Towards Seamless and Secure Mobile Authentication

by

James Tyler Romo

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2014 by the
Graduate Supervisory Committee

Gail-Joon Ahn, Chair
Partha Dasgupta
Winslow Burleson

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

With the rise of mobile technology, the personal lives and sensitive information of everyday citizens are carried about without a thought to the risks involved. Despite this high possibility of harm, many fail to use simple security to protect themselves because they feel the benefits of securing their devices do not outweigh the cost to usability. The main issue is that beyond initial authentication, sessions are maintained using optional timeout mechanisms where a session will end if a user is inactive for a period of time. This interruption-based form of continuous authentication requires constant user intervention leading to frustration, which discourages its use. No solution currently exists that provides an implementation beyond the insecure and low usability of simple timeout and re-authentication. This work identifies the flaws of current mobile authentication techniques and provides a new solution that is not limiting to the user, has a system for secure, active continuous authentication, and increases the usability and security over current methods.

ACKNOWLEDGEMENTS

I would first like to thank my research advisor, Dr. Gail-Joon Ahn, for all his support and guidance throughout my thesis. I appreciate all the time and dedication that went into helping me make my vision a reality. Working in the Security Engineering for Future Computing (SEFCOM) laboratory has been a great learning experience. Without your patience and understanding, this work could not have been accomplished.

I would also like to thank Dr. Winslow Burleson for being part of my committee and providing insight into the usability portion of the research as well as his students in his Ubiquitous Computing course for their wisdom and support as they gave insight deriving from numerous fields of study.

I would like to thank Dr. Partha Dasgupta for providing security-related contributions to the thesis committee. His input was instrumental in ensuring the validity of the solution.

Finally, I would like to thank my friends and family who are always there for me, providing support and belief even in times when I questioned my own abilities. You have all built me up to where I am today.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

DEFINITION OF TERMS

Mobile Authentication: The act of confirming the identity of a person on a mobile device during an expired session.

Initial Authentication: The part of authentication where the user normally enters their credentials and gains access to the system.

Active Session: The period after the initial authentication where a user is able to freely use the system with the identity they were authenticated to use.

Expired Session: When a user is not authenticated and is presented with the authentication (lock) screen.

Continuous Authentication: Additional authentication performed during device use. This is traditionally done using an inactivity timeout where the session will expire if the device has been active for a set period of time. At this point, the user is presented with the initial authentication screen again.

Inactivity-based Continuous Authentication: Only authenticates when the session expires (timeout due to user inactivity), meaning if the attacker is able to gain physical access to the device, they can maintain the session indefinitely.

Active Continuous Authentication: Authenticates during both active and expired sessions on a set interval. If it fails, the session will become or remain expired. This technique requires the authentication method to be non-interrupting so that the user does not have to stop their current activity to continue their session.

CHAPTER 1: INTRODUCTION

The initial authentication and inactivity-based continuous authentication of current mobile devices fail to provide mobile users with the usability they desire. Internet, social media, and other communication are just one click away in this modern age; however, so are all their inherent risks. Numerous applications, both for pleasure and productivity, have been released causing mobile device use to emerge as a key part of many users' daily activities. Why then do so many people leave those devices unprotected? Most people know that if their device is lost without any sort of security, anyone can access not only anything that the device holds, but also the history of what the device has been used to access and even the data that applications on it are able to access. This includes any search history, location information, email, private messages, social media applications, and potentially even more dangerous information such as banking information.

Despite this high ceiling of risk there is "an estimated 38% to 70% of users (that) do not lock their mobile phones and tablets" [1]. People either don't feel the need to secure their devices or don't want to take the time to go through the unlock steps each time they want to perform a simple task such as opening a text message. This process of confirming the identity of a person on a mobile device is called mobile authentication and is usually done through a lock screen. The security screen, usually requiring a password or PIN, is displayed to the user when they first turn on the device or if they were inactive for a certain period of time.

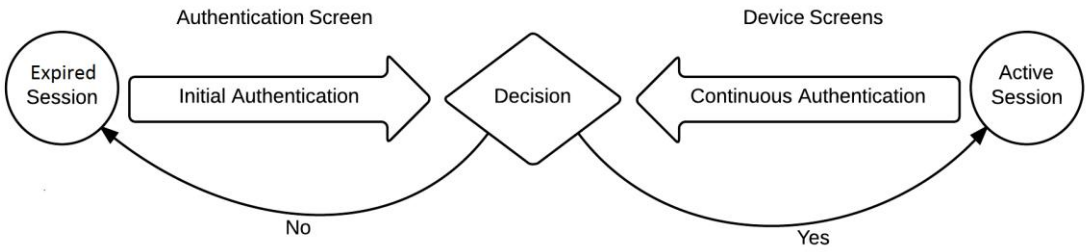


Figure 1-1: Mobile Device Authentication States

As displayed in Figure 1-1, beyond initial authentication, active sessions are maintained using continuous authentication in the form of optional timeout mechanisms where a session will end (become inactive) if a user doesn't interact with the device for a set period of time [5]. However, this security technique has two major flaws. First, the lock screen is highly unusable to the user, forcing them to take the time to enter in their password on a device that is meant for fast, easy information. While it might not seem like a long time, entering credentials on an error-prone virtual keyboard each and every time a user needs to perform a quick task on their device can be annoying enough that they choose to disable authentication entirely.

Secondly, the technique uses inactivity-based continuous authentication to determine when a user should re-enter their credentials, leaving a window for unauthorized use [5]. This form of authentication only verifies when the session expires (times out due to inactivity), meaning that if the attacker is able to gain physical access to the device during an active session, they can maintain the session indefinitely. Active continuous authentication, on the other hand, validates during both active and expired sessions on a set interval, meaning that even if the attacker is able to access the device,

they will be locked out when it re-authenticates despite still interacting with the device, which effectively removes this vulnerability. The technique requires that the authentication method is non-interrupting so that the user does not have to stop their current activity to continue their session, therefore requiring a solution different from the traditional lock screen. In addition, when using inactivity-based continuous authentication, the user is able to change the inactivity timeout allowing them to replace the shorter, more secure default with a long timeout duration. This means the attacker is better able to steal the device during an active session and access the device.

Despite these major flaws, all current solutions use inactivity-based continuous authentication requiring constant user intervention, which leads to users either use their security improperly or not at all. Until a new solution is adopted that abandons the traditional use of a lock screen, their problems will linger. Due to the high rise in mobile devices, the need for a continuous monitoring solution that does not hinder usability and actively authenticates is essential to minimize the loss of confidential information resulting from unauthorized device use.

This research provides an authentication technique where security is not limiting to the user, has a system for active continuous authentication, and increases the usability and security over current methods. To find a technique that properly fixes the flaws of mobile authentication, each of the current techniques are individually evaluated based on attributes in usability, deployability, and security. This idea is based on the evaluation framework in the paper by Bonneau, Herley, Oorschot, and Stajano [2] where numerous forms of web based user authentication are compared with the purpose of finding a solution that performs better than traditional passwords.

These three areas are essential to assessing mobile authentication and cover all necessary areas of evaluation. Security, the most obvious category, covers the risks and concerns associated with mobile authentication. No matter how innovative or clever a new solution is, if it does not pass the necessary security requirements, the authentication is worthless. However, as previously stated, current mobile devices are being left unsecured not due to insufficient security, but because of the time it takes to use. For this reason, the usability category is required to evaluate if an end user will be willing to use each authentication technique. This includes aspects such as the technique requiring little user memory and can be completed in a short amount of time. The usability category ensures that even if a technique achieves perfect security scores, it will not be deemed the best technique if it is not usable because in the end, a perfectly secure technique is pointless if it is never used. Although the security and usability categories cover the basic user need, the aspect of getting the new authentication techniques into the market is still a concern. The deployability category covers how easily the technique can be integrated into current mobile architectures. For instance, a new technique that integrates with a current phone screen will have high deployability as long as it is compatible with current devices, does not require any additional cost, and fits into current security models. An example of low deployability would be something that requires a custom hardware solution and does not function like current mobile authentication techniques.

Despite the strengths of the evaluation framework created in the paper by Bonneau, Herley, Oorschot, and Stajano [2], without quantitative measurements, the results of the evaluation are much less convincing especially considering their own thoughts that “the benefits chosen as metrics are not all of equal weight.” This makes the

resulting table difficult to draw conclusions from and determine the strength of each mobile authentication technique. However, one of their main reasons for not assigning weights and scores to each benefit was that the "importance of any particular benefit depends on target use and threat environment" [2]. In this work, the target use is specified much narrower to be authentication on current mobile devices and the threat environment is assumed to be during normal use. This eliminates cases such as when security is of the utmost importance or anything else outside the target audience of typical users for mobile manufacturers.

Schologlhofer and Sametinger apply this idea of an evaluation framework using a generic Android authentication application called SecureLock, which incorporates numerous mobile authentication methods so they can be compared [3]. The main difference between this and the previous framework is that the security and usability benefits in this comparison specifically target concepts related to mobile authentication. This includes understanding concepts such as the touch screen and other technologies such as NFC tags.

In their comparison, each technique is first displayed in a table to show which security mechanisms the technique employs including knowledge, ownership, and inherence. Knowledge is based on "what we know" such as a traditional password that can be remembered, ownership is based on "what we have" including any sort of physical object used for authentication, and inherence is "what we are" covering biometrics and other aspects of a person [3]. After this comparison, security and usability benefits are broken down into specifics to see if each technique covers the area sufficiently, does not cover the area, or almost covers the area. The various techniques and benefits were taken

into consideration when forming the evaluation framework for this work. However, without quantitative measurements, this comparison like the previous came across as high level estimates instead of an in depth evaluation.

With this in mind, a paper by K. Renaud [4] was used to create a quantitative system with appropriately assigned weights to all the categories and benefits. His work applied weights and values to his comparisons giving the results much more detail and complexity. The decisions for the categories, benefits, and weights were based on the previously mentioned existing research into authentication evaluations and studies into the needs and desires of current users. First, each benefit is rated on a scale from 0 to 100 with 100 meaning that it fulfills the benefit completely. Then, a weight is applied to the benefit from 0.0 to 1.0 to determine the importance of the benefit for the category. The weights for each category will add up to a total of 1.0. Once all the benefits are totaled for each category, the values will be multiplied by the weight of each category: .4 security, .4 usability, and .2 for deployability. These totals are then summed up to reach the a total value. This, along with the values for each category, will then be used to compare each mobile authentication technique.

The main idea of this research is that mobile devices should use continuous authentication techniques that do not interfere with the user in order to increase the number of secured devices. However, the hypothesis is that current techniques do not sufficiently fulfill the specified categories so the question remains: can a new mobile authentication technique replace current methods by providing secure and usable continuous authentication? This proposed solution fixes the flaws of current techniques by addressing the neglected usability, security, and deployability concerns. To ensure this

new solution provides benefits to existing authentication, it is compared to various techniques currently used in Android, iOS, and Windows phones and tablets. In addition, “alternative” techniques that are either only used on a few devices or are not currently in production are also evaluated and compared to the proposed solution to ensure that they provide users with the most usable method that is still secure and protects their private information.

CHAPTER 2: BACKGROUND

In this section, works are surveyed that are related to mobile authentication; especially research involving increased usability and continuous authentication.

2.1 Continuous Authentication:

The basis of continuous authentication comes from a paper by Niinuma and Jain [5] where the idea of a user, authorized or unauthorized, gaining access to the session of a signed-on user is presented. This research identifies the process of having an initial login session followed by a timeout-based session as a critical security flaw present on both low and high-security systems. It recognizes the fact that the user must be constantly authenticated during a session, but it must be done in a user friendly, passive manner for it to be usable [5]. Otherwise, the user needs to be constantly prompted to perform some action to confirm their identity. However, this paper uses the user's clothing and facial recognition as a means to maintain a session, which is not as viable in a mobile setting where there is no guarantee of a constant camera with adequate lighting.

Another paper by Feng, Lui, Carbunar, Boumber and Shi [6] attempts to provide the same continuous authentication, but on a mobile device. Their research hinges on the fact that their solution must be transparent to the user during authentication (providing the highest level of usability) and provide continuous identity management during normal user interaction in the form of fingerprint scanners [6]. The alternative solution they propose involves a custom touch display that is able to read partial user fingerprints during normal interactions and constantly check to make sure they are the authenticated user. This is done by placing fingerprint scanners at "hotspots" on the device that they determined were common places that a user would touch. The overall solution

incorporates the scanners to provide both local and remote identity management that balances usability with security to provide a means for continuous authentication. However, this solution requires custom hardware not currently present on mobile devices. This would require mobile manufacturers to completely change how touch screens are designed and created making this a difficult solution to shift to from current platforms.

2.2 Evaluation Framework:

Without an accurate means to evaluate a new solution, there is no way to confidently say a new proposal is superior to mobile authentication techniques that are currently in use. The paper previously mentioned to be the basis of this research's evaluation by Bonneau, Herley, Oorschot, and Stajano [2] aims to create a quantitative framework that can be used to not only evaluate a mobile authentication technique, but also compare it to others in the hopes of finding the best solution. In their research, they break down techniques into categories and schemes and then evaluate them by determining whether various benefits are offered, almost offered, or not offered at all. In addition, they determine whether these benefits are better than passwords, worse than passwords, or are the same in the hopes of finding a solution that can provide higher usability without lowering security. They evaluate technologies that are not used for local mobile authentication or are too proprietary/specific for the purposes of this research, but the manner in which the techniques are evaluated and compared is essential to proving the validity of a proposed solution.

2.3 User Studies:

Another important aspect to consider is how users feel about security. Previous studies relating to security have revealed that not only do people not care about security

during normal use, but they will also go out of their way to dismiss and ignore warnings to complete the task at hand [7]. In one study in particular, users were presented with various situations where security was enabled and disabled. Three attack clues were presented to the victims, which were "indistinguishable from those that a user might encounter during a real attack" [7]. These included removing HTTPS indicators, removing site-authentication images, and presenting a warning page. All 63 participants ignored the HTTPS indicator removal, 58 out of 60 participants (97%) entered their passwords despite no site-authentication images, and 30 out of 57 participants logged in despite the warning page [7]. The results of this study showed that unless the warnings were extremely obvious, nearly all participants would proceed with their current task at hand even during banking applications. In the obvious case, over half of the users still proceeded. This study proves that users are not knowledgeable enough with their security to make educated decisions when given a choice. Using this knowledge, the proposed solution must be usable enough that users will not disable the security even if they don't care or understand its significance.

CHAPTER 3: CURRENT INDUSTRY APPROACHES

3.1 Overview:

In the following section, the current industry approaches for mobile authentication are compared and discussed. These approaches are used by current mobile manufactures to locally authenticate a user usually through some sort of lock screen. Mobile devices include any smart phone or tablet that can be easily transported and carried by its owner.

