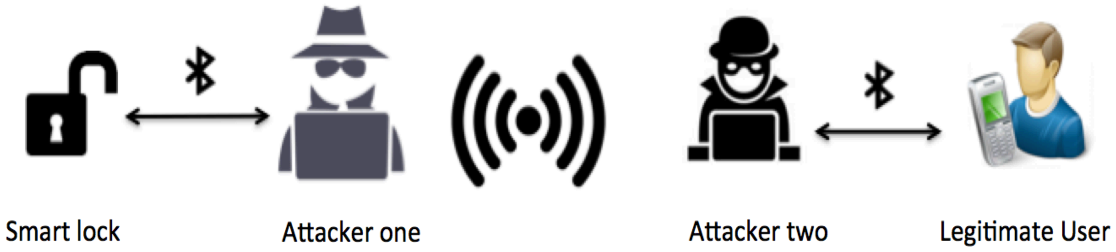

Figure 2.4 Relay attack in Smart lock systems

There will be two attackers involved; one to capture signals from the lock and the other from the user's mobile device. The signals are transmitted between them and the captured responses are relayed back to the communicating parties, impersonating both the smart lock and the legit user thus gaining access to the home. Here both the parties would not have any idea about the attacker in between with whom they communicate.

2.3.1.1 Existing solutions for Relay Attacks

Securely verifying that the user is not too far away from the door would stop many attacks. This type of approach is classified as location-limiting defense and is used to propose techniques for preventing relay attacks. The two possible instantiations of this approach are discussed below.

NFC: One approach to verify that the lock and user are within a small distance from each other is to use Near Field communication (NFC) as a medium of communication in place of BLE or Wi-Fi. NFC is designed for very short-range communication, in the range of few cm to no more than 10 cm [8].

This approach mitigates the threats posed by an attacker in case of relay attacks, such as unlocking the door for a wrong user. However, studies have shown that the NFC is vulnerable to relay attacks [10, 11]. There are no additional checks imposed to limit the distance between the communicating parties other than the field strength or the short-range signal strength generated by these NFC devices. Also, prior works have shown that attackers can easily boost or magnify the signal between the two devices and can still carry out a relay attack [10,11].

Distance-Bounding Protocols*:* Another approach used to securely verify that a user is near the lock is the distance-bounding protocol. In these protocols, an upper bound on the distance between the two communicating parties is established. The communicating parties are engaged in a series of challenge-response and an upper bound distance is calculated taking the round-trip time into account. Using a specialized hardware and custom protocols, some of the prior works have demonstrated the ability to verify distance to within 12cm [16].

A lock could use these distance-bounding protocols for secure auto-unlocking. When the user's device comes within the BLE range of the lock, it authenticates as normal and then runs the distance bounding protocol repeatedly. The lock unlocks automatically only if it is confirmed that the user is in fact within a short distance. The distance-bounding computation employed in these protocols makes it secure against relay attacks and will prevent an attacker from spoofing the location of an authorized user.

The hardware limitations of the mobile devices makes it harder to support distance-bounding calculations employed in these protocols. Also, as BLE currently does not support distance bounding, it is not feasible to implement this mechanism in a smart lock.

Touch based intent communication was also proposed for mitigating relay attacks as an alternative to inferred proximity, using touch as a signal for conveying user's intent [13]. The implementation works on the idea of transferring "intent-signal" between the lock and the key over a data channel, using body-area networking (BAN) [17]. The user wears a wearable device such as ring, bracelet or a smart watch which can communicate securely with the lock on a wireless channel (e.g., BLE or Wi-Fi).

Touch-based intent protocols have been show to resist the relay attacks against a smart lock. But in general, they also have a few limitations. Like distance-bounding protocols, they require the user of new hardware. Also, users need to touch the door or the lock with the hand they wear their wearable device on. This eliminates some of the features of the lock like: auto-unlocking. This again marks a trade off between usability and security, removing the auto-unlocking feature of the lock without the need of manual intervention.

2.3.2 Access Revocation and Log Evasion Attacks

Smart locks enable tracking whoever is entering and leaving the home and support access extension to other users. In case of a direct connectivity model, where the smart lock has a Wi-Fi capability, it can directly interact with the manufacturer's server to get the access list and can directly push the log records to the server. But in case of DGC model, the lock solely relies on the user's mobile to authenticate it and also to push the log records to the server. Access extension in this case by pushing the authenticated certificate by the server onto the user's mobile which is then provided to the lock. Revocation of the access works in the same way wherein the server pushes a revocation certificate onto the user's mobile. But if the user's device is switched to offline mode, there is no way a server can push this revocation certificate and the user's device still maintains access to the lock. In the same way, the user can stop from pushing the log records to server by switching it to offline mode. This type of situation where the lock doesn't know when to allow or block a guest user in case of DGC architecture paves way to access revocation and log evasion attacks.

2.3.2.1 Existing solutions to Access revocation and Log Evasion Attacks

Lock state's related attacks are targeted in DGC over Direct-Connectivity architecture as the Direct-Connectivity locks have an Internet connection to be able to take decisions on whom to allow into the home by contacting the servers directly. Hence, the user's mobile doesn't play any role in this architecture model for getting unauthorized access into the home.

Eventual consistency [13] can be used for updating security-critical state, such as access control lists, and for deciding when to allow access to users in the presence of server unavailability. It is based on the CAP Theorem for distributed systems which states that if network partitions can occur, it is impossible to provide full availability for the system's service, while simultaneously maintaining the latest, consistent state across all nodes in the system [4].

Eventual Consistency: The DGC architecture can support eventual consistency. The end-to-end communication is shared through a long-term symmetric key that is shared between the lock and the server during the lock installation. The access control list maintained at the server is signed and version-controlled. Each time a user interacts with the lock, the user's mobile app fetches a signed, version incremented access control list from the server. The lock updates its access control list too if the list provided by the user is signed and fresh. If it does not receive a valid updated list, it will not update its local access control list but will still provide access to the user.

It can be seen that this design only allows a thief or revoked attacker to maintain unauthorized access to the lock (once the attacker's access is revoked) as long as no legitimate user uses the lock. As soon as a legitimate user interacts with the lock, the updated access control list is provide to the lock which will reject the future communications from the attacker's device. Hence, most of the illegitimate communications can be stopped but it can be seen that the model still gives the attacker a very small window to gain unauthorized access to the home until the owner returns home and syncs the access list.

2.3.3 Replay Attacks

A replay attack is a man-in-the-middle attack where the communication between authentic parties is captured and is played back at a later time giving a false notion as if the communication is from an authentic party.

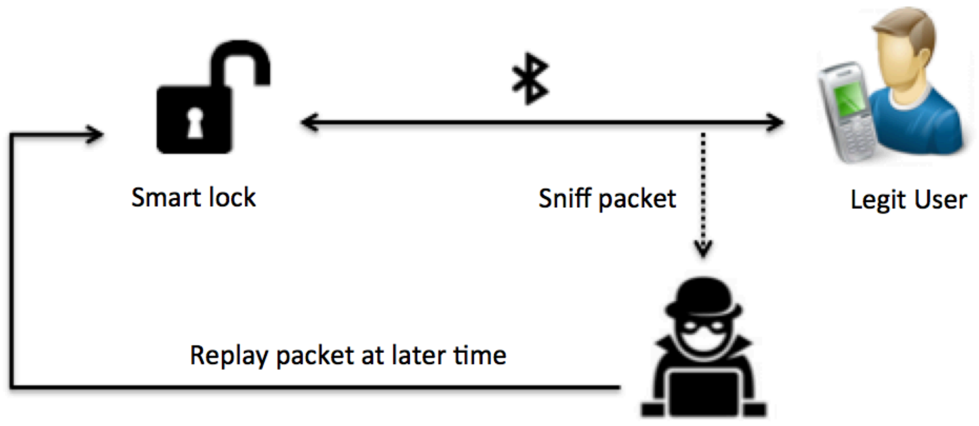

Figure 2.5: Replay attack in a smart lock system

In a smart lock system, it works by the attacker sniffing the BLE communication between the user and the lock and then replaying the same packets to the lock later. This requires the user to be in close proximity of the user's mobile device to be able to sniff the packets.

There are no studies yet on the replay attacks in smart lock systems. However, studies do indicate the possibility of replay attacks in IoT systems [20].

CHAPTER III

PROPOSED SOLUTION

The existing smart lock systems are either based on Device-Gateway-Cloud or Direct Connectivity architecture. Smart locks embedded with Wi-Fi modems in the case of a Direct Connectivity model can mitigate access revocation evasion attacks discussed in section 2.3.2 easily, but can also pave way to more threats as they are directly connected to internet. Any compromise made to the smart lock can potentially compromise the other devices too, connected to the same network.

In the DGC architecture, there is no direct way to access the lock because the user's mobile acts as a gateway in between and requires the attacker to first compromise the device before the lock is broken, thus adding a layer a security. Also, the economic costs and the power consumption for the direct connectivity model systems are relatively higher when compared to DGC models because of the need of Wi-Fi modem in the former models. Hence, targeting the security at an architecture level, the proposed solution for secure smart locks is constructed on top of DGC. The design and implementation of a basic smart lock system is introduced first, on top of which the mechanisms for mitigating relay and access log evasion attacks are constructed, which are discussed in detail in later sections

3.1 DESIGN

The communication between the smart lock and the user's device occurs on BLE. In below sections, the proposed design and implementation of a secure smart lock is discussed.

3.1.1 Designing a basic Smart lock using Core Bluetooth API

The concept of central and peripheral discussed in section 2.1 can be extended to mobile devices in designing a basic smart lock.

Two mobile devices are used. One acts as a peripheral or smart lock and the other as a user's mobile device. The lock advertises its data and services and the mobile device connects to it and authenticates itself. If the authentication is successful, the lock/unlock request is sent to the lock depending on the signal strength of the lock.

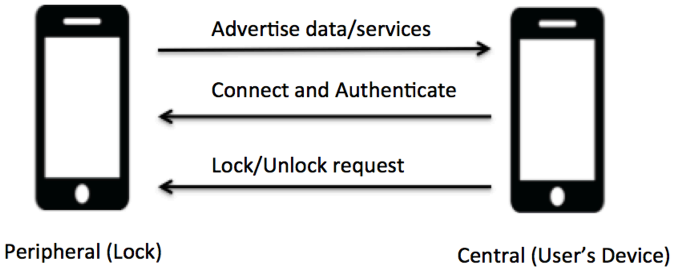

Figure 3.2 Designing a basic Smart lock using the Core Bluetooth API

Figure 3.2 gives a brief overview of the steps involved in a basic smart lock design. The lock is automatically unlocked if the user's device is within the proximity of the lock else it is locked.

3.1.3 Design of Relay attacks

Revisiting the working of relay attacks in smart locks again, they are carried out between the lock and the legitimate user's device, when the user's device is not in proximity of the lock. There will be two attackers involved, one to capture from and to relay signals to a lock and the other attacker to relay the received signal from the first attacker to the lock and then transmit back the captured signal from the lock to the attacker one.

Four iOS devices are needed to imitate the responsibilities of lock, Attacker one, Attacker two and the user's mobile device.

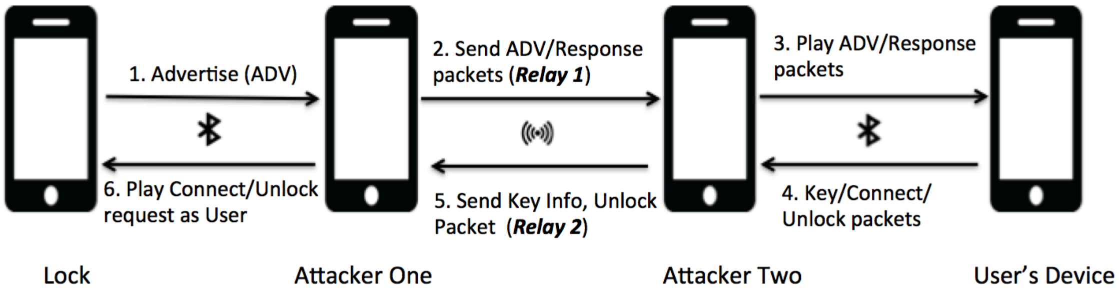
1. Advertise (ADV)
6. Play Connect/Unlock request as User
2. Send ADV/Response packets (*Relay 1*)
5. Send Key Info, Unlock Packet (*Relay 2*)
3. Play ADV/Response packets
4. Key/Connect/ Unlock packets
Lock
Attacker One
Attacker Two
User's Device

Figure 3.3: Designing a Relay attack in a Smart lock

The above figure explains the Relay attack scenario in the case of Smart Locks in terms of BLE communication or packets exchanged using the above components.

The steps involved in a Relay attack are:

Step 1: Attacker one captures the advertising packet (or response packets) from the smart lock.

Step 2: Attacker one transmits the captured information to Attacker two through Relay one cloud.

Step 3: Attacker two relays the received signal to the user's device as a smart lock.

Step 4: The user's device gets fooled into responding to the relayed signal and sends a connection (or unlock packet).