The current industry security standard for mobile devices involves two mechanisms for authentication: initial authentication via an authentication screen and continuous authentication via a session inactivity timeout [5]. The initial authentication is when the user enters their credentials to prove their identity. This can optionally be disabled to not require credentials to access the device making it vulnerable to unauthorized access. These credentials can be in the form of a dot connecting gesture, a PIN, a password, or even biometrics. The continuous authentication aspect of mobile security maintains the active session of the user. This usually correlates to the display timeout of the device. After the timer runs out, the screen will shut off and put the device to sleep. The device will then become locked and will require the initial authentication credentials before the user can actively use the device again. However, the user can change this session timeout to be very short or extremely long. If the session is too long, it provides the opportunity for unauthorized access by another user when the owner leaves the device unattended. Since a session only ends after user inactivity, if an attacker comes in contact with a device during an active session, they are able to retain access indefinitely and the device will never know it has been compromised.

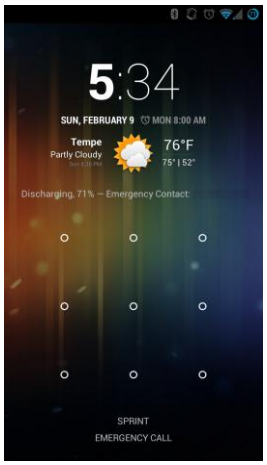
Despite security efforts, many usability problems still plague mobile authentication to the point that many users choose to not lock their devices. No current mobile authentication technique provides a balance between security and usability so that the user does not feel hindered, making this issue an open problem [1]. The key is to create an authentication technique with a balance such that a user is protected, but does not feel the burden of their security. Current mobile authentication techniques can be broken into traditional mobile approaches and alternative approaches. Traditional approaches are those currently in use and established as an effective technique. Alternative approaches include new or unique techniques that attempt to fix the current issues of traditional approaches.

To properly evaluate authentication techniques, an evaluation framework has been constructed based on existing research [2]. The goal is to effectively evaluate and quantify the values of various authentication techniques based on specified criteria. These numerical values can then be compared to decide how well traditional and alternative techniques accomplish mobile authentication goals. The framework uses the following categories for various qualities: usability, deployability, and security. Each category then breaks down into numerous benefits, which are evaluated based on fulfilling the benefit, almost fulfilling the benefit, and not fulfilling the benefit.

3.2 Traditional Approaches:

Traditional approaches are well-established mobile authentication techniques used by current mobile operating systems to allow a user to log into their device. These are currently in production, used on numerous devices, and are familiar to users. The security

community has extensively tested these mechanisms and they have survived malicious attempts of misuse.

a) 

b) 

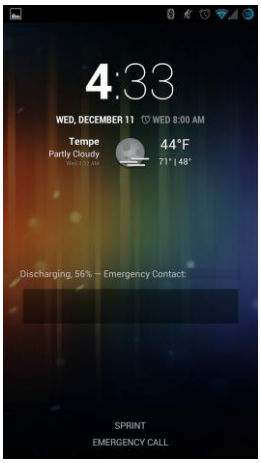
c) 

Figure 3-1: Traditional Approaches a) Android Gesture Lock with 3 by 3 grid b) Android PIN Lock Screen c) Android Password Lock Screen

The most common lock screen for Android devices is a 3 by 3 grid that uses a dot connecting gesture as displayed in Figure 3-1a. The security involves the user setting a swipe gesture to connect the dots on the screen, which is valid as long as it follows some rules. The pattern must connect at least four and at most nine dots, the dots in the pattern must be distinct, and if the connection passes through other dots, they must already be used. This provides 389,112 possible valid combinations of different gestures [8]. Various iterations of Android allow for the possibility of up to a 6 by 6 grid. The device is locked when the display times out or is manually put to sleep. For every five incorrect entries, a 30-second cool down is required before another can be entered [9]. The gesture combination is stored using an unsalted SHA-1 encryption [10].

The main positive of this approach is that this graphical interface is optimized for touch screens, allowing for quick actions that are easy to remember. The inclusion of a

cool down period for unsuccessfully entering a gesture adds difficulty for an attacker attempting to brute force the combination. In addition, this type of password is difficult for a person to describe, meaning that attacks through social engineering, scenarios where attackers attempt to trick a user into revealing their information, are difficult to perform. However, this approach is susceptible to multiple attacks. A brute force attack or dictionary attack is still possible despite the timeout mechanism on both a rooted and un-rooted Android device. Since the gesture key file is unsalted, a Rainbow table could be created to crack multiple devices using this security technique. Even a device with the largest number of dots (9), can be cracked in less than 200 ms [10]. This type of security is also vulnerable to shoulder surfing, where an attacker witnesses the gesture being entered, a smudge attack, where smudges on the screen can reveal the gesture, and weak continuous authentication involving a timeout.

Using a PIN is another traditional approach to mobile security as displayed in Figure 3-1b with similar protection as a dot gesture. This is present in all major mobile operating systems involving some sort of secret numeric password with varying length. Each platform has different requirements for minimum and maximum length and additional security. Apple's iOS has a unique cool down system that scales with the number of unsuccessful attempts making brute force attacks much more difficult than Android. All the platforms still use the same timeout based continuous authentication systems with the same flaws of potentially allowing authorized access depending on user settings [9]. In addition, a longer PIN while more secure, is much less usable with a short PIN being very insecure. The main strength of PIN is that it is quick and easy to remember. However, it has all the flaws of dot gestures with the addition of being

susceptible to social engineering because a simple PIN is easy to send. The short, easy to enter nature of PINs also allow for easy external brute forcing as a robot 3D printed for research purposes that was able to break a 4-digit Android PIN is 19 hours [9]. Additionally, on Android the password key is stored using a salted SHA-1 hash and the salt is stored as a MD5 hash, which can be extracted and used to crack up to 10 digits in an hour [11].

Traditional alphanumeric passwords, the most common authentication technique on desktop computers, are also available on mobile devices as displayed in Figure 3-1c. This type of security has the highest number of possibilities out of the traditional methods making it the most secure to protect against brute force attacks, especially when combined with a cool down mechanism. All major platforms support passwords and users are very familiar with this type of authentication. However, passwords are the most unusable form of authentication on mobile devices due to the difficulties of entering information using a small, on-screen keyboard. In addition, it does not solve the shoulder surfing, social engineering, or continuous authentication problems that plague the previous solutions.

3.3 Alternative Approaches:

Alternative approaches are new or fairly untested mobile authentication methods that attempt to remedy the problems that plague traditional approaches. These include techniques that provide unique methods to quickly authenticate a user in a secure way that does not jeopardize security. They commonly employ new technologies such as biometrics or other attributes easily made available by mobile devices to provide mechanisms unavailable to desktop computers.

Figure 3-2: Alternative Approaches a) Example ‘Hotspots” on an image using PGA b) Apple Fingerprint Scanner c) Android Face Unlock Screen d) RSA Security physical tokens e) NFC Ring

Microsoft introduced Picture Gesture Authentication (PGA) with Windows 8, which expands on the simple dot gesture of Android in the hopes of making something more secure while keeping the touch screen optimized interface. PGA allows the user to choose their own image and create a gesture-based password using a sequence of lines, taps, and circles performed on the image. This provides the user with a much higher ceiling of possible combinations than the Android dot gestures while maintaining all the benefits making it appear to be a strictly better solution. However, studies have shown that, in practice, a user’s picks can be predicted and the attack space reduced by Points of Interest or ‘hotspots’ of the pictures that a user is more likely to create a gesture onto as displayed in Figure 3-2a [12]. This makes a brute force attack much more plausible.

In addition, gesture vulnerabilities such as shoulder surfing, smudge attacks, and the flaws of the continuous authentication method are still present. Unlike traditional gesture authentication, this is also susceptible to social engineering because the pattern can easily be described unlike most 3 by 3 dot gestures meaning the user can accidentally leak the secret to an attacker. It does, however, maintain the other benefits of gesture authentication including having a touch-optimized interface that allows for quick actions that are easy to remember.

Recently, Apple has released a fingerprint scanner with their new iPhone 5S bringing biometric authentication to iOS devices. This scanner, present on the home button, can be used to register any finger as an alternative authentication method with PIN or password primary authentication also being available as displayed in Figure 3-2b [13]. It works by scanning a sub-epidermal layer of the finger making it difficult to replicate. Apple claims that only a 'living' finger would work [13,14]. This method provides quick authentication with nothing for the user to remember. It also protects the user against shoulder surfing, smudge attacks, and makes normal screen timeouts less annoying due to the ease of each authentication.

However, the main problem with any biometric authentication is that once is password is compromised it can never be changed. Even the best passwords are changed regularly for security purposes and even though biometrics use people for authentication, in the end the data is read into the device as data in the form of a password. Apple mitigates this fact in some way by allowing any finger to be used to unlock the device as well as using more advanced biometric techniques than previously used in production [14]. Yet, this method does not solve the continuous authentication issue of mobile

devices because the user is still given an interval where anyone is able to access the device with or without permission and the user is still forced to re-authenticate when a session expires. In the end, all this solution does is provide an alternative way for a user to log in rather than address the problems of the overall approach.

Another alternative approach is the use of face unlock. This is currently in place on Android devices and uses the front-facing camera to authenticate a user's face as displayed in Figure 3-2c. If the user is in an area with poor visibility, a face is not found, or a user cannot be identified, they are provided with an alternative lock screen in the form of PIN or traditional password. This provides many of the positives of the Apple fingerprint scanner, except with a much higher rate of false-positives and false-negatives. Despite being immune to attacks such as shoulder surfing, the use of only a frontal face image is not secure enough to prevent an attack [15]. It is far too easy to spoof a simple camera using a high quality image or another technique. This form of attack is called a photo attack where a photo or short video of sufficient quality can be placed in front of the victim's camera when the authentication mechanism uses frontal face information only [16]. Also, due to lighting and other poor picture conditions common with a mobile device, the user may experience many circumstances where the device denies a valid user causing further frustrations and lowering usability. Additional measures can be take such as requiring a side facing picture as well or requiring the user to blink, but these just make the attack harder and do not solve the issue. Face unlock suffers from the same negatives of other biometrics and still does not solve the continuous authentication process.

A variation of face unlock involves using 180 degree panning stereo images to produce 3D effect as opposed to the usual frontal face information only [16]. Overall, this means more data and more effort required for an attacker to break the system. However, this also means a less usable solution for the user who will have to take much more time to unlock their devices when compared to the normal face unlock. Despite the higher security, the user would have to take the time to take numerous images of themselves in a manner that is awkward to do in public. This process is highly unusable and taxing, defeating the purpose of a quick authentication technique despite the security increases.

One alternative approach that has gone through numerous iterations is the physical key or token. The original market leader of this concept RSA SecurID [17] involves the use of keeping a physical token that contains a shared secret that can be used to authenticate a user. The device simply contained a display, which it would use to display a cryptographically strong PIN every 60 seconds using a secret seed stored on the device as displayed in Figure 3-2d [17]. It could then be used to authenticate remote connections, office productivity software, network connections, confidential data, badge access, and theoretically anything that could use a traditional password. This two-factor user authentication solution became widely adopted as a business solution, with RSA Security holding "72 percent market share in the traditional hardware token market in 2003. This was seven times greater than its nearest competitor" [18]. Despite this success, attackers defeated RSA SecurID in March 2011 by compromising the seeds used to generate tokens [19]. Since the encryption algorithm is public knowledge, if the seeds could be predicted, the system was compromised. Even though the system failed due to attackers, the process as a whole had low usability. It required constant user interaction as

passwords would reset at low intervals and each authentication required the user to manually enter it in. While it did increase the security of a normal PIN by not requiring the user to remember anything and changing so fast that losing the PIN was not a problem, it did not increase usability much because although the user no longer had to memorize anything, they were now burdened by having to carry a new device and enter in an ever-changing number.

Since this failed attempt, numerous startups and other companies including Google and Facebook have experimented with the potential of a physical key [20, 21]. Physical keys are entities that a user constantly keeps on their person such as a bracelet or ring that acts as another way of unlocking their device. In some forms this directly replaces the traditional password, similar to SecurID for two-step authentication [20], but can also used in place of traditional authentication methods as displayed by the NFC Ring in Figure 3-2e [21]. In this case, once the key comes within sufficient distance, the user becomes authenticated and can use their device. This is extremely usable with the user having nothing to remember and requiring little user interaction. It removes numerous traditional vulnerabilities including shoulder surfing, smudge attacks, and social engineering. However, despite the positives, this technique is untested and unproven, the key can be stolen and could be expensive, and it must be kept in the user's possession. It also does not address the continuous authentication aspect of mobile authentication. Even though authentication is fast and easy by simply using NFC to touch and unlock the device, there is still no mechanism to maintain that session without a timeout and re-authentication by touching the device again. In addition, these types of devices may fall victim to antenna amplification attacks depending on the manner they are created. This

vulnerability allows an attacker to boost the range of the key, making the device think it is close enough to the key to be authenticated.

CHAPTER 4: PROPOSED SOLUTION

The proposed solution is to apply principles from another industry, the auto industry, to mobile devices. Car manufacturers have started using smart keys (key fobs) greatly increasing usability and rendering actually inserting the key into the lock unnecessary. In addition, the car is able to detect when the key is within an acceptable range and perform appropriate actions. If this idea could be applied to mobile devices, the goals of this thesis can be achieved including greatly increased usability and secure continuous authentication.

Like a key used to unlock a car, for mobile authentication a smart physical key will be kept with the user to unlock their phone. This key could be in the form of a ring, wristband, keychain or another easily transportable physical form. The key should take on a form that the user is willing to carry or wear and easily integrate into their current lifestyle. For instance, if the user always carries keys, a keychain would be a good form. If not, the key could be in the form of a card if the user always carries a wallet. The ultimate goal of the key is to be in the user's possession whenever they want to use their device, but not feel like a burden that they must track.

4.1 Inspiration:

Traditional physical keys are a concept every person is familiar with. As long as the person is in possession of the key, they are able to gain access to anything that key is able to open. As new technologies were created, electronics needed to emulate this behavior with a similar concept and passwords were born. The touch screens of mobile devices do not provide an easy means for entering secure passwords, resulting in a need for gestures or other more usable methods. However, they suffer from not being able to

provide a continuous kind of authentication to ensure the current user is still valid, instead defaulting to a timeout-authorized period of time. Traditional keys avoid this issue by relocking in the case of a door or requiring the key to remain in the lock as in the case of a vehicle. It is up to the owner of the key to remain in control of the vehicle while they key is in the car (user is authorized). However, in recent years advances have been made to car keys that increase usability.

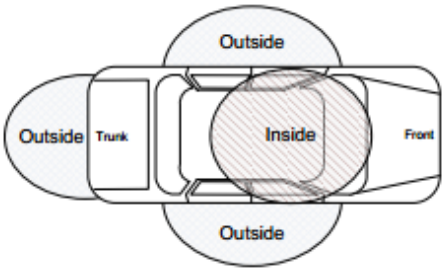
a)  b) 

Figure 4-1: PKES Components a) PKES Key and physical backup key b) Visualization of PKES system sensors

A new technology, Passive Keyless Entry and Start (PKES) systems, "allow users to open and start their cars while having their keys 'in their pockets'" as displayed in Figure 4-1 [22]. This provides the ideal usability to the user with inherent security without user intervention. Using a magnetically coupled radio frequency signal, the car detects the proximity of the key to see if it is within remote distance, directly outside the car, or within the car to perform context aware actions [22]. As displayed in Figure 4-2, the PKES protocol involves performing a challenge-response between the car and the key using the car ID and key response.

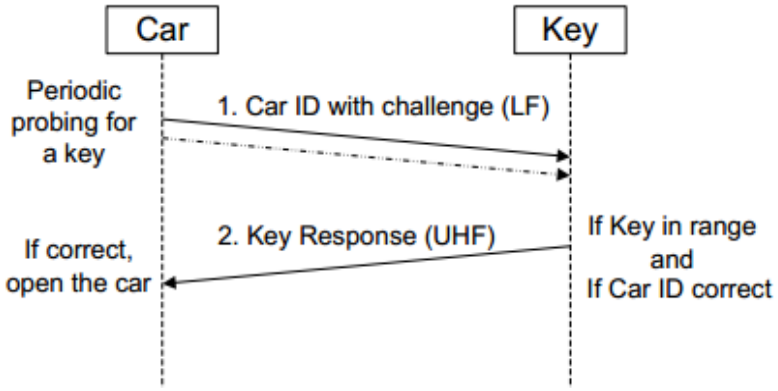


Figure 4-2: Visualization of PKES Protocol

In practice, this works by the owner of the car being able to keep the key in their pocket at all times and instead using the door handle to initiate a lock or unlock request. The antennas on the car's exterior and interior are used to detect if the key is within the sufficient range or inside the vehicle. The PKES system's distances typically allow for 100m to unlock/lock a car while the inside distance is determined by sensors within the car as displayed in Figure 4-1b [22]. If the key is in range, the car can be unlocked when the door handle is pressed. If the key is inside the vehicle, the ignition can be started by a push button and if not, the car will not start. Additional features are for the car to not lock if the key is inside and for the car to shut off if a driver attempts to drive a started car after the key has left the vehicle. On effective devices, the reaction time is fast enough that the car will authenticate the key as the user is performing the initial authentication (pressing the door handle) giving them the feeling that the car is unlocked. If the device is low on power or the battery is completely dead, a backup physical key can be used that is usually located in the device or a transponder is included with the key that does not need a battery. For this system to be effective, the key must be locked so that a customer

cannot reconfigured it to work with another car, otherwise it would be much easier to manipulate the device. In the event of a stolen or lost key, the owner must go to the car manufacturer or dealer for the key to be replaced. While this process is expensive, it provides extra security.

However, this system, despite its strengths, is vulnerable due to trusting its sensors to accurately detect the physical proximity of the key by "assuming that the ability to communicate implies proximity" [22]. This leaves the system prone to relay attacks where someone is able to relay key messages to the car with the intent of appearing as though it is closer to or even inside the vehicle. The attack's main strength is that is does not need to interpret or modify the signal making it "completely transparent to most security protocols designed to provide authentication or secrecy of the messages" [22]. Papers have been published providing possible solutions to this vulnerability. These solutions include various methods to shield or disable the key when not in use so that it cannot be used except when the owner wishes to use it with the car. While effective, none of these solutions are really usable in that they cause the user to have to intervene in the authentication process to protect themselves, which is exactly what this system is trying avoid. However, one solution involves the use of RF Distance Bounding where the verifier measures a lower-bound distance to the prover [22]. This effectively means that the attacker can say they are further than they appear, but are never able to spoof that they are closer.

4.2 Overview:

The basic idea of the proposed physical smart key is for it to reside wherever the user intends to carry it and never need to be moved. When the user wishes to use their

device they would press the normal unlock button (power button for Android and home button for iOS) that is normally used to wake the phone and the key will be detected bypassing the normal lock screen. At this point, the device screen turns on and the user is now “authenticated” meaning they can freely use the device without ever needing to input any information with their session being maintained on an interval separate from user inactivity. An important point is that this solution provides the user with behavior as if their device has no authentication security from their point of view.

One thing to note is the difference between a screen timeout and an authenticated session. The purpose of the screen timeout is to turn off the display after a set period of time to save battery. This will also usually end the active session meaning that the device will need to check with the key again for the device to become unlocked when the unlock button is pressed. The active session is the period of time that the user is able to interact with the device that is unlocked (screen is on). While authenticated, the device will continue to check to make sure the key is there (active continuous authentication) and will end the session if the key cannot be found (locking the device). The user will then need to use an alternate form of authentication (PIN/password). Having the key in range of the device does not turn on the display, but simply allows the user to wake the device and bypass the lock screen that would normally appear.

4.3 System Framework:

The system framework includes five main components: the RFID Manager, Encryption Layer, Policy Manager, Hardware Key, and Device Id as displayed in Figure 4-3. These components are in both the physical smart key and mobile device, but provide different functionality in each.

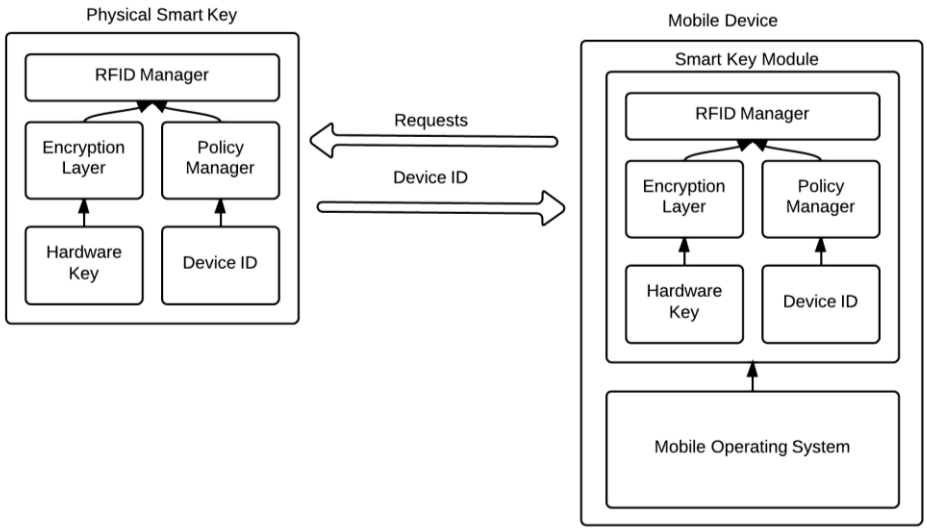


Figure 4-3: Smart Key System Framework

The RFID Manager handles incoming and outgoing RFID transmissions. Encryption Layer decrypts and encrypts these transmissions using symmetric-key encryption with the shared secret Hardware Key so that the messages can either be read or sent. The Policy Manager will take the decoded messages and handle them according to its policy. For the smart key, this includes whether or not the received transmission is valid and contains the correct Device Id so that it can transmit its response. For the device, this validates the returned transmission by checking the smart key's Device Id and unlocking the device. The Hardware Key is the shared secret key used for symmetric-key encryption stored on both the device and physical smart key. The Device Id is the unique identification for the device and smart key used for identification verification.

4.4 Design Goals:

The following are the design goals for the proposal as described in the system framework shown in Figure 4-3. Provide a replacement for traditional mobile authentication methods that:

- Provides security that is not limiting to the user
- Either inherent or completely invisible
- Functionality should act as though there is no security
- Occurs during normal user interaction
- Provides a system for continuous authentication
- Increases usability over current methods without jeopardizing security

Hardware and software standards are currently in place for all major providers, which means the following must be true for it to be a viable solution:

- Easily integrate into existing security architecture
- Easily integrate into current hardware
- Provide backwards-compatible solution for existing devices.

4.5 Protocol:

The authentication event is triggered by pressing the button that is normally used to wake the device. This parallels using the car handle or car key button like in the auto industry implementation. The authentication process travels between the physical smart key and the user's device if the key is within a sufficient distance. From there, the device would be able to actively and continuously authenticate on short intervals that do not correspond to user activity unlike current mobile methods. This is only possible because

the process does not require user intervention. If the key is not found, the device will remain locked or if during an active session would become locked and would require a normal password to unlock. This solution would provide users a secure authentication process, protecting them during initial contact as well as continuing to validate their authorized session. Additionally, it also provides them with a high level of usability that allows a user to focus on the task at hand without consciously having to worry about security.

4.5.1 Overview

The proposed protocol takes ideas from the car key protocol as displayed in Figure 4-4:

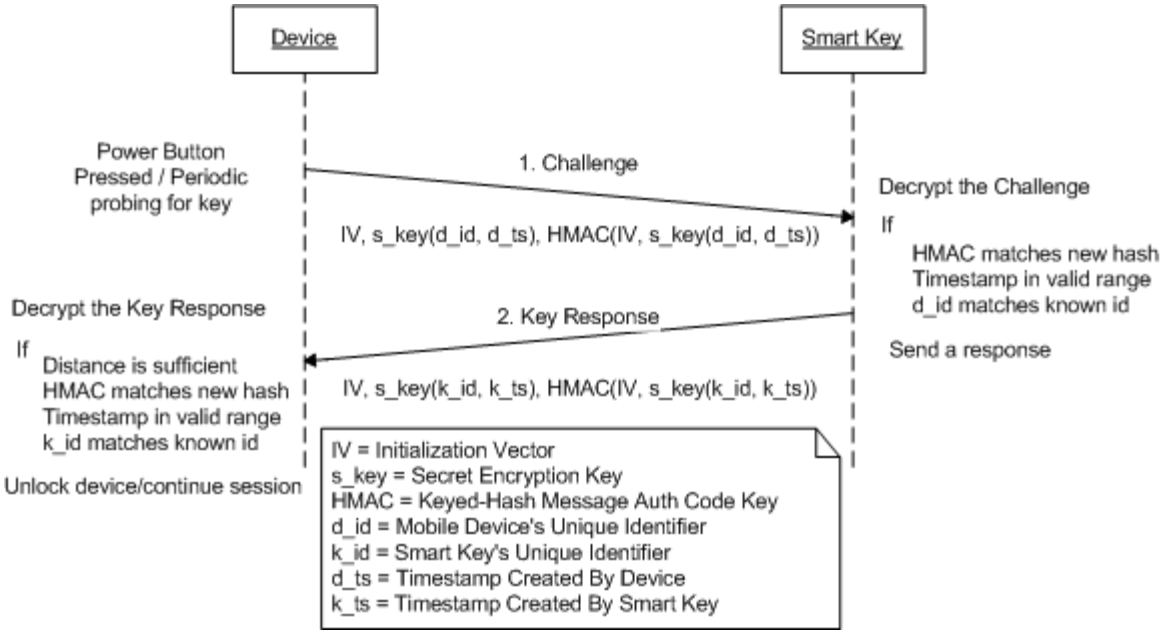


Figure 4-4: Visualization of Proposed Protocol

The mobile device holds its own unique identifier (d_id) as well as its valid smart key unique identifier (k_id). The smart key in turn holds its own unique identifier (k_id) as well as its valid mobile device pair (d_id). Both sides also hold a shared secret key and a

keyed-hash message authentication code (HMAC) key set offline at the time of manufacturing before being sold to the user making symmetric-key encryption possible.

The process itself is called Encrypt-then-Authenticate, the same ordering used in Internet Protocol Security (IPsec), and is generally accepted as the most secure ordering compared to Authenticate-then-Encrypt and Encrypt-and-Authenticate orderings [31, 32]. First, the message is encrypted using AES-128 providing confidentiality and then the encrypted message is hashed using HMAC-SHA-256 to provide integrity and authentication. This is superior to the other orderings because the encrypted data does not need to be changed when verifying its integrity by recomputing the HMAC and comparing it to the one received. This contrasts Authenticate-then-Encrypt where the data must first be decrypted before it can be verified. The authentication is ensured because both sides must know the separate HMAC secret key to verify the signature. RF Distance Bounding is also performed on the mobile device side so to determine the closest distance the key can be and prevent relay attacks [22].

The communication process works by the device first sending out a challenge. This action is prompted whenever a session needs to be established or maintained such as when the user clicks the wakeup button on the device. If a key is within range, it will check the validity of the message using the secret HMAC key, decrypt the cipher text using the secret encryption key, verify that the timestamp is within an acceptable range, and finally check to make sure the device identifier matches the smart key's known device. If everything is correct, the smart key send out its response. The device will first use RF Distance bounding to ensure that the distance has not been spoofed using a rely attack, which will be further explained in the attacks section. If it is passes the check, it

will follow the same verification process as the smart key to verify that the message is acceptable and matches the identifier that the mobile device expects for the smart key.

In the event of a dead or otherwise unusable battery, a password override can be used along with the smart key similar to how Apple's fingerprint authentication or Android's face unlock allows for alternative authentication. If a key is lost, the device must be sent in to the carrier to obtain a new key. In the meantime, the user is able log into the device using their password to disable the physical key unlock so that it cannot be used to unlock their device.

4.5.2 Encryption Evaluation

For this encryption protocol to be successful and secure, it must accomplish certain encryption objectives. These include confidentiality (preventing unauthorized access to the data), integrity (maintaining the valid state of the data), authentication (verifying the identity of the sender), and nonrepudiation (proofing the sender actually sent the message and preventing them from disputing they were the sender) [33]. For this proposal, the use of symmetric-key encryption with AES-128 provides confidentiality and the use of HMAC-SHA-256 provides both integrity and authentication. The use of identifiers also helps with authentication and the use of timestamps provides nonrepudiation. The following describes how the protocol accomplishes each goal.

As previously stated, confidentiality entails preventing an attacker from listening in on transmissions [33]. This involves using some sort of encryption to send cipher text instead of clear text during any sort of communication.

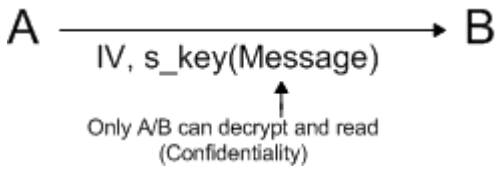


Figure 4-5: Protocol – Confidentiality

As displayed in Figure 4-5, the confidentiality aspect of the protocol involves using a shared secret key that only A and B know to encrypt their messages. These keys are of length 128 to work with AES-128 encryption to provide sufficiently secure two-way encryption between each entity. Since the physical smart key is a low power solution while still requiring fast responsiveness, AES-128 was chosen as it provides sufficient security and performance.