Step 5: Attacker two now captures the key info (or unlock packet) and relays it to Attacker one.

Step 6: Attacker one attempts to connect to the lock using this key info (or unlock packet). The lock gets fooled into believing that the connection is from a legitimate user and responds appropriately. Attacker one now has access to the lock.

The function calls that are made in the Core Bluetooth API when the lock and the user's device communicate with other another are exploited in order to design a solution for a relay attack, which is explained in detail in the implementation section of Relay attacks.

3.1.4 Design of a Secure Access Extension mechanism

The access revocation and log evasion attacks occur in case of the DGC architecture since the smart lock has no capability for decision making and solely relies on user's device to authenticate it. Hence, this dependency on the user's device can be removed by incorporating the decision-making capability to the lock.

In current systems, where the access works by pushing an authenticated certificate to the user's device and then a revocation certificate to the same device for revoking access, the use case is replaced by the owner phone pushing the access token details to the lock through BLE communication.

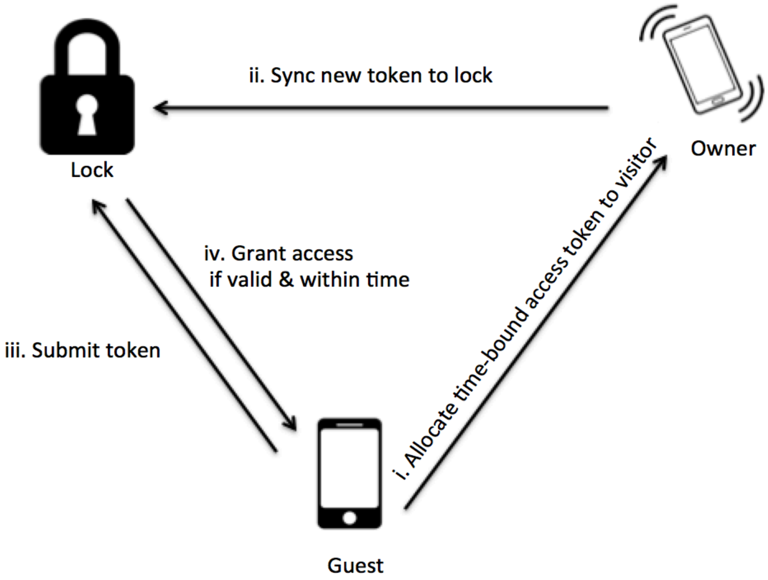
ii. Sync new token to lock
Lock
Owner
iv. Grant access
if valid & within time
iii. Submit token
i. Allocate time-bound access token to visitor
Guest

Figure 3.4: Designing secure access extension mechanism in a Smart lock

In this model, the owner first generates a new unique token when he wants to give access to a guest user. This token is shared with the guest and is synced to the lock. The validity period of the token: “from” timestamp, “to” timestamp are also synced to the lock along with the new access token. When the guest authenticates, he pairs with the lock on BLE using the access token provided to him. The lock checks whether the token is valid and then whether the access token falls within the validity period and provides or rejects access accordingly.

3.1.5 Design of Replay attack model

Replay attacks are carried out by playing an authentication signal to the authenticating device, which was captured in the past from a legitimate device. To address this scenario in the case of Smart locks, the current time stamp can be appended before each authentication signal. The receiving party calculated the difference between timestamps of the authentication signal and its current timestamp to determine whether the signal is a original or replayed signal.

3.2 IMPLEMENTATION

This section discusses the implementation details of the design overview given in section 3.1.

3.2.1 Implementing a Basic Smart lock

The concept of peripheral and central is extended to a mobile device for implementing a basic smart lock. Starting with extending the role of peripheral to the iOS device, it starts with identifying the characteristics of peripheral with respect to smart lock properties of a smart lock. The two properties that define the smart lock includes: lock/status of lock and digital keys. These two properties determine who has access to the lock, the authentication phase and the how the lock status is accessed/modified.

These two properties are identified as the token characteristic and lock characteristic. The token characteristic imitates the property of "digital keys" and the lock characteristic, the lock property of the smart lock. These characteristics are added as a service to the peripheral manager before it starts advertising its data/services. The central is programmed to get the handle of the token characteristic first and get authenticated before it tries to write to the lock.

Steps at peripheral end:

1. Create a peripheral object and initialize it

2. Create the mutable lock and token characteristics with Unique Universal Identifiers (UUID) and add them as services before the peripheral starts advertising its data. The service UUID is made a note of, which is the one the central scans to.
3. Handle the appropriate reads/writes to the characteristics.

Steps at Central end:

1. Initialize the central manager object.
2. Scan for the peripheral of interest with the service identified at the peripheral end.
3. After the services and characteristics are discovered the following steps take place at the central end (in didDiscovercharacteristics() function) :Get a handle of the discovered characteristics
   1. Get a handle on the characteristics.
   2. Perform a write to token characteristic by passing the digital key. This would be the authentication phase. If the write fails, the connection is immediately terminated.
   3. If the token write succeeds, an initial write is done to lock with an "UNLOCK" value and the timer is switched on.
   4. Enable the timer to read the RSSI value continuously at intervals. If the RSSI threshold falls within a specific interval, the "LOCK" value is written to the lock characteristic else an "UNLOCK" value/status is maintained at the lock.

Accordingly the smart lock is either locked/unlocked.

Alternatively at the peripheral end, the following actions further take place when the write to token from central to lock is received:

1. It is a first write (when the lock is installed for first time): The requested value is persisted locally to peripheral and this acts as the "owner digital key".

2. Not a first write to lock/peripheral: The value received from central is compared with the already existing value for token or a set of user tokens. If the key matches to a owner or any user token, it authenticates the central and responds with a success message.

This is how the authentication through "digital keys" is carried out using Core Bluetooth API.