Integrity, which involves the verification of data to determine if it was tampered with, is important so that an attacker is not able to manipulate data that is en route [33].

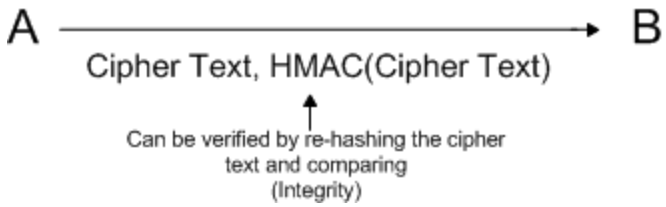


Figure 4-6: Protocol – Integrity

As displayed in Figure 4-6, the receiver gets both the cipher text (encrypted message) and a keyed-hash message authentication code (HMAC). They are then able to re-hash the cipher text and compare it to the received HMAC to see if the transmission has been tampered with. If the hash matches, it means that everything is intact.

Authentication, proving the sender is who they claim to be, is important so that attacks such as the man in the middle attack cannot be performed [33]. This prevents an

attacker from intercepting and sending messages between two parties without them noticing.

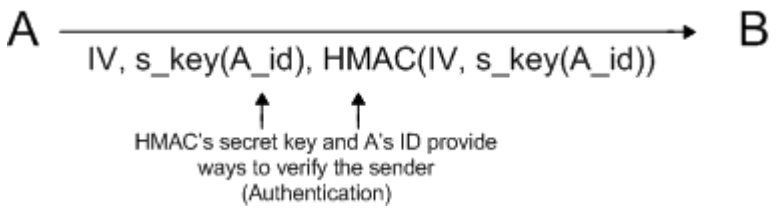


Figure 4-7: Protocol – Authentication

As displayed in Figure 4-7, the message itself contains the unique identifier of the sender. The receiver will then verify that the sender is a known party and will only accept the message if they are recognized. The identifier is protected by the confidentiality and integrity of the message, ensuring that it is not tampered with. Additionally, HMAC authenticates the sender by using another secret key separate from the AES encryption key that only the two parties know. This means that when verifying the cipher text by rehashing using HMAC-SHA-256, both the integrity and authentication goals are completed at the same time.

The final goal is nonrepudiation, which involves verifying that the sender actually sent the transmission and creating evidence so that they cannot deny it later [33]. This is especially important for this proposed protocol because otherwise an attacker would simply need to intercept transmissions going to and from a mobile device and resend the response at the appropriate time.

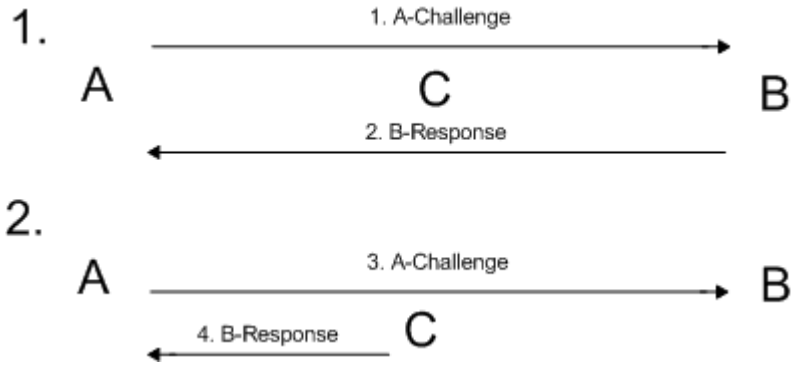


Figure 4-8: Protocol – Repudiation Attack Scenario

Figure 4-8 displays a scenario where the protocol could be exploited without nonrepudiation. In this scenario the attacker (C) first intercepts the challenge (A-Challenge) and the response (B-Response). Notice that B still receives the challenge and sends its response as it does not know anything is wrong. Then, when A sends out a challenge and B is not present, the C would resend B's previous response without manipulating it. Instead of failing, A would receive a valid transmission and would authenticate.

To prevent this vulnerability, nonrepudiation needs to be used so that a simple replay attack like this cannot be performed.

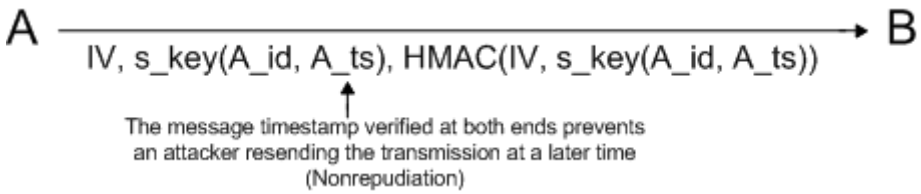


Figure 4-9: Protocol – Nonrepudiation

As displayed in Figure 4-9, by adding a timestamp into the message, the transmission is now time-dependent and cannot be re-sent by the attacker when B is not present. At each end of the transmission, the receiver checks the current time and verifies that the transmission time is within an acceptable range. Since the confidentiality and integrity

goals have been fulfilled, the attacker is not able to modify the transmission time and is unable to perform a relay attack, which is further explained in the attacks section.

4.6 Mobile Device Requirements:

For the proposed solution to function properly, some requirements must be met for the mobile device. Most importantly, like most authentication solutions, it would need operating system permissions to prevent access and respond to system actions such as when the device wakes up or when a phone call is received. This is similar to the traditional "lock screen" that prevents access when a session is not active. In addition, the solution must be allowed to perform actions during normal use to maintain a session for active continuous authentication. This involves sending out an encrypted message to the physical smart key after a set amount of time while the user is using the device. If access cannot be established, the user's session would expire and they would return to the lock screen.

The lock screen for the proposed solution should only appear long enough for communication to be made with the physical smart key. Ideally, this communication would be so fast that the user does not even see it. This means that when the session first starts (device is woken), the authentication process begins and by the time the screen comes on, the user sees their home screen. This process would also occur during their session to provide active authentication during their session. When this occurs, it should not interrupt the user unless the authentications fails, at which point the lock screen will appear and prevent device access.

When authentication fails, the user will be presented with a lock screen with an option for an alternative login mechanism. This option is in case the physical smart key

malfunctions, is lost, or the user does not have it in their possession. The alternative login is in the form of a password with strict length and character requirements. While this is less usable to the user, it is not the intended use of the solution and should be exceptionally secure in case someone attempts to circumvent the physical smart key. In addition, the lock screen should allow for resending the authentication message in case the key becomes available.

For communication purposes, RFID must be available on the device so that communication is possible. This allows for the device to transmit the RFID message and for any smart key to receive it without the need to pair as in the case of Bluetooth. The device must also support AES encryption and a form of HMAC so that the use of symmetric-key encryption with authentication is possible. AES is used when encrypting and decrypting the message and HMAC is used to validate the signature. Additionally, the device must be able to create and compare timestamps so that the transmission can be verified as new and replay attacks cannot be performed.

4.7 Smart Key Requirements:

The requirements for the physical smart key are similar to the mobile device, but instead of the difficulties of integrating into the mobile device operating system, the difficulty is mainly involved in the hardware. Physically, the key must be in a form that is convenient to the user. This means being small enough that it can be easily carried and either wearable or in a form that is easy to keep in the user's possession at all times. For communication, the key must also allow for active RFID transmission that is always listening for the device to make a request. This implies a very low power solution that does not need its source of power changed. However, it must also be able to store

persistent data such as the encryption and HMAC secret keys and a smart key ID as well as perform encryption and other logic-based actions.

The key must be able to use AES encryption, HMAC, and create timestamps similar to mobile device. This can be more difficult with a less powerful device depending on the implementation. Encryption and decryption can be expensive, but the key must be able to perform both actions fast enough that the user does not feel a lag when trying to authenticate their device.

CHAPTER 5: IMPLEMENTATION DETAILS

5.1 Proof of Concept Overview:

With the previous requirements for the physical smart key and mobile device taken into consideration, technology was chosen that can emulate this behavior and is realistic to implement in a research setting. The purpose of this proof of concept is to show the proposed method working and provide of better understanding of the usability and security details. While it will not be an accurate representation of the final product, the prototype will allow for a basic understanding. Any compromises made for the sake of the proof of concept are explained and justified.

5.2 Mobile Device Technology:

For the mobile device, an Android phone was used for the proof of concept due to the openness of the operating system and the freedom of development. The other major operating systems, iOS and Windows Phone, have greater limitations making their use more difficult without any direct benefit. In addition, Android allows for the application launcher to be replaced, providing an avenue for the prototype to replicate replacing the lock screen [23].While it does not directly act as a lock screen, it does provide a way to respond to the "home" button to imitate waking the device.

Due to some limitations with the physical smart key, Bluetooth communication was used instead of RFID for transporting information to and from a device. It provides similar functionality and is more supported by Android libraries. The main drawback of this choice is that the two devices must be paired first for Bluetooth to work and then connected to each other. While it doesn't change the functionality significantly, it can be difficult when going in and out of range. The idea of the devices being "connected" does

not exist in the context of the proposal, but it does provide tighter pairing for the proof of concept. In addition, it is difficult to measure the distance of the devices using Bluetooth, meaning accurate distance bounding cannot be performed. Despite the few changes, if this proposal were to be taken to production and full access to the operating system was provided, reproducing the solution would be trivial on other platforms with different communication methods.

For the Android software, the Bluetooth Chat sample app by Google was used as a baseline for the communication service. In addition, sample code and various Java libraries were used to extend the functionality of the proof of concept. Since the only way to fully emulate a lock screen is to change the operating system itself, the application was created as a launcher so that it could override as much as possible. The sample application was created with the assumption that it would remain open when the device was put to sleep and respond when the device is awoken. In this way it does not directly prevent access because of the limitations of Android to prevent malicious applications, but still provides a sufficient testing environment.

5.3 Smart Key Technology:

Finding an adequate solution for the physical smart key that emulated the desired behavior was the most difficult proof of concept decision. Raspberry Pi devices provide extensive functionality and possibilities as a small Linux machine capable of running Python scripts, connecting to a monitor, and using standard USB devices making it possible to create a powerful prototype [24]. After plugging in an Ethernet cable, downloading the appropriate Bluetooth libraries, and plugging in a Bluetooth USB

Adapter, the device is able to communicate via Bluetooth. This is in addition to a computer monitor and any necessary input devices required to start everything up.

Since the device is run on a Linux distribution, Raspbian, it is able to use standard Python for the smart key service. Since Python is so well supported, first and third-party libraries cover any encryption or Bluetooth needs required to connect to the Android device. This includes PyCrypto, a Python Cryptography Toolkit, which is used for the AES encryption [25]. For the service to run, the Raspberry Pi is booted and the Python script is started. This service waits for a connected Bluetooth device and Android message before confirming the device ID and sending back its own key along with a timestamp.

While this is all that is required for the needs of a functional prototype, additional components including a portable power source and enclosure would be necessary for testing the portability of the solution. If this were to go to production, a completely custom solution would need to be designed and created that confirms to the everyday mobile needs of a customer. This would mean a solution that can be attached to a user's keychain or worn in a way that is not a hindrance.

5.4 Lock Screen Design:

If everything is working properly, the smart key service would be running constantly on the Raspberry Pi and would not need any user intervention. The only component the user would see is the lock screen on the Android device. For the purposes of the prototype, the screen looks like Figure 5-3.

Figure 5-3: Prototype Lock Screen

Starting from the top left, the text below the Smart Key title displays whether or not the service is connected to a smart key. The magnifying glass button to the right allows the user to manually search for a smart key to connect and pair with. The button to the right of that makes the Android device discoverable in case that the physical smart key is unable to discover it. The middle portion of the lock screen displays any communication to or from the Android device. The text box at the bottom of the screen allows for secondary login using a password for when the key is inaccessible. The final two buttons at the bottom are in place in case the device does not connect successfully to the known smart key by default or if the challenge needs to be resent.

The flow of the authentication process on the device works by automatically connecting to a known device when the user wakes up the device via the home button.

Any successful connection will result in the application generating a timestamp and sending its device ID to the physical smart key, which will respond with its own ID and timestamp if the Android device's ID is known. The Android device will then verify that its timestamp and the one sent by the physical smart key are within an acceptable amount of time and the ID matches the Android's known smart keys. If all the components validate correctly, the lock screen will dismiss and allow the user to use the device.

CHAPTER 6: SECURITY ASPECTS AND ATTACKS

For this proposal to be a viable solution, it must address all the security concerns of the previous methods as well as any others that may be introduced by the new technology.

6.1 Physical Key Attacks:

Physical key attacks involve any attacks targeting the proposed solution related to the user requiring them to have the smart key in their possession. This means addressing new vulnerabilities that having the key may have introduced.

6.1.1 Seeding Algorithm Vulnerability

When reviewing related solutions and the history of physical keys, the RSA SecurID solution was one of the biggest and most influential. It failed due to having its seeded algorithm compromised. The proposed solution fixes this problem by instead employing symmetric-key encryption that is set along with hardware IDs before the device and key are given to the user. This cannot be changed at any time after it is given out except by returning it to the mobile provider or manufacturer. This provides the best-known method for securing communication via symmetric encryption because the secret key does not need to be exchanged [26]. Unlike the RSA SecurID method, this does not require any kind of seeding, which can be compromised. The secret keys are implemented at a hardware level making it inaccessible to the rest of the operating system to prevent unauthorized access or manipulation. The RSA solution was also not intended for mobile authentication and required a constantly changing password, which is different from the proposal.

6.1.2 Theft of Key

If the key is lost or stolen, the device owner still has the option of authenticating their device using a normal PIN or password. The device would notify the user that the key could not be found and would prompt them to enter in their credentials. At this point, the user is able to disable smart key access by going into the security settings while in an authenticated session. By doing this, even if the thief was able to get physical access to the device, the smart key would no longer unlock it. In addition, the user can go back to their mobile carrier and request a new key. The carrier or manufacturer will then take the device and register it with a new key. One complaint can be that this might be an expensive process, but as RSA Security stated when they were leading the market, other options including passwords are costly as well if compromised [18]. The consequences of leaving a device unprotected can be extreme compared to the onetime costs of a physical key. In addition, the wasted time and effort of using traditional authentication methods cannot be over-emphasized. The increased usability alone will save countless amounts of time while maintaining tight security of a device that quite literally can hold the information about someone's life.

6.1.3 Social Engineering

Social engineering is an attack involving the psychological manipulation of a victim with the intent to gain something. This can be to gain access to confidential information or even to trick them into performing some sort of action. Relating to authentication security, this usually involves some sort of phishing attack where the victim is tricked into giving their username or password to someone they believe is trustworthy. However, the case with a physical key is rather unique. Instead of being

tricked into divulging information, the victim would be tricked into giving up their phone and smart key. This is a valid vulnerability for this system, but it is in fact harder to perform with the smart key in use. Without having a smart key, the attacker can still do the same thing by asking the user to place a call with the phone. The owner would unlock the device and then the attacker could run with a now authenticated device. With the key in use, the attacker would now need to steal two things. In the end, having the smart key makes this solution just as if not less vulnerable to social engineering as other solutions. The main defense that makes this solution superior is that the owner can simply keep in possession of the key while the other person uses the device. That way, if they try to steal it, they will not be able to get into it because they do not have the key.

6.1.4 Client Attack

A client attack involves the attacker physically accessing the device and using some sort of guessing attack or an exhaustive search attack to try and discover the key to break the encryption [27]. These attacks focus on exploiting low keyspaces in order to brute force the key. This can be done by simply trying every possible combination through brute force guessing or using dictionary attacks when dealing with hashes or even rainbow tables if a salt is not used. For the solution proposed in this work, AES encryption is used for communication, which is the successor to DES and the current standard. Over the years it has proven to be secure as an attack has not been found that can obtain the key with reasonable computation resources in a reasonable time [28].

6.2 Challenge Response Attacks:

Challenge-response (handshake) is the process of establishing communication between two parties usually through a serious of steps [29]. However, since the proposed

solution's protocol uses symmetric-key encryption with the secret key established securely before being shipped, many of the steps are unnecessary. However, this communication can still be vulnerable and the following attacks are addressed.

6.2.1 Relay Attack

For a relay attack, an attacker is able to trick the key and target device into thinking they are closer to one another than they actually are making the device think the user is close enough for authentication when they could in fact be far away. This allows an attacker to use a device even if the user set it down and walked away with their key. The same solution for keys in the auto industry is applicable to the proposed solution in the mobile device setting. As long as the device uses RF Distance Bounded to ensure that the distance between the key and the device is no further than what is reported, there is no way for an attacker to pretend to be closer than they actually are, effectively preventing the attack. This is performed using mathematical equations on the time it takes for the signal to return to the device [30]. With this in place, attacks dealing with the distance of the key and the device are prevented.

6.2.2 Replay Attack

Replay attacks (also known as a playback attack) involve the interception of a response to authenticate with a device. In the context of a physical key, it happens during the challenge-response actions where the device is listening for the key's response [27]. What happens is the attacker intercepts a message during the challenge-response process, and later resends it pretending to be the smart key. This works because the encrypted response is still correct and hasn't been modified. The device just sees a correct response to the challenge and starts a valid session. To prevent this, the proposed protocol uses a

timestamp to monitor a challenge-response session. When the device issues a challenge, it records a timestamp. The smart key's responsibility is then to respond with its ID and a current timestamp encrypted using the secret key. If the transmission was intercepted and then used later, the timestamp would be old and would not be accepted by the device.

6.2.3 Man-in-the-Middle Attack

A Man-in-the-Middle Attack involves the attacker eavesdropping in on communication to and from the intended parties. The attacker will create independent connections between the two victims and trick them into thinking they are talking directly to each other. What actually happens, is the attacker is intercepting all communications, reading it, and sending whatever they want to the receiver. Meanwhile both parties are receiving messages that look like they came from the intended person.

The proposal prevents this by employing symmetric-key encryption with the secret keys set before the devices are shipped. This is more secure than public key encryption because the keys are never sent over the network and can be guaranteed that they were safely exchanged. During the initial handshake and key exchange, there is an opportunity for a MITM attack, but since the keys are transferred offline, the attacker does not get that chance. The mobile device and smart key are able to use this shared secret for all communication without fear that a third party intercepted it.

6.3 Denial of Service Attack:

Denial of service attacks involve overwhelming a system and making it unusable. This can be done by flooding a network with requests or numerous other means. In the context of the proposed solution, it would mean overflowing the device with responses until it ran out of resources and could no longer function. However, due to the simplicity

of the protocol and the fact that the device only listens when it is waiting for a response the attack is much more limited. Unlike most systems, as soon as any valid response is sent, it will authenticate and no longer listen until it needs to refresh the session. The only way to attack would be to flood it with invalid responses while the valid response is on its way. This would be difficult to do because the time it takes to perform a handshake is very small due to only requiring one response to one challenge. Since it is nearly done instantaneously, the attack would need to send out a flood of response before the challenge comes out and after in an attempt to prevent valid communication.

This can also be done targeting the smart key by sending out numerous challenges so that the key is unable to talk to the correct device. However, since the response is directed at any device, if the key does send out a response, the intended device will still be able to receive it.

6.4 Intrusion Detection:

Intrusion detection is used to recognize when the front-line defenses have failed and the device has been compromised [27]. This is a concept that has not been implemented on mobile devices besides the standard screen timeout with a lock screen. However, as previously mentioned, if an attacker were to take the device before the timeout occurred, they would be able to simply keep the device active and use it indefinitely.

With the active authentication proposed in this thesis, the device will be constantly monitoring to ensure that the key is still within range to validate the session. Since this occurs whether or not the user is engaged with the device, it eliminates the vulnerability of the screen timeout with a lock screen system. The physical key has

compromise detection in the form of "observation of physical loss" [27] meaning that once the user notices the key is stolen, they know that the physical key system is compromised and should disable it on the device to prevent the attacker from being able to access it.

6.5 Symmetric-Key Encryption Problems:

When determining the type of encryption to use there are two basic choices: symmetric, which involves a secret key known by both entities, and asymmetric, which uses a key pair of a private and public key. Each has their strengths and weaknesses. For this proposal, symmetric-key encryption was chosen meaning its potential problems must be addressed.

6.5.1 Scalability Issues

A concern with using symmetric-key encryption is that it does not scale well. Each pair must have their own secret key meaning that in order to create a system for 100 entities, 99 keys must be created. The proposal does not have this issue because the relationship between the mobile device and smart key is one to one meaning only one key will be needed for the encryption between the two. This is not counting the additional key required for the keyed-hash message authentication code. Even in the possible future work where one device can be related to multiple keys, the number of keys will be small enough that scalability does not become a concern.

6.5.2 Key Distribution

The primary reason for choosing asymmetric encryption over symmetric-key encryption is so that the keys can be transferred safely between the entities. The public key and private key system provides a way to transfer information between two parties

safely. The problem with symmetric-key encryption is that there is nothing in place to safely transfer keys and if the secret is ever leaked the system falls apart. However, the proposal does not have this problem because the keys are set before the device is sold meaning the keys do not need to be transferred online and are guaranteed to be shared safely as long as the manufacturing process is secure.

CHAPTER 7: EVALUATION

7.1 Evaluation Table:

Results and findings from this research have shown that current mobile techniques do not provide usable, continuous authentication that does not interrupt the user. Instead, authentication involves a lock screen that is presented to the user when their session ends or before it begins. This session, usually corresponding to the screen timeout, stays active while the user interacts with the device. Once interaction stops, a timeout will start. Once this timeout ends, the lock screen will again appear and the user will have to log in again to use their device. Since the user is able to change this duration, this timeout duration is often increased so that the user is inconvenienced less often, which also decreases security. Numerous studies have found that users are more likely to reduce or fully disable security if it hinders their usability. The more security that is inherent and invisible to the user, the more it will be used. To find the optimal solution, the techniques must be evaluated to find one that is usable enough that it is accepted by the users, but also secure enough that they are protected.

However, to validate whether a solution is better than another, a quantifiable measurement is required. Previous work establishing an evaluation framework created the groundwork for comparing various authentication techniques across non-mobile platforms. The framework by Bonneau, Herley, Oorschot, and Stajano established a framework without quantitative measurements on the basis that the importance of each benefit depends on the specific environment [2]. In this instance, the benefit of a specific platform and the threat environment is known allowing us to be able to determine the weight of each metric on a much more accurate level. The work of Schloglhofer and

Sametinger gave some perspective into the evaluation of mobile authentication techniques specifically, including the different hardware components such as the touch screen and the differences in user desires. However, both of these papers only compared at a high level and did not make any quantitative comparisons making the results less conclusive. Their main argument against a numerical measurement was that the platform and target audience were not sufficiently established to allow for specific values and weights. The work of K. Renaud [4] is essential to the analysis in this work because it involves the assigning of specific weights and metrics to categories and benefits. Using all of these papers, an evaluation framework was created to compare each authentication technique to see how the proposal compares to each of the existing techniques.

An evaluation table has been created to display the findings of each authentication technique in a visual manner. The framework involves the use of three categories: security, usability, and deployability. These are then broken down into specific benefits, which describe a specific aspect of the category as it relates to mobile authentication. Each of these benefits are rated on a scale from 0 to 100 with 100 meaning that it fully fulfills the benefit or it is not relevant. The benefits are weighed by importance as they pertain to the category by multiplying them by a number between 0.0 and 1.0. This allows the benefits to add at up to a total of 100 for each category. Then, each category's total is multiplied by its category weight. These weights add up to a total of 100 for all the categories and determine how important the category is for the overall rating of the technique. The final value is achieved by adding up all the weighted category totals.

	Value	x	Weight	=	Total	
Accessible	100		0.35		35.00	
Integrates-With-Current-Devices	65		0.35		22.75	
No-Additional-Hardware	30		0.30		9.00	
					66.75	Deployability

Figure 7-1: Example Visualization of Evaluation Technique

This process is further displayed in Figure 7-1. The figure shows example values for the deployability category, their corresponding weights, and the weighted totals of each of its benefits. The totals are then added up to find the total value of deployability for this authentication technique. This would be performed on each of the other categories in a similar manner.

7.2 Evaluation Criteria:

The evaluation criteria from the table are given in the form of categories of usability, deployability, and security and their corresponding benefits for mobile authentication to provide tangible factors to compare each authentication method. These are inspired by previous evaluation frameworks and converted for mobile authentication [4, 10, 11]. Table 7-1 displays the techniques broken up into traditional and alternative.

		Usability							Deployability			Security							
	Technique	Nothing-to-Carry	Easy-to-Learn	Infrequent-Errors	Easy-to-Change	Requires-No-User-Intervention	Time-Efficient-to-Use	Memorywise-Effortless	Accessible	Integrates-with-Current-Devices	No-Additional-Hardware	Resilient-to-Shoulder-Surfing	Resilient-to-Smudge-Attack	Resilient-to-External-Brute-Force	Resilient-to-Internal-Brute-Force	Resilient-to-Theft	Resilient-to-Social-Engineering	Resilient-to-False-Positives	Active-Continuous-Authentication
Traditional	Gesture																		
	PIN																		
	Password																		
Alternative	Picture Gesture																		
	Fingerprint																		
	2D Face Unlock																		
	3D Face Unlock																		
	RSA SecurID																		
	Electronic Physical Key																		
	Proposed Physical Key																		

★ = fully fulfills (>= 90)
● = fulfills (>= 70)
○ = partially fulfills (>= 40 < 70)
= does not fulfill (< 40)

Table 7-1: Empty Visual Evaluation Table

To visualize the data, symbols are used to describe how each benefit is fulfilled by the techniques. If the benefit is given a value of 90 or greater, it is considered fully fulfilled and given a star. This means that the benefit either does not apply to the technique or that it is completely protected in the case of security or takes full advantage of the benefit in the cases of usability and deployability. A value of greater than or equal to 70 and less than 90 is considered fulfilled and is displayed with a filled in circle. In these cases, the benefit is considered adequate for the mobile authentication technique. When a technique receives a value of greater than or equal to 40 and less than 70 for a benefit, it is considered partially fulfilled and is given an empty circle. This means the

technique has some elements of the benefit, but not enough for it to have the benefit. Any benefit that is given a value of less than 40 is considered not fulfilled and is given nothing in the table. A 0 is given if the technique has no aspect of the benefit and a value up to 40 is there if some negligible benefit.

This form of data representation is useful for an overall view of the techniques and high level characteristics. For instance, at a glance, attributes can be compared to see which ones fulfill certain benefits. However, this does not include weights or a final score of each benefit, category, or technique so the overall best authentication technique cannot be determined. As it stands, the least important benefits appear the same as the most important so a proper comparison cannot be made.

To fully understand Table 7-1, the benefits and the meaning of their values must be described. The following is a breakdown of each category and each benefit:

7.2.1 Usability

The usability category consists of benefits that make the technique easy to use. These are characteristics often most desired by users because it makes their daily lives easier and makes them feel they are not wasting time. These benefits target areas of user frustration such as aspects that are difficult and/or time consuming.

1. Nothing-to-Carry: The technique does not require the user to carry anything additional outside of what they would normally carry when not using the technique. This includes any kind of physical key or entity outside of the device itself for the purpose of authentication. A score is given based on how easy the entity is to carry.
2. Easy-to-Learn: The technique is easy for a user to learn and remember how to use. This is assuming the person has minimal technology knowledge. A score of 90-100 is

given if the technique is either very straightforward and obvious or if the idea is common knowledge such as a password. 70-89 is given if the technique requires some thought at first but is easily remembered once used. A score of 40-69 is given if the concept is somewhat confusing or not natural to the user. They may have to think about how to do it for a while before the idea is learned. Anything less than 40 is given when a concept is confusing and difficult to learn.

3. Infrequent-Errors: When a user is genuinely attempting to use the system, it does not frequently produce errors. This ensures that when a user is trying to authenticate they do not become frustrated because it is not working either because they made a mistake or the system incorrectly rejected correct credentials (false negative). A score of 90-100 is given if it is extremely rare for a user to enter their credentials incorrectly or for the system to give a false negative. 70-89 is given if errors can sometimes occur when the user is authenticating but not often. A score of 40-69 is given if authentication errors are fairly common. Anything less than 40 is given if errors usually occur at some point when a user is trying to authenticate their device when using the technique.
4. Easy-to-Change: The credentials for the technique are easy to change if lost or stolen without issue. Although reasons for changing credentials are usually a security concern, if the process is difficult or time consuming for the user, it is an inconvenience. The scores for this benefit are mainly broken into 90-100 for techniques that are able to be quickly changed and less than 40 for techniques that either cannot be changed or only have a small set of possible changes such as biometrics. 70-89 is given to cases like picture gesture, which can be easily changed

but the change requires some time and thought. Not only must the gesture be changed, the picture can be changed as well. A score of 40-69 is given when the credentials can be changed but the process requires more than a little time.