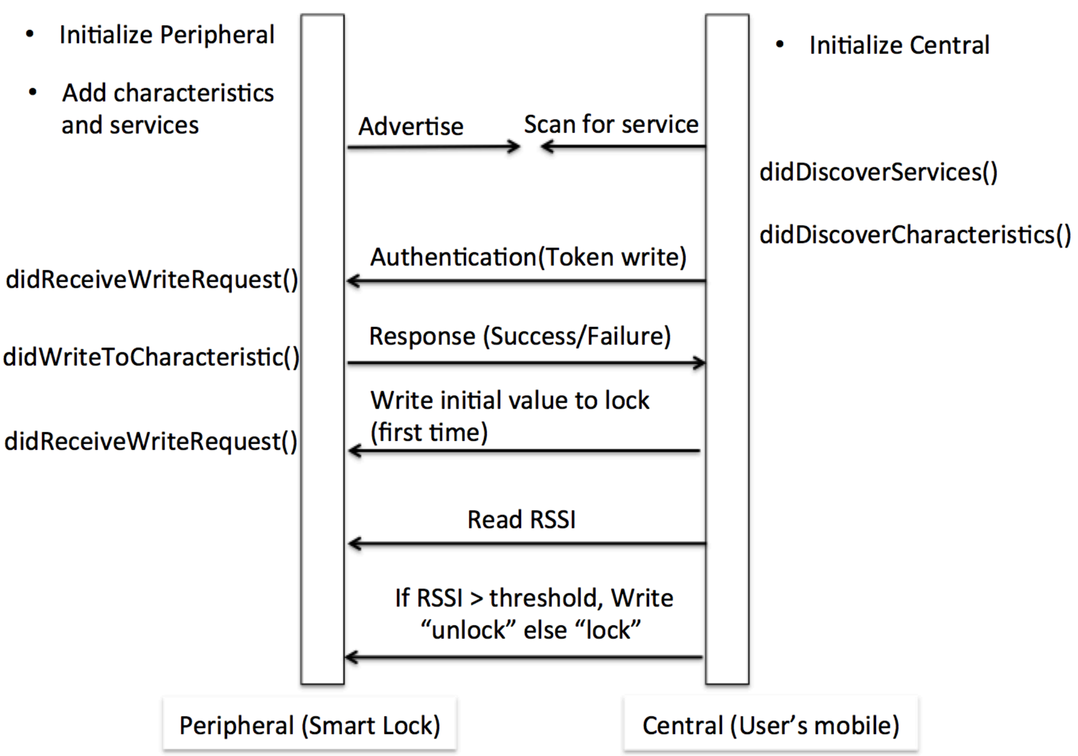
• Initialize Peripheral
• Add characteristics and services
• Initialize Central
Advertise
Scan for service
didDiscoverServices()
didDiscoverCharacteristics()
Authentication(Token write)
didReceiveWriteRequest()
Response (Success/Failure)
didWriteToCharacteristic()
Write initial value to lock (first time)
didReceiveWriteRequest()
Read RSSI
If RSSI > threshold, Write "unlock" else "lock"
Peripheral (Smart Lock)
Central (User's mobile)

Figure 3.5: Smart lock implementation using Core Bluetooth API

The communication model of the smart lock system between a peripheral and central is given out as a sequence diagram in figure 3.5.

3.2.2 Mitigating Relay Attacks

From the design overview of section 3.1.1, it can be observed that there are a set of function calls that are made on the peripheral (lock) during connection establishment, authentication phase and while locking/unlocking the lock. These function calls can be exploited to establish a timing threshold for the detection of relay attacks in smart lock systems, which were introduced in section 2.3.1.

Relay attacks are not easy to carry out practically because of the use of sophisticated hardware to avoid signal delays to successfully fool the parties. The attackers should be using a high-speed channel to transfer the packets between them in order to carry out the attack successfully. For this scenario, the attack has been carried out as a software implementation model instead of hardware.

The different components used in carrying out a programmatic approach of the attack are explained below:

Technologies Used: Core Bluetooth, Kinvey (Cloud), four iOS devices.

Kinvey Cloud as a Packet Transfer: The packet transfer between Attacker One and Attacker Two can be done through Wi-Fi or cloud. Since Wi-Fi transmission needs both parties to be on same network, the cloud approach is used for transferring packets between parties involved in this scenario.

The Cloud used here is Kinvey MBAAS (Mobile as a backend service) as its speeds approximates to that of a high-speed channel. Storing or retrieving from Kinvey cloud is very fast and the implementation is also easier.

Two tables referred to as “collections”: Relay 1 and Relay 2 are maintained at Kinvey cloud to aid in packet transfer between parties. The collections contain two pieces of information:

identifier of packet (a random number) and the message (which can be either advertisement packet or unlock packet or connection packet).

Relay 1:

Interface between Attacker one and Attacker two. Used by Attacker one to transfer the captured packets from the lock to Attacker two to replay them to legit user's device.

Relay 2:

Interface between Attacker one and Attacker two. Used by Attacker two to send back the packets captured from the lock to Attacker one.

The implementation of the same design is divided in detail into two steps: "Capturing authentication packet" and "Capturing unlock packet" from a User's Device. The steps are described next. Figure 3.6 outlines the steps

Step 1: Fooling the User's Device and Capturing Connection/Digital Key packet

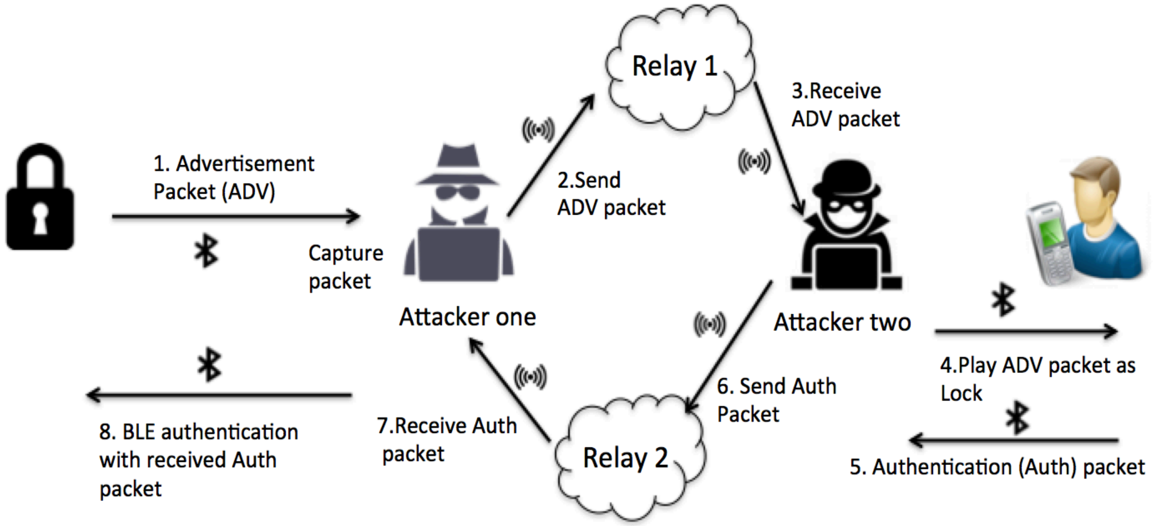
Relay 1
3.Receive ADV packet
1. Advertisement Packet (ADV)
2.Send ADV packet
Capture packet
Attacker one
Attacker two
4.Play ADV packet as Lock
6. Send Auth Packet
8. BLE authentication with received Auth packet
7.Receive Auth packet
Relay 2
5. Authentication (Auth) packet

Figure 3.6: Capturing Authentication packet in Relay attack

Step 1: Capturing Advertisement packet:

Lock advertises its manufacturer's data and the services offered by it. It is assumed Attacker one would be actively listening for the advertisement packets from the lock.