5. Requires-No-User-Intervention: The technique does not require the user to do anything (such as enter information) for the authentication to succeed. This means the user is able to use the device right after they wake it up as if they do not have any security set. A score of 90-100 is given if the user is not required to do anything outside of waking up their phone for the authentication technique to work. This behavior exactly mimics having no security from the user's point of view. 70-89 is given if the technique requires the user to wake up their device in a different way or perform a simple action when waking up their device. In practice, it still looks like there is no security employed on the device. An example would be the user having to tap the back of their device with an NFC Ring to wake up their device as an alternative method. A score of 40-69 is given for techniques that require enough work by the user that it is beyond simply waking the device up, but they are still not required to do anything. An example would be in the case of the 2D Face Unlock where the user must simply angle the camera to their face when unlocking. It is still a small user interaction, but does not require the user to enter anything. Anything less than 40 is given when the user must enter something to unlock their device.
6. Time-Efficient-to-Use: The technique is fast enough that the user does not feel encumbered by the system. This benefit is difficult for techniques such as PIN or password where the length can vary. It is assumed that a sufficient length is given that meets the requirements of all the basic mobile platforms. A score of 90-100 is given if

the technique is nearly instant and almost unnoticeable to the user. 70-89 is given if the time it takes to authenticate is very minimal. This should be quick enough that the average user does not feel that they are wasting much time unlocking their device. A score of 40-69 is given when the technique takes noticeable time (beyond a second or two) to authenticate. Anything less than 40 is given when the technique takes a decent amount of time to enter such that the average user would complain.

7. Memory-wise-Effortless: The technique does not require the user to remember anything difficult to enter their credentials for the authentication technique. A score of 90-100 is given if there is nothing for the user to remember to use the technique. 70-89 is given if there is little for the user to remember or the credentials are easy to remember (visual credentials are found to be easier than non-visual to remember). A score of 40-69 is given if the technique's memory requirements are fairly significant, but can be expected by the average person to remember. Anything less than 40 requires the user to remember something they might have difficulty remembering and may even have to write down.

7.2.2 Deployability

The deployability category describes the difficulty it would take for the technique to be taken from an idea to the market. This includes manufacturers cost, additional costs to the consumer, difficulty to implement on current devices, and the ability for customers to use the technique. This is significant because even if the technique has perfect security and usability, if it has flaws that make it unfit for sale, it will still never be seen by the public. However, it has a lower weight because if an idea appears that is good enough, the market can change to accommodate it.

1. Accessible: Users that are able to authenticate using passwords are able to use this technique despite any disabilities. This mainly has to do with the user being able to hold the device to perform an action that is different than how it is to enter a password. A score of 90-100 is given if the technique does not require anything different than pressing the keys for the password. 70-89 is given if a slightly different motion is required, but not significantly different than a standard tapping motion. A score of 40-69 is given if a different action is required than touching the screen such as holding the device in a certain way so that face unlock can function. Anything less than 40 requires significantly different motions than tapping the screen for the password.
2. Integrates-With-Current-Devices: The authentication technique is able to integrate into current mobile architectures without significant work. This means that the operating system is set up to handle the technique in the user interface and backend. A score of 90-100 is given to a technique that follows the traditional lock screen model of authentication. The normal password or PIN login is simply replaced by this method without any changes. 70-89 is given to a technique that requires minimal changes to fit the current model such as additional setup or the use of alternative setup in the case of fingerprint. A score of 40-69 is given in the cases where the authentication technique goes outside the model of current devices, but does not require significant work to modify. This is the case with the physical keys that do not use the lock screen and do not require a new model. They simply authenticate themselves and default to the lock screen if it does not work. Anything less than 40

does not fit into the lock model and requires the setup of a whole new model to function properly.

3. No-Additional-Hardware: The technique does not require any additional hardware outside of the device and what was shipped with it. A score of 90-100 is given if the technique requires no additional hardware at all. 70-89 is given if no hardware is given depending on the implementation of the technique and the device. A score of 40-69 is given if the technique requires some hardware changes. Anything less than 40 means the technique requires separate hardware outside of the device such as a physical entity.

7.2.3 Security

The security category involves how the technique protects the device against attacks and overall protects the device from being compromised. In the scenario of mobile devices and focusing on the basic consumer, security and usability are weighed the same. This maintains the mindset that even the most secure authentication technique in the world will not be used if it is not usable enough for people to use it in their daily lives. Unlike the previous categories, a majority of these are similar in how they are scored. Any benefit focused on being resilient from an attack has the following scoring system: 90-100 means that the technique fully protects the device, 70-89 means the device is mostly protected, 40-69 means it provides some protection, and anything less than 40 means it provides little to no protection.

1. Resilient-to-Shoulder-Surfing: The attacker is able to compromise the system after observing the user. This usually involves someone looking over the victims shoulder, as the name implies, to watch the user enter their credentials. Any memory-based

technique is susceptible to this in some way. The longer the credentials, the more difficult it is for shoulder surfing due to the device hiding the characters as they are typed so that they do not appear on the screen.

2. Resilient-to-Smudge-Attack: The attacker is able to compromise the system after seeing screen smudges. This is another vulnerability for a technique that involves entering in credentials. Instead of seeing the user type everything in, the attacker is able to see the smudges of short passwords or gestures and repeat it to gain access to the device. Gestures are particularly susceptible because the path can easily be traced. PIN and password become more difficult because the order is not known and repeated keys make it difficult.
3. Resilient-to-External-Brute-Force: The attacker is able to compromise the system after numerous guesses using the system interface and any throttling it might add. This involves the attacker simply entering in random combinations trying to get into the device. It also takes into account addition delays the device might add for incorrect guesses.
4. Resilient-to-Internal-Brute-Force: The attacker is able to compromise the system by getting into the device and trying to crack the password. This involves the attacker having access to the data inside the device and attacking the authentication system's stored credentials. The strength of this benefit depends on how the credentials are stored and the number of possibilities in the search space.
5. Resilient-to-Theft: If something physical is used for authentication, the attacker does not gain access to the device if it is stolen. This includes physical smart keys and NFC tags. In the case of face unlock, a photo of the person can be taken to spoof the

authentication system. This is also true with certain forms of fingerprint authentication.

6. Resilient-to-Social-Engineering: The attacker is not able to easily trick a user into reveling their credentials. Although the social aspect of social engineering cannot be protected against besides simply informing the user, certain credentials can be shared more easily than others. For instance, it is easier for a user to describe their password when they are tricked into revealing their credentials than describing a swipe gesture. In addition, certain techniques like a face unlock cannot be compromised using social engineering.
7. Resilient-to-False-Positives: The system does not allow for easy false positives by attacker. This means that the system does not mistake incorrect credentials to be accepted as correct. Techniques such as face unlock, which are not precise, have this issue and can be easily spoofed. Any technique that does not have very high precision can be susceptible to false positives.
8. Secure-Continuous-Authentication: The benefit means that the authentication is validated during an active session in addition to initial validation without user intervention. Unlike previous security benefits this is a yes or no answer if the device is protected. If the technique only authenticates once per session until the timeout ends, then it is not protected. If the technique authenticates during a session even when the user is still active after the initial credentials are entered, then the technique fulfills the benefit.

7.3 Results:

Each of the techniques were evaluated using the previously mentioned method and each of their results are included in Appendix A. Each technique has a complete worksheet that includes a table for each category with its benefits. The benefits have their values, weights, and weighted values calculated with the un-weighted total for each category totaled up at the end. The category weights were determined based on the assumption that usability and security were equally as important. Deployability, while also important, does not affect the performance of the technique directly. However, the deployability of a technique can determine whether or not it is ever released to the public due to the cost to manufacture and add to existing products so it must still be taken into consideration. As displayed, the weights for each category are: usability: .4, deployability: .2, and security: .4. At the bottom of the worksheet is a table that includes the total for each category, its weight, and the weighted total. This is then added up to display the overall weighted total for the authentication technique.

		Usability							Deployability			Security							
	Technique	Nothing-to-Carry	Easy-to-Learn	Infrequent-Errors	Easy-to-Change	Requires-No-User-Intervention	Time-Efficient-to-Use	Memorywise-Effortless	Accessible	Integrates-with-Current-Devices	No-Additional-Hardware	Resilient-to-Shoulder-Surfing	Resilient-to-Smudge-Attack	Resilient-to-External-Brute-Force	Resilient-to-Internal-Brute-Force	Resilient-to-Theft	Resilient-to-Social-Engineering	Resilient-to-False-Positives	Active-Continuous-Authentication
Traditional	Gesture	★	★	●	★		●	●	●	★	★			○		★	●	★	
	PIN	★	★	●	★		○	○	★	★	★		○	○	○	★		★	
	Password	★	★	○	★				★	★	★	○	●	●	○	★		★	
Alternative	Picture Gesture	★	●	○	●		●	●	●	●	★				○	★	●	★	
	Fingerprint	★	●	●		○	★	★	★	○	○	★	★	★	●	○	★	●	
	2D Face Unlock	★	●			○	●	★	○	★	★	★	★		●		★		
	3D Face Unlock	★	○					★		●	○	★	★	●	●	●	★	○	
	RSA SecurID		●	●	★			★	★	○		★	★	★	●	○	★	★	
	Electronic Physical Key		★	★		○	★	★	★	○		★	★	★	★		★	★	
	Proposed Physical Key		★	★		★	★	★	★	○		★	★	★	★	○	★	★	★

★= fully fulfills (>= 90)
● = fulfills (>= 70)
○ = partially fulfills (>= 40 < 70)
= does not fulfill (< 40)

Table 7-2: Visual Evaluation Table

Table 7-2 was derived from the values in Appendix A to provide a high level visualization of how each technique fulfills the benefits for each category. This is efficient for simply showing the characteristics of each authentication technique and proving a basic comparison. Each of the technique's strengths and weaknesses can be quickly identified using this table. For instance, trends can be identified by looking at the table at a high level. Alternative techniques, for the most part, are more protected against the security concerns that plague traditional techniques. In addition, the more similar an alternative technique is to a traditional one, the more of the deployability techniques it fulfills. However, since the table uses symbols instead of actual values, the end result is a

collection of estimates, which is effective for an overview, but poor for a statistical comparison. In addition, since the weights are not included in this table, a final evaluation of each technique cannot be performed based on the common consumer target audience.

		Usability	Deployability	Security	Overall
Traditional	Gesture	68.00	91.25	42.00	62.25
	PIN	58.75	100.00	45.25	61.60
	Password	41.50	100.00	63.75	62.10
Alternative	Picture Gesture	66.75	89.50	39.00	60.20
	Fingerprint	82.00	72.75	75.50	77.55
	2D Face Unlock	65.00	86.00	40.75	59.50
	3D Face Unlock	46.00	52.25	70.25	56.95
	RSA SecurID	40.00	56.00	80.25	59.30
	Electronic Physical Key	76.75	56.00	80.00	73.90
	Proposed Physical Key	86.00	56.00	97.00	84.40

Table 7-3: Numerical Results Table

Table 7-3 was created to provide a high level quantitative representation of each technique to summarize the numerical results of the worksheets in Appendix A. The table lists each technique, how they scored in each category, an overall score using the weights for each category, and their relative place compared to each other. Although concise, Table 7-3 provides a clear comparison using numerical data for each technique. It also displays an interesting look into the techniques' strongest categories. The table makes the weights more interesting because it shows how significant they are in determining the best overall technique. If they were changed for different audiences such as a government job that is more security oriented, the results would be much different. The process of analyzing this table takes more time than Table 7-2, but provides a different level of detail despite its smaller size.

7.4 Evaluation:

To fully evaluate the results, both Tables 7-2 and 7-3 along with worksheets in Appendix A must be analyzed. Each has their own strengths and weaknesses in regards to describing the visualizing the data so they all must be looked at together to get a complete picture of the mobile authentication techniques.

7.4.1 Visual Evaluation Table

Table 7-2 provides an efficient resource for performing a high level analysis and comparison of the characteristics of each benefit without getting bogged down by numerical details. The table shows that traditional techniques perform much better than the alternative techniques in deployability. This is expected because deployability relates to the cost to put the technique into production and since all of these techniques are already used, there would be little to no additional cost. The accessible benefit, the ability for someone to use the technique with disabilities if they were able to use a password, was fulfilled by nearly every category except face unlocks and gestures. Face unlocks have an issue because if someone is unable to hold their phone in one hand up to their face, they will not be able to authenticate properly. At the very least the task will be more difficult if they are used to setting the phone down and tapping to perform all of their actions.

Gestures can be another accessibility issue. If a person is unable to swipe back and forth in a precise manner, they will unable to use either of the gesture techniques. While the argument could be made that the person would not be able to use current mobile operating systems, the swipes in most mobile software are simple, requiring less single motion that is not precise. Gesture unlocks involve swiping back and forth

accurately while leaving your finger on the screen, which can be a difficult task for someone with a disability. Picture gesture is even less accessible because of the different types of swipes and the use of a user provided picture, which can cause issues for a person with vision issues.

The remaining benefits for deployability are more straight forward. The techniques involving physical entities fall short in both integrates with current devices and no additional hardware because of their extra cost to manufacture, which translates to the customer as well as integration costs to incorporate into the current devices and operating systems. While this does put techniques that employ physical entities at a disadvantage in terms of deployability, the physical portion is key to their security and usability strengths.

Looking at the security category, the number of fulfills is much lower for traditional when compared to alternative techniques. However, it is not necessarily indicative that traditional methods are less secure. As previously stated, the benefits are simply listed and their weights are not taken into consideration, the primary weakness of this type of table. For instance, traditional gesture appears very insecure compared to 2D Face Unlock. Yet, 2D Face Unlock has a compromising vulnerability that makes it very easy to break even though it is protected against the vulnerabilities that are common for traditional techniques.

In fact, nearly all of the alternative techniques are protected against the attacks that plague traditional. This is due to the fact that all of the traditional techniques focus on input through the touch screen while most of the alternative use a different means to input credentials such as biometrics or a physical key. This negates attacks like the

smudge attack or shoulder surfing that focus on the user entering in their credentials. Since the user isn't entering in information they know, performing an external brute force attack is also difficult for the attacker because they need to generate different inputs, which is not an easy task for non-user generated values. However, traditional techniques do much better in being resilient to theft and resilient to false positives because they use values that the user knows negating a theft attack and values that are exact and not measured, negating a false positive attack that can be used on biometrics.

The proposed physical key is the only technique that is able to provide active, continuous authentication for mobile device. This is a key detail because it provides something fully that every other technique both traditional and alternative receives a zero. The benefit is also key to the security of mobile devices in a world where a user carries their phone on them at all times. How many times have people left their device on a table while they are distracted with something else, opening themselves up to theft and loss of information? The proposed physical key is the only technique that will protect a user in the case where they have an active session running and an attacker gets in contact with the device. The technique also fulfills every other security benefit except for resilient to theft due to its reliance on a physical entity, which can be stolen. However, its impact can be mitigated by revoking the key's access on the device and using traditional lock screen security until a new key can be used.

For the usability category, it would appear that traditional and alternative techniques are fairly similar when it comes to usability. Since many of the techniques have the same number of fulfills benefits, but different ones it is difficult to determine which ones are stronger. The traditional techniques tend to have stronger values in

nothing to carry, easy to learn , infrequent errors, and easy to change due to their current popularity and reliance on memorable values. The alternative techniques tend to measure in the case of biometrics and are difficult to change in nearly all cases. Physical keys can be very difficult to change and biometrics are impossible. Once the number of human parts are exhausted, another unique value cannot be created. The physical keys also come up short in terms of nothing to carry.

However, where the alternative techniques shine is in the requires no user intervention, is time efficient to use, and is memory wise effortless. These are key benefits that the alternative techniques are trying to fulfill because they strongly affect the user. If a technique were to not require the user to do anything, not remember anything, and was nearly instant, it would be received very well in terms of usability. The only two that achieve this feat are the physical keys. In the other tables each method's evaluations become more detailed and telling of their true strengths and weaknesses in terms of security, deployability, and usability due to the addition of weights.

7.4.2 Numerical Results Table

Table 7-3 provides a concise look at the summarized numerical analysis of each authentication technique. By giving the weighted total for each category and the overall weighted value, the strengths of each technique become obvious and easy to compare to one another. Despite not knowing the details behind the values, insight can be taken from this table.

For the usability category, the alternative techniques have a few that score much higher than the rest. Only the traditional gesture technique, which was made specifically for mobile, scores close to the top four alternatives. This makes sense when considering

that both password and PIN were used before the popularity of the mobile device and simply adopted. The resulting techniques do not take full advantage of the mobile hardware such as the touch screen requiring the user to instead use the often error-prone virtual keyboard. Meanwhile, gesture and the other techniques were created with mobile in mind making them much more efficient for the user to use.

PINs perform better than passwords due to the shorter length in most cases, but become less secure as a result. This is due to their smaller search space, which only includes numbers instead of the alphanumerical passwords. The benefits that score the highest in usability tend to be simpler to remember like in the cases of the physical keys and fingerprint, while also being the fastest to enter. Although fingerprint scores high for usability, the other biometrics do not do as well. This is because of the high accuracy and speed of the fingerprint whereas face unlocks require certain lighting conditions and are prone to both false positives and negatives.

Deployability is the easiest category to analyze due to the smaller number of benefits and basic idea. As expected, the existing techniques perform very well the alternatives have some lower scores. The more the technique deviates from the traditional way of entering in credentials from a lock screen the lower it scores. The additional cost of incorporating new aspects into the device and including new hardware make it more difficult to ship into the market. However, as reflected in the overall score, the category's weight is lower than the others (.2 compared to .4) so it have a much lower overall impact.

The security category has overall higher scores with the alternative techniques. This is due to the focus on fixing the flaws of traditional and providing security where the

user is not in charge of using remembered credentials. Attacks like social engineering and smudge attacks are only applicable for the most part to traditional techniques where the credentials can be easily shared and are entered via the touch screen. However, like in the case of 2D face unlock, other vulnerabilities can exist which drop the score significantly.

When analyzing the results displayed in Table 7-3, it becomes apparent that the proposed physical key has a major edge in both usability and security, but scores one of the lowest in deployability. Looking at Appendix A, this is due to the high cost of additional hardware and the fact that additional work must be made to integrate into current devices. However, if the primary objective of mobile manufacturers is to increase the usability and security of their devices, the extra cost of deployability would be worth it. Not only would they gain a major edge over competitors by having a solution that they could tout as more secure than traditional methods, but mobile users would want to use devices that have this security technique available fir usability reasons. Customers would be more secure and waste little to no time with their authentication as it happens without their knowledge in the background.

7.4.3 Appendix A Worksheets

The Appendix A Worksheets provide a detailed breakdown of each authentication technique complete with benefits, categories, and weights. This level of detail is difficult to skim through, but provides insight into the reasons for the strengths and weaknesses of the techniques allowing for them to be accurately compared. In addition, having all of the information in one place makes the process of identifying the improvements of the proposed physical key much easier.

The use of passwords, the most traditional authentication credential, does not hold up as well on mobile devices. Its purpose is to provide the most established technique that users are most familiar with due to their time using traditional computers. Overall, it has a low usability score as it is less time efficient having to type a more complex combination using a virtual keyboard and being more difficult to remember than the credentials of other techniques due to the high possibility of characters. Digging deeper into the usability category, the main problems are found. Despite perfect scores in Nothing-to-Carry, Easy-to-Learn, and Easy-to-Change, the remaining benefits score poorly. The Infrequent-Errors benefit scores low due to the technique's reliance on entering an alphanumeric string using an error-prone virtual keyboard that results in inaccuracy when the user attempts to authenticate. The length of time to use and difficulty of remembering a secure password result in poor scores in Time-Efficient-to-Use and Memorywise-Effortless. Its reliance on the traditional inactivity-based continuous authentication results in a poor score in Requires-No-User-Intervention.

For deployability, passwords receive a perfect score across the board due to being an already established technique requiring no additional knowledge or actions outside of normal keyboard use. As for security, the technique has strengths and weaknesses. Its biggest strength is its defense against someone trying to unlock the device if found. With no component to steal it is scores perfect in Resilient-to-Theft and due to the difficulty and exact nature of passwords scores high in Resilient-to-External-Brute-Force and Resilient-to-False-Positives. Smudge attacks are also difficult to perform with passwords because of the need to determine the order and if keys were pressed more than once. Also, unlike other solutions, since the normal keyboard is used for authentication just like

for other applications, it can easily become covered in fingerprints not related to the password. However, it scores a bit lower in Resilient-to-Shoulder-Surfing since an attacker is able to watch the person enter in the password. To mitigate this, the characters on the screen are usually obfuscated and its length makes it difficult to remember as the user types it in on the virtual keyboard. Once the device is stolen, internal brute force, while difficult is possible. Where this technique's security fails the most is not being Resilient-to-Social-Engineering where a person accidently shares their credentials to the attacker and not providing Active-Continuous-Authentication. Since passwords are commonly words or other sequences that are easy to remember, they can easily be shared via chat or other social avenue where an attacker can pose as someone the user trusts to steal their password.

The use of a PIN was introduced as an alternative to passwords for quicker, more usable solution. When compared to passwords for usability, all of the benefits score the same or better as a result of the technique being so similar, but easier to use. The fact that it only uses numbers and the standard length is four means that it is much easier for the user to remember as opposed to passwords greatly increasing its Memorywise-Effortless and Time-Efficient to use benefits as opposed to using passwords. In addition, its Infrequent-Errors benefit is higher because of the less error-prone numeric keyboard. Its deployable scores perfect scores due to the fact that it is a standard like passwords.

However, despite the increases to usability as opposed to passwords, it scores much lower in security. It scores lower in Resilient-to-Shoulder-Surfing due to being shorter in length and only using numbers so an attacker is able to better identify which keys the user is pressing. It also scores much lower in Resilient-to-Smudge-Attack due to

using a keyboard type that is not used in normal device use meaning the keys would not be covered by fingerprints from normal keyboard use. A PIN also scores lower in Resilient-to-External-Brute-Force and Resilient-to-Internal-Brute-Force do to the smaller number of possible characters and the fact that the length is usually shorter, reducing the time to break it. The other security benefits match passwords. Overall, the use of a PIN sacrifices security to make the solution more usable.

The gesture technique was created to utilize the unique touch screen of mobile devices in the hopes of being more accurate, reliable, and usable than the error-prone virtual keyboard. It scores higher in usability due to being optimized for mobile device use resulting in less time for the user to spend authenticating and less mistakes. Gestures score a perfect 100 for Nothing-to-Carry and Easy-to-Change and nearly perfect for Easy-to-Learn because it is a specific concept for mobile devices, which means some learning. In addition, Infrequent-Errors, Time-Efficient-to-Use, and Memorywise-Effortless are higher due to the gesture being fast, accurate, and easier to remember compared to typing something into a virtual keyboard. Deployability is slightly lower due the Accessible benefit. Swipe actions require more than the simple tap action of the other techniques, which may be more difficult for someone to perform. Precise swipes require more finger control for the elderly or someone with disabilities.

The security results are similar to the PIN. Resilient-to-Shoulder-Surfing and Resilient-to-Smudge-Attack both score worse because the swipe persists on the screen and is easy to follow. Both brute force benefits score nearly the same as PIN due to the small number of possibilities. The main thing that gestures score better on is Resilient-to-

Social-Engineering because a swipe is difficult to describe making is very hard to accidently tell an attacker.

Microsoft's Picture Gesture Authentication provides many of the benefits of normal gesture with the additional benefit of being even easier for a user to remember by associating it with an image. The addition of taps and circles was also an attempt at making the gesture more secure. However, the study by Z. Zhao, G.-J. Ahn, J.-J. Seo, and H. Hu provides results that indicate that instead of increasing security and making the gesture easier to remember for the user, it makes the gesture easier for the attacker to predict due to "hot spots" on the image that the user is prone to use [12]. These spots narrow the search space for the attacker when they attempt to compromise the device. As a result, the security of the technique scored lower than others. The other factors are nearly identical to normal gesture.

Fingerprint scored much higher in both usability and security than the previous techniques. Its primary purpose is to provide a solution that fulfills Memorywise-Effortless, Time-Efficient-to-Use, and Infrequent-Errors so that the technique is as usable as possible without hindering security. By only requiring the user to place their finger on the home button, the process is very fast, does not depend on the virtual keyboard, and only requires them to remember which finger was used. However, it still does require the user to perform an action to authenticate and since it is biometric, cannot be changed beyond the number of fingers of the user. Despite these negatives, the increased speed and ease of use make it much better for usability than the previous techniques.

As for security, the technique is immune to many of the common attacks that plague mobile devices. This results in perfect scores for Resilient-to-Shoulder-Surfing,

Resilient-to-Smudge-Attack, Resilient-to-External-Brute-Force, and Resilient-to-Social-Engineering. However, attacks have been performed to spoof the fingerprint and since the inherent nature of biometrics do not allow a user to change their credentials, it cannot be changed if compromised. No security system is completely safe, meaning the ability to change credentials is a must. This results in a lower score for Resilient-to-Theft and Resilient-to-False-Positives because the attacker is able to steal a fingerprint and use it to compromise the device. An area where this technique struggles compared to the others is deployability. Unlike the previous techniques, this requires additional hardware and needs some effort to integrate into current devices.

2D face unlock is another alternative technique that attempts to be easier for the user by not requiring anything to remember similar to the fingerprint scanner. However, unlike the fingerprint solution, face unlock score low in Infrequent-Errors frustrating users. It requires good lighting and for the user to hold the device up to their face in an awkward position that they would not normally do in public. Even if the conditions are acceptable, the user is still faced with an unacceptable amount of false negatives where the correct user is not able to access their device.

For usability, 2D face unlock receives perfect scores in Nothing-to-Carry, Easy-to-Learn, and Memorywise-Effortless. The concept is easy to learn with the popularity of photography on mobile devices due to today's reliance on social media. It scores well in Time-Efficient-to-Use, but since it scores so poorly in Infrequent-Errors, the resulting experience can be time consuming for the user. When it works, the process is convenient and when it doesn't it is frustrating. Face unlock's other weakness, like fingerprint, is its low score in Easy-to-Change. However, unlike fingerprint with its ability to use different

fingers, this cannot be changed at all besides different lighting and clothing, which should not matter when identifying the person. A strength of the system is that it scores well in Requires-No-User-Intervention due to the action of unlocking it requiring little extra effort. The user simply wakes up the device and faces it towards themselves. Everything else is automatic making this whole process easy. For deployability, this technique's only difficulty comes with the Accessible benefit due to its need to move the camera into position to unlock the device. This is something extra outside of traditional techniques and may be difficult for someone with disabilities. The remaining benefits including Integrates-with-Current-Devices and No-Additional-Hardware both score perfect scores due to front-facing cameras being a common occurrence.

As for security, face unlock nullifies some of the major security concerns of mobile devices, but has a few glaring issues that cause some huge concerns. It fully fulfills Resilient-to-Shoulder-Surfing due to not requiring any entered credentials, Resilient-to-Smudge-Attack due to not requiring the touch screen, and Resilient-to-Social-Engineering due to not having any credentials to share. Face unlock also scores well in Resilient-to-Internal-Brute-Force. However, its flaws are detrimental. Since the solution is prone to false positive attacks the Resilient-to-False-Positives benefit is not fulfilled at all. The attacker is also able to acquire a photo of the victim, which can be used to access the device causing a low score in Resilient-to-Theft. In addition, Resilient-to-External-Brute-Force, the most common and convenient way for attack, scores extremely low because multiple attempts at gaining access can compromise the device.

3D face unlock attempts to solve the security flaws of 2D unlock while maintaining the usability benefits. While it maintains having Nothing-to-Carry and

Memorywise-Effortless fully fulfilled, the additional requirements to use the technique greatly reduces the others. The process of acquiring a 3D image takes additional time reducing the value of Time-Efficient-to-Use, takes time to learn reducing Easy-to-Learn, and is difficult to do consistently resulting in false-negatives reducing Infrequent-Errors. The result is a system that takes additional time and effort meaning it is much less usable.

Its deployability also suffers as it is less Accessible due to requiring multiple angles to use and scores lower in Integrates-with-Current-Devices and No-Additional-Hardware due to the requirements that current devices do not fulfill. For security, 3D face unlock fixes the problems that plagued 2D face unlock. The additional data required for a 3D image makes it much more Resilient-to-External-Brute-Force and makes it less susceptible to the use of a photo raising its value in Resilient-to-Theft. The additional data and requirements cause Resilient-to-False-Positives to score higher as well.

The remaining techniques involve the use of physical keys for the purpose of providing usability similar to biometrics where nothing has to be remembered and the strong credentials of a password. RSA SecurID, an authentication technique that became a standard for many years in business environments for desktop environments, focused on the security element and as a result, suffered in usability. It uses a physical device that generates security tokens for the person to use as their credentials. While it scores high in Easy-to-Learn, Infrequent-Errors, Memorywise-Effortless, and Easy-to-Change, it takes a lot of time to use. This technique is the best example of one that has nearly everything in a category, but does not fulfill the benefits with the most weight. Since time is the most important aspect of usability, Time-Efficient-to-Use is weighted the highest and RSA SecurID scores poorly. It also does not fulfill Nothing-to-Carry or Requires-No-User-

Intervention because the user must carry the device on their person and enter a new generated password for their sessions.

For deployability, the technique scores low for benefits that other techniques fulfill. It requires some effort to be compatible with mobile devices lowering the value of Integrates-with-Current-Devices and the necessity of the physical device mean that it does not fulfill No-Additional-Hardware. However, this technique performs extremely well in security covering the benefits that plague the other techniques. It fully fulfills every benefit besides Resilient-to-Internal-Brute-Force because the device can still be compromised at an internal level, Resilient-to-Theft because the key can be stolen with some difficulty during authentication, and Active-Continuous-Authentication because it does not continue to authenticate during an active session.