Attacker one actively scans for these advertisement packets continuously. Whenever an advertisement packet becomes available, it captures the advertisement packet. This completes the advertisement packet capture.

Steps 2,3: Relaying Advertisement packet to Attacker two:

As soon as the attacker one reads the advertisement packet from the lock, it writes the captured packet to "Relay 1". At the other end, Attacker two is actively scanning for packets written by Attacker one to "Relay 1". Whenever a packet becomes available, Attacker two reads it. This completes this step.

Steps 4,5: Relaying the Advertisement packet to a User's mobile device:

Attacker two acts as a lock to the user's device. Hence, it transmits BLE signals to the user's device using the captured advertisement information. The user's device by default scans for a particular "advertisement" packet (Name, Service Id's, etc.). Since the attacker two plays the packet with the same information, the user's device gets tricked into believing that the packet is from a legitimate lock. It then attempts to connect to the lock using its digital key or token. Attacker two would then receive this packet and immediately sends it to Attacker one using the Relay 2 cloud.

Steps 6,7,8: Relaying the Key info/Authentication packet to lock:

As soon as Attacker one sends the captured advertisement packet to Attacker two, it switches back to active listening mode on Relay 2 cloud. As soon as it finds the token packet, it retrieves it and plays it to the lock as a legitimate user.

This completes part one of a Relay attack.

Step 2: Fooling the Lock and Getting Access to the lock

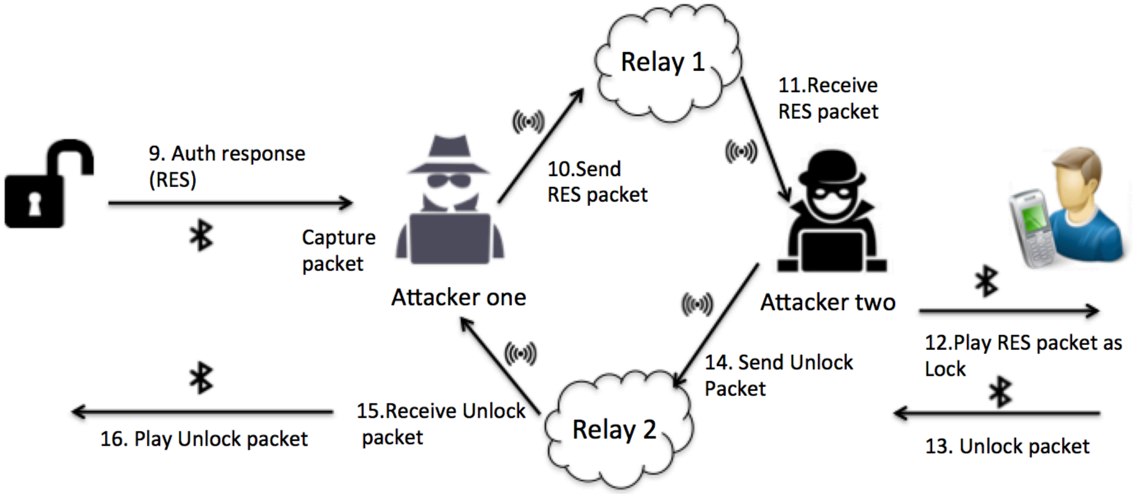

Figure 3.7: Capturing Unlock packet in Relay attack

Steps 9,10: Relaying the response of Auth packet from lock to Attacker two:

After the authentication (digital key/token) write, the response of the write is sent back to the central. This response is received by Attacker one in this case and is sent to Attacker two through Relay one.

Steps 11,12: Retrieving the Auth response packets and relaying it to User's device:

Attacker two would be actively listening on Relay one for the response packet. As soon as it receives it, it plays the response signal to the user's mobile device. Since the authentication is done using the user's digital key, the response packets validates to success. This response is relayed to the user's mobile device.

Step 13: Capturing the unlock packet from the user's device:

As soon as the response is played to the user's device, and since the response is of "Success" type, the user's device sends an unlock packet next to the lock which is attacker two in this case. This packet is captured by Attacker two and sent to Attacker one through Relay 2.

Steps 14,15,16: Retrieving the unlock packet and relaying to the lock:

As soon as the Attacker one receives the unlock packet at Relay two, it retrieves this packet and plays it to the lock. The lock state is then updated to "Unlock" state as soon as it receives the "Unlock" write to the lock characteristic. And thus the Attacker one gains access to the lock.

This completes the relay attack.

3.2.2.1 Identifying Relay attacks

The above relay attack can be viewed with respect to BLE communication as shown in the following sequence diagram:

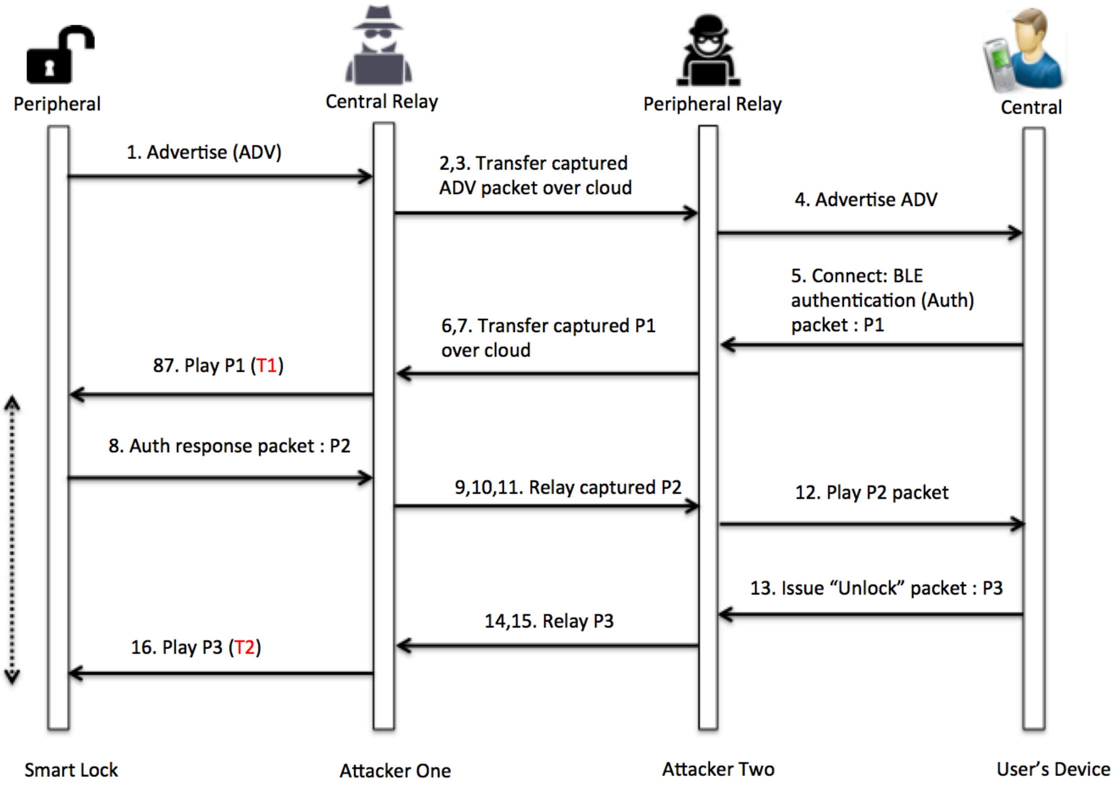
Peripheral
Central Relay
Peripheral Relay
Central
1. Advertise (ADV)
2,3. Transfer captured ADV packet over cloud
4. Advertise ADV
5. Connect: BLE authentication (Auth) packet : P1
6,7. Transfer captured P1 over cloud
87. Play P1 (T1)
8. Auth response packet : P2
9,10,11. Relay captured P2
12. Play P2 packet
13. Issue "Unlock" packet : P3
14,15. Relay P3
16. Play P3 (T2)
Smart Lock
Attacker One
Attacker Two
User's Device

Figure 3.8: Sequence diagram of Software approach to Relay attack

It can be seen from Figure 3.8 that the signals are transmitted over Wi-Fi/cloud in case of a Relay attack. Instead of direct communication between two parties over BLE, the signals are transmitted back and forth over the network. For steps (7) and (13), we have a set of function calls at the lock, which are called for the received actions. In the normal case, the delay between step (7) and (13) is negligible, but the difference in timing is significant in a relay attack. The time difference can be measured at the lock end when a particular set of BLE calls are invoked at lock. By recording the time difference between these steps, the delay that is introduced between step (7) and step (13) (delay: T1 – T2) can be tracked for determining the presence of a Relay attack if the delay is above a particular threshold value.

3.2.3 Mitigating Access Revocation and Log Evasion Attacks

Looking at the implementation details for the design mechanism discussed for counteracting access revocation attacks in section 3.1.3, the access token exchanged between the owner and a guest user can take place through Wi-Fi communication. For the current scenario, Kinvey cloud is used to exchange the token information between the owner and the guest. A unique random access token is generated at the owner end and exchanged with the guest over the cloud. The owner can then sync the same token to the lock by issuing a write to the "token characteristic" on the lock. Only the owner has the ability to sync the access tokens to the lock. The token characteristic write can be in any of the below two formats:

i. AccessToken#CurrentTime

ii. AccessToken#CurrentTime@GuestToken_FromTimestamp_ToTimeStamp

The former case is a regular authentication step for the owner or a guest. The latter format is used when the owner wants to sync a new access to lock. The received request value for the token characteristic is checked for the presence of "@". If there is one, the request is interpreted as a new access token sync request else an authentication request.

The string before "#" is the access token of the communicating mobile device. The sync is performed only if this token matches the owner token maintained by the smart lock. The new token along with its validity time is written to the lock if the request is from the owner.

3.2.4 Mitigating Replay Attacks

Replay attacks are comparatively easy to handle when compared to that of Relay attacks. It is believed that the access token signal is captured and is used at a later time for authentication. To counteract this scenario, a timestamp string is used along with the access token, Eg: AccessToken#CurrentTimestamp) during a write to token characteristic or during the authentication phase. At the receiving end, the lock checks for the current timestamp and if the difference between the timestamps of the received request value and the current one is higher than the threshold, the connection is rejected or cancelled. The choice for this time stamp can be decided by the implementing manufacturer to be in order of a few milliseconds to a few seconds.

To mitigate clock skew where the lock might have a different time than the user's device, the lock will sync its clock with the user's device after each successful authentication.

CHAPTER IV

EXPERIMENTAL RESULTS

Section 4.1 discusses the device used and their specifications used for the implementation of the proposed model. The automatic unlocking feature of the lock requires a threshold to be established for the signal strength below which, the lock remains locked and above which the lock will be unlocked.. A series of experiments for determining the same are carried out which are discussed in section 4.2. The timing threshold for accepting or rejecting the communications in case of Relay attacks is also discussed next in section 4.3.

4.1 SETUP

Two iOS devices are used in designing a smart lock and four in carrying out a Relay attack. The specifications of the iOS devices include:

Device/Parameters	***Device 1***	***Device2***
Name	*iPad Mini 3*	*iPad 4G 2*
iOS Version	*10.2.1*	*10.2.1*
BLE Version	*Bluetooth 4.0*	*Bluetooth 4.0*
Used As	*Lock*	*User's device*

Table 4.1: Device specifications of iOS devices used as lock and user's device

Device/Parameters	***Device 3***	***Device 4***
Name	*iPad (4th Generation)*	*iPad (4th Generation)*
iOS Version	*10.3.1*	*10.3.1*
BLE Version	*Bluetooth 4.0*	*Bluetooth 4.0*
Used As	*Attacker 1/Guest*	*Attacker 2*

Table 4.2: Device specifications of iOS devices used as Attacker 1 and Attacker 2 in Relay attack

4.2 DETERMINING THRESHOLD SIGNAL STRENGTH FOR AUTOMATIC UNLOCKING

The automatic unlocking feature of the smart lock enables to unlock the door when the user's mobile device is within the proximity of the smart lock. In order to establish a threshold for unlocking the door when the user is within a particular range, the signal strength of the BLE is taken into consideration.

The signal strength or the RSSI value of the smart lock is measured at varying distances by the user's mobile device and when the signal falls within or below a particular threshold, an unlock request is issued to the lock by the device and when the user's device moves away, a lock request is issued to the lock.