Electronic physical keys are a fairly new concept with no current accepted solution. For usability, it provides the benefits of the RSA SecurID with improvements in everything except Easy-to-Change, which it has difficulty with due to being a physical entity that does not use a credential generator. It comes in various forms depending on the implementation such as a ring mitigating the negatives of Nothing-to-Carry. In its best form it requires little action on the part of the user; only that the user perform a simple action with the key resulting in high scores in Easy-to-Learn, Infrequent-Errors, Time-Efficient-to-Use, and Memorywise-Effortless. Its deployability benefit scores remain the same as RSA SecurID due to the additional hardware requirements. For security, the electronic physical key scores higher in Resilient-to-Internal-Brute-Force, but lower in Resilient-to-Theft as there are no countermeasures if stolen.

The final remaining technique is the physical smart key proposed in this research. The primary purpose is to take the electronic physical key, fix all of its issues, and provide active-continuous authentication. The technique scores slightly higher in Nothing-to-Carry due to coming in a form that is easy for the user to carry and not requiring the user to take it out to use and in Easy-to-Change due to the processes in place for disabling and changing physical keys. The main benefits that were improved were Requires-No-User-Intervention and Time-Efficient-to-Use because the process of authenticating should happen automatically without the user even being aware. By simply having the key on their person, the user is able to use their device freely. Like the previous physical keys, the proposed key scores the same in the deployability category due to its need to integrate into the current architecture and requiring additional hardware to function. As for security, the proposed solution fulfills nearly every benefit. The only benefit the technique does not fully fulfill is Resilient-to-Theft due to the technique being potentially compromised if stolen. However, there are some mitigations in place in case of theft of the key. The user is able to disable key access and use a secure password as well as change the valid key.

Overall, the proposed technique provides the highest scores in both usability and security. Key characteristics include its secure and quick method of sending its credentials, ease to learn with no memory requirements, security components that protect against the major vulnerabilities of mobile devices, and its unique active continuous authentication. The main issue that sets it apart from the other techniques is its low deployability score. While this difficulty makes the technique more difficult to implement

for manufacturers, the fact that it protects users in a way that is usable enough that they will continue using it is an accomplishment that will lead to much more secure devices.

CHAPTER 8: DISCUSSION

8.1 Weaknesses:

Despite the strengths of the proposal, there are some weaknesses. A weakness of the proposed physical smart key is its key management. The solution relies on the user to maintain control of the key exposing it as well as the mobile device to potential loss. If kept together, the combination of using the proposed physical key and mobile device provides no real security because the attacker would simply steal them both, rendering the solution meaningless. In its current form, the solution requires that both entities be kept separate so that the theft of one does not compromise the solution. This weakness highlights the need for a usable form factor for the key that the user will be comfortable carrying.

The proposed physical smart key's reliance on the manufacturer for setting and maintaining the secret encryption keys for the solution also presents a concern. By only having one entity that is able to set the keys, all the responsibility for keeping those keys safe lies with that entity. If it is compromised, the whole system falls apart. Instead, by having a situation where multiple entities are able to pair the physical key and mobile device, this risk is mitigated.

In addition to the proposed mobile authentication technique, the evaluation methodology has some weaknesses. Despite the research that went into determining the various possible benefits, they are not necessarily all-encompassing. Security, usability, and deployability are broad concepts and difficult to fully cover. With the limited number of benefits chosen, some aspects of these categories are not accounted for leading to different results. For this to be more accurate, not only must the topics be more fully

researched, but user studies must be performed using both subjects knowledgeable in the field and common users of mobile devices. Only through these studies with actual users will a majority of the benefits be found.

The evaluation methodology also falls short in its use of weights. Not only is the assumption made that security and usability should be weighted the same with deployability being weighted low in comparison, but each of the benefits are weighted based only off of research into related work. These would drastically benefit from in depth user studies into what people find most frustrating and easy to use for their various authentication techniques. This also applies to security so that the benefits most relevant to a user can be weighted higher. For instance, if a security vulnerability is present for an authentication technique, but is rarely targeted, it should be weighted less than a vulnerability that is commonly used to compromise a device. User studies provide a way to more accurately determine these benefits and weights so that the techniques can be more appropriately evaluated.

8.2 Future Work:

For this proposed solution to be viable, additional work will be necessary. One of the biggest areas that will require additional work is the need for user studies as stated previously. In its current form, the evaluation methodology hinges on these benefits and weights that were determined by using related research. However, to find exact numbers personal opinion plays a factor despite all precautions to ensure everything is supported by findings. User studies take more of the opinion out of the equation by providing first-hand findings by people that are using mobile devices every day. Even though the specific preferences and needs will vary from person to person, the result will be benefits

that better encompass each category and weights that better reflect the needs of a majority of consumers.

Another potential area of future work includes exploring the idea of targeting specific user needs and desires for their mobile authentication. In its current form, the weights are set at hard values based on the assumption that the goal of the proposal was to deliver the solution to the average mobile device user. However, everyone's needs are different and so are their priorities when it comes to their devices. The weights can be tailored to these various personas to more accurately fulfill those needs. By creating an evaluation framework that can be changed based on a "target persona", these varying needs can be reflected. Personas could include those focused on their security such as government workers and technology companies or even those that simply want a usable device such as children. Even if some personas are missed it is sufficient as long as everyone can fit into one that is included. This can be as simple as increasing usability so that children are able to use it or security so that it is safe for confidential government information.

Future work for the proposed physical smart key includes the notion of a federated ID. Authentication conflicts can occur when multiple devices and keys are in use. In this scenario, the keys and devices will need to understand how to properly handle multiple messages from various keys and understand whether or not to authenticate a session when there is one valid and one invalid key in the area. In addition, conflicts will occur when an owner is using multiple devices and keeping multiple keys on their person. A single federated ID would simplify these concerns by aggregating the keys into one ID.

Another scenario involves the device owner wishing to allow someone else to use the device for a period of time. In the current implementation, the owner would need to stay by that person or give them the key in addition to the device, compromising their security. However, another method to allow temporary access to that person without the need to give the key would be beneficial. In one possible solution, the device owner would be able to set the device in an extended authentication mode for an amount of time so that the person could use the device freely. There would be a maximum duration so that the person could not give a period of time that compromises security.

One portion of the research that would need to be expanded in the future would be the hardware and method of communication. Bluetooth, while providing everything necessary for the research, requires pairing and communicates only when connected. A better solution would be to use technology such as active RFID, which would provide a much better communication model for this proposal by not requiring one to one connections, and allowing the device to "broadcast" its challenge for anyone to hear and respond to. However, for this research, the additional overhead was too great to justify. Bluetooth provides additional security with its pairing process that would need to be taken into account when using a custom solution. In addition, the hardware and support are both much more available when using an Android device and Python scripting on the Raspberry Pi with Bluetooth than the additional parts and software required to create a new solution using active RFID. Yet, the main hardware component that needs future work is the overall size and design of the physical key. The Raspberry Pi works fine for research purposes into the technology, but to make user studies possible, the key must be portable enough to emulate the idea of the proposal. In its current form, a normal user

would never use it in production due to its size. If a smaller form factor is created, research into the proposal's usefulness will yield better results and accurate user studies can be performed.

A potential fix for the remaining vulnerabilities on the evaluation table is to integrate the physical smart key into the human body in a biometric-like solution. Embedding the key would solve all the issues involved in theft of the key and the usability of keeping it with a user at all times. It would also mitigate the biometrics problems because the password would be exact unlike trying to read imprecise data from something like a retina or fingerprint, which can yield false negatives or even false positives. These systems can cause user frustration if authentication does not work the first time or even becomes compromised by an attacker. In addition, the password can be changed if compromised unlike biometric solutions. This embedded solution, while seemingly all-encompassing, is outside the scope of this research and can be explored in the future.

CHAPTER 9: CONCLUSION

9.1 Contributions:

Several contributions were made as a result of this research. It establishes the necessity of mobile authentication that is both secure and usable. Current techniques are lacking in usability resulting in low usage by consumers. However, a solution that is completely usable, but is lacking in security would be pointless because it would not provide the protect that the user needs. This work demonstrates the usability and security concerns of timeout-based mobile authentication. Not only is the initial lock-screen authentication unusable as it interrupts the user and takes a significant amount of time, but the window between when the user becomes inactive and the authentication times out is a security hole. If the attacker were to gain access to the device in this period of time, they would be able to use the device indefinitely until they were inactive.

This research establishes a mobile authentication evaluation methodology for comparing authentication techniques. Through the use of benefits and weights, techniques are evaluated in the areas of security, usability, and deployability to determine their strengths and weaknesses. The work proposes a solution that provides active continuous authentication that does not interrupt the user and compares it to existing industry-standard techniques. It also demonstrates the feasibility of the solution through a working proof of concept.

9.2 Summary:

A rigorous survey was performed to determine the challenges of current mobile authentication techniques. As evident by previous user studies, the common person does not value security over their usability. Since the current mobile authentication techniques

do not provide the usability necessary for a significant amount of users to want to secure their devices, a new technique must be provided. Goals of this technique include: provide seamless and inherent security that is does not interrupt the user, use continuous authentication that validates the user even during an active session instead of just after a length of inactivity, and integrate in current hardware and software.

All of these goals are fulfilled by the proposed physical smart key, which based on the tables and worksheets, solves the usability and security problems of current techniques while introducing minimal issues. Where it falls short is its reliance on hardware leading to additional costs and consequences when stolen. However, these are mitigated by allowing the user to disable smart key unlock at will so that it is no longer a security concern if an attacker steals it. This solution is implemented in a proof of concept which is validated through numerous experiments to demonstrate its effectiveness. The mobile authentication evaluation methodology proposed in this research provides a framework for comparing the strengths and weaknesses of authentication techniques through the use of various tables and worksheets. This methodology is able to determine the optimal technique for the provided benefits and weights making it a valuable tool for evaluating current techniques as well as anything that might surface in the future.

REFERENCES

1. Kuo, Cynthia; Asokan, N. “Distributed Computing and Internet Technology: Usable Mobile Security.” Springer Berlin Heidelberg, 2012. 1-6.

2. Bonneau, J.; Herley, C.; van Oorschot, P.C.; Stajano, F. "The Quest to Replace Passwords” *Security and Privacy (SP),* 2012 IEEE Symposium. pp.553,567, 20-23 May 2012.

3. Schlöglhofer, R.; Sametinger, J. “Secure and Usable Authentication on Mobile Devices”, *MoMM2012 - 10th International Conference on Advances in Mobile Computing & Multimedia*, Bali, Indonesia, 3-5 December, 2012.

4. Renaud, K. “Quantification of authentication mechanisms: a usability perspective,” J. Web Eng., vol. 3, no. 2, pp. 95–123, 2004.

5. Niinuma, Koichiro; Jain, Anil K. “Continuous User Authentication Using Temporal Information.” Michigan State University, 2009.

6. Feng, Tao; Liu, Ziyi; Carbunar, Bogdan; Boumber, Dainis; Shi, Weidong. “Continuous Remote Mobile Identity Management Using Biometric Integrated Touch-Display.” *IEEE/ACM 45th International Symposium on Microarchitecture Workshops,* 2012.

7. Schechter, S. E.; Dhamija, R.; Ozment, A.; Fischer, I. “The Emperor's New Security Indicators,” in SP '07: *Proceedings of the 2007 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 51-65.

8. Aviv, Adam J.; Gibson, Katherine; Mossop, Evan; Blaze, Matt; Smith, Jonathan M. “Smudge Attacks on Smartphone Touch Screens”, in *Proc. 4th USENIX WOOT*, Aug. 9, 2010, pp. 1–7.

9. Engler, Justin; Vines, Paul. “Electromechanical PIN Cracking Implementation and Practicality.” *iSEC Partners, Inc*. July 2013.

10. Tahiri, Soufiane. “Android Forensics: Cracking the Pattern Lock Protection.” *InfoSec Institute.* Aug 19 2013. <http://resources.infosecinstitute.com/android-forensics-cracking-the-pattern-lock-protection/>.

11. Spreitzenbarth, Michael. “Cracking PIN and Password Locks on Android.” *Forensic Blog*. Feb 28 2012. <http://forensics.spreitzenbarth.de/2012/02/28/cracking-pin-and-password-locks-on-android/>.

12. Zhao, Z.; Ahn, G.-J.; Seo, J.-J.; Hu, H. “On the Security of Picture Gesture Authentication,” *in Proceedings of the 22nd USENIX Security Symposium*, 2013.

13. “iPhone5s Design.” *Apple*. Apple Inc. 2014. <http://www.apple.com/iphone-5s/design/>.

14. Branscombe, Mary. "Why the iPhone's Fingerprint Sensor is Better Than the Ones On Older Laptops." *Cite World*. IDG Enterprise. Sept 11 2013. <http://www.citeworld.com/security/22399/iphone-fingerprint-scanner-better-biometrics>.

15. Findling, R. D.; Mayrhofer, R. “Towards Face Unlock: on the Difficulty of Reliably Detecting Faces on Mobile Phones.” in *Eric Pardede & David Taniar,* ed., 'MoMM', ACM, 2012. pp. 275-280.

16. Findling, R. D.; Mayrhofer, R. “Towards Secure Personal Device Unlock Using Stereo Camera Pan Shots.” in *R. Mayrhofer and C. Holzmann,* editors, Second International Workshop on Mobile Computing Platforms and Technologies (MCPT 2013), 2013.

17. "RSA SecurID(R) Solution Named Best Third-Party Authentication Device by Windows IT ." *PR Newswire.* PR Newswire Association LLC, 16 Sept. 2004.

18. "RSA SecurID - Quick Reference Sales Guide." RSA Security Inc, 2003.

19. Bright, P. “RSA Finally Comes Clean: SecurID is Compromised,” Jun. 2011, <http://arstechnica.com/security/news/2011/06/ rsa-finally-comes-clean-securid-is-compromised.ars>.

20. "The Magic Key: Google and Facebook Planning on Improving Security with Physical Tokens." SpiderOak.com, 16 Jan. 2014. <https://spideroak.com/ privacypost/cloud-security/the-magic-key-google-and-facebook-planning-on-improving-security-with-physical-tokens/>.

21. McLear, John. "NFC Ring." Kickstarter, Inc. 2014. <https://www.kickstarter.com/projects/mclear/nfc-ring>.

22. Francillon, Aurélien; Danev, Boris; Capkun, Srdjan. “Relay attacks on passive keyless entry and start systems in modern cars.” Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.

23. “Intents and Intent Filters.” Android. 2014. <http://developer.android.com/guide/components/intents-filters.html>.

24. “What is a Raspberry Pi.” Raspberry Pi Foundation. 2014. <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>.

25. Dwayne Litzenberger. DLitz.net. Oct 2013. <https://www.dlitz.net/software/pycrypto