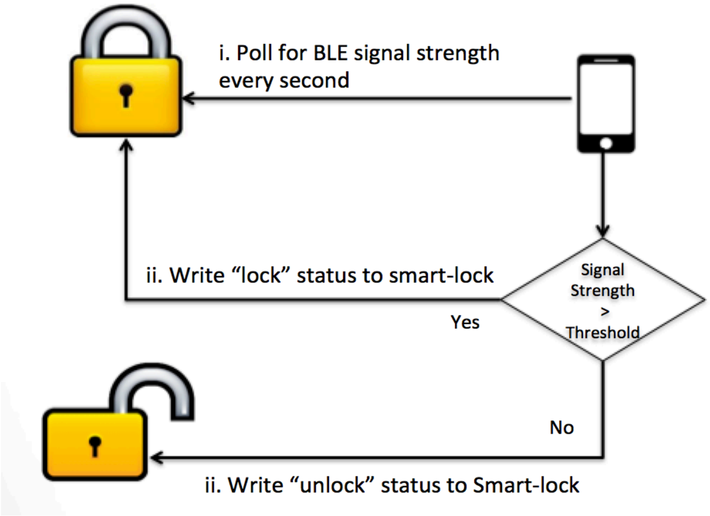
i. Poll for BLE signal strength every second
ii. Write "lock" status to smart-lock
Signal Strength > Threshold
Yes
No
ii. Write "unlock" status to Smart-lock

Figure 4.1: Automatic unlocking in Smart lock on signal strength

Automatic unlocking of the smart lock is achieved by maintaining a signal strength threshold. To establish an optimal threshold, two sets of experiments have been carried out, by varying the distance between lock and the user's device.

Experiment 1: Estimating a balance between signal strength and distance

The distance between the smart lock and the user's device is varied and the unlocking distance (below which the lock automatically unlocks) for particular signal strength is measured. Different values of signal strengths are taken into consideration ([-80 dB, -10 dB]) and the unlocking distance obtained is noted down. The devices are placed side by side and in a top-bottom order.

The distances measured are in order of centimeters.

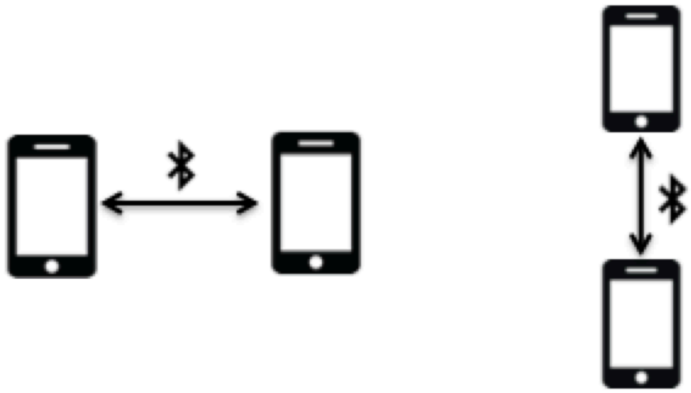

Figure 4.2: Side-Side and top-bottom placement of devices

a. With respect to sides

The devices were placed side by side and the distance between them is varied. The signal strength is varied between [-70 dB, -10 dB] and the unlocking distances obtained were in the range of: [0, 71] centimeters. The devices were placed side by side and the distance between them is varied.

b. *With respect to top and bottom*

The devices were placed in a top-bottom order and the distance between them is varied. The signal strength is varied between [-70 dB, -10 dB] and the unlocking distances obtained were in the range of: [0, 62] centimeters.

From the above experiments it has been estimated that the ideal signal strength can be taken in the range [-50 dB, -40 dB].

Experiment 2: Estimating ideal signal strength for unlocking distance

The average or the mid-range signal strengths from Experiment 1 fall in the range: [-45 dB, -50 dB]. The unlocking distances were measured by fixing the threshold value of signal strength in this range and the corresponding upper bound of unlocking distance obtained in each case was noted down.

The graph below gives a plot the unlocking distances measured against the signal strength: -50 dB. It can be observed from the graph that the unlocking distances are obtained in the range: [0, 47] centimeters.

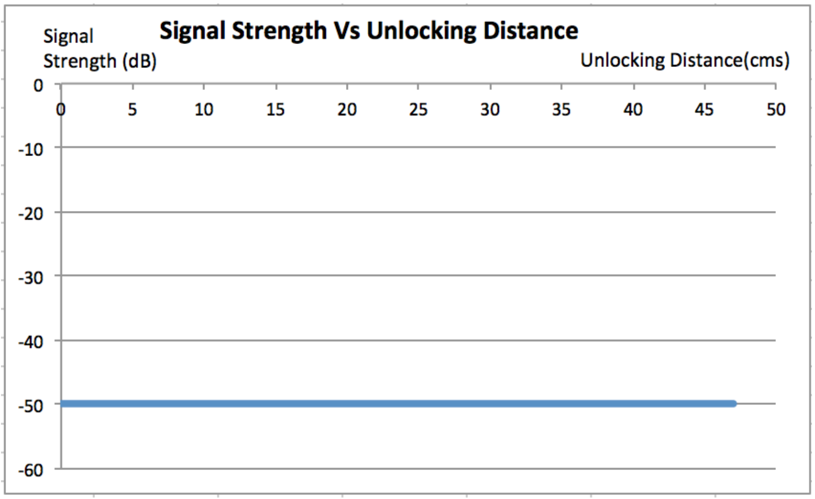

Figure 4.3: Unlocking distances for signal strength: -50 dB

From the above experiments, the threshold signal strength has been established as -50dB for the current model and the unlocking distance can be achieved up to 40-50 centimeters (an ideal unlocking distance) depending on environmental factors along with signal strength.

4.3 DETERMINING TIMING THRESHOLD TO COUNTERACT RELAY ATTACKS

Timing analysis has been carried out between the token write (authentication step) and an initial lock write step to distinguish between a normal and relayed communication. The results of the timing analysis for a normal and relayed communication are given below:

i. Normal Communication (Distance: [0,47] centimeters)

Token Write Time (Authentication Step)	**Lock Write Time**	**Difference (seconds)**
1498066955.9676	1498066956.1149	0.1473
1498067028.3621	1498067028.4816	0.1195
1498067090.0328	1498067090.1528	0.1201
1498067174.8817	1498067174.9985	0.1168
1498067218.9464	1498067219.0652	0.1188
1498067271.0528	1498067271.1713	0.1185
1498539193.0473	1498539193.1942	0.1470
1498539313.7848	1498539313.9359	0.1511
1498539389.4593	1498539389.5783	0.1190
1498539459.7994	1498539459.9197	0.1203
1498539525.2343	1498539525.3529	0.1186
1498539585.6240	1498539585.7427	0.1187
1498539665.8701	1498539665.9895	0.1194
1498539769.0325	1498539769.1515	0.1190
1498539837.9498	1498539838.0693	0.1195
1498539923.6849	1498539923.8023	0.1174
1498539982.1196	1498539982.2388	0.1192
1498540054.3665	1498540054.4850	0.1185
1498540125.3993	1498540125.5478	0.1486
1498540204.6057	1498540204.7552	0.1494
1498540309.4089	1498540309.6474	0.2385
1498540431.2746	1498540431.5734	0.2988
1498540507.7390	1498540508.0993	0.3603
1498540604.2774	1498540604.4273	0.1498
1498540789.3981	1498540789.6364	0.2383

Table 4.3: Timing analysis for normal communication in Smart lock

ii. Relayed Communication (Distance: [0,47] centimeters)

Token Write Time (Authentication Step)	**Lock Write Time**	**Difference (seconds)**
1498063956.7795	1498063957.2861	0.5066
1498064197.5975	1498064198.1063	0.5088
1498064310.5776	1498064311.0867	0.5091
1498064573.2833	1498064573.8490	0.5657
1498064704.3920	1498064704.8711	0.4791
1498064828.7206	1498064829.2596	0.5390
1499347943.6469	1499347944.3963	0.7494
1499348073.8322	1499348074.2515	0.4193
1499348270.3535	1499348270.8028	0.4493
1499348380.3264	1499348380.7453	0.4189
1499348494.4383	1499348494.8880	0.4497
1499348681.9285	1499348682.3479	0.4195
1499348842.3891	1499348842.9287	0.5396
1499348962.0777	1499348962.5271	0.4494
1499349124.6716	1499349125.3009	0.6293
1499349255.4612	1499349256.2104	0.7492
1499349388.4534	1499349388.8723	0.4189
1499349624.4864	1499349624.9057	0.4193
1499349747.0283	1499349747.8672	0.8388
1499349854.3432	1499349854.8525	0.5093
1499349947.7937	1499349948.3027	0.5091
1499350034.4621	1499350034.9112	0.4492
1499350133.3963	1499350133.8456	0.4493
1499350273.7749	1499350274.2641	0.4892
1499350440.7796	1499350441.1773	0.3977

Table 4.4: Timing analysis for a relayed communication in Smart lock

It can be observed from the above results that the time difference between a token write and an initial lock write can range between: 0.1 – 0.4 milliseconds in case of a normal communication but is always higher than 0.35 milliseconds in case of a relayed communication. Also, the higher value for a normal communication can account for the scenario where a higher distance separates the user's mobile from the lock. Even if the initial communication is cancelled, as the user approaches the mobile, the timing falls within the 0.1-0.4 milliseconds again. Taking this into account, establishing a time threshold as 0.35 milliseconds takes care of almost all of the relay attacks. The connection can simply be cancelled in this case.

CHAPTER V

CONCLUSION

In this paper, the need to secure IoT devices has been discussed by taking a subset of the home automation field of the IoT: Smart lock as an example. Two different architectures currently existing for the smart lock systems were introduced and the attacks that are possible by exploiting their vulnerabilities were discussed.

A defense mechanism for both the attacks has been presented, that is built on one of the currently existing architectures. In one, timing analysis has been proposed as a defense mechanism by establishing a timing threshold. In the other, the access extension model is slightly modified to eliminate dependencies to counteract these types of attacks.

In both these mechanisms, there was no additional hardware introduced, but the BLE capabilities have been exploited in designing a secure smart lock system. Also, since most of the smart locks follow a similar kind of architecture, the defense mechanism designed has been targeted at an architectural level of the smart lock

Lastly, it is a known fact that there is always a trade-off between usability and security or availability and security. A trade-off has been established between them in order to not compromise any of them.

Thus, by establishing a fine line between the usability and security, this project has enabled the design of a secure smart lock operating on BLE, the cheaper and easier mode of communication, and in counteracting the most common attacks that can occur in a smart lock.

REFERENCES

[1] August. http://august.com/

[2] L. Bauer, S. Garriss, J. M. McCune, M. K. Reiter, J. Rouse, and P. Rutenbar. Device-enabled authorization in the grey system. In International Conference on Information Security, 2005.

[3] I. Boureanu and S. Vaudenay. Challenges in distance bounding. Security & Privacy, IEEE, 2015.

[4] E. Brewer. CAP twelve years later: How the “rules” have changed. Computer, 2012.

[5] Danalock. http://www.danalock.com/.

[6] S. Drimer and S. J. Murdoch. Keep your enemies close: Distance bounding against smartcard relay attacks. In USENIX Security, 2007.

[7] C. . Ericsson. https://www.youtube.com/watch?v=pJ5fSWspBpo.

[8] N. Forum. http: //nfc-forum.org/what-is-nfc/about-the-technology/.

[9] A. Francillon, B. Danev, S. Capkun, S. Capkun, and S. Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In NDSS, 2011.

[10] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis. Practical NFC peer-to-peer relay attack using mobile phones. In Radio Frequency Identification: Security and Privacy Issues.

2010.

[11] Z. Kfir and A. Wool. Picking virtual pockets using relay attacks on contactless smartcard. In Security and Privacy for Emerging Areas in Communications Networks (Secure Comm), 2005.

[12]A.Levi,E.C ̧etinta ̧s,M.Aydos,C.K.Koc ̧,and M.U. C ̧ag ̆layan. Relay attacks on Bluetooth authentication and solutions. In Computer and Information Sciences (ISCIS). 2004.

[13] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, David Wagner. Smart Locks: Lessons for securing Commodity Internet of Things Devices. 2016

[14] Lockitron. https://lockitron.com/.

[15] Kevo. http://www.kwikset.com/kevo/default.aspx.

[16] K. B. Rasmussen and S. Capkun. Realization of RF distance bounding. In USENIX Security, 2010.

[17] M. Seyedi, B. Kibret, D. T. Lai, and M. Faulkner. A survey on intrabody communications for body area network applications. IEEE Transactions on Biomedical Engineering, 2013.

[18] http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[19] https://techcrunch.com/2016/08/08/smart-locks-yield-to-simple-hacker-tricks/

[20] Borgohain, Tuhin, Uday Kumar, and Sugata Sanyal. "Survey of security and privacy issues of Internet of Things." arXiv preprint arXiv:1501.02211 (2015).

[21] https://developer.apple.com/documentation/corebluetooth

VITA

Palle Saiprasanna

Candidate for the Degree of

Master of Science

Thesis: SMART LOCKS: EXPLORING SECURITY BREACHES AND ACCESS EXTENSIONS

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July 2017

Completed the requirements for the Bachelor of Technology in Computer Science at G. Narayanamma Institute of Technology and Science, Hyderabad, Telangana, India in 2012.

Professional Experience: 3 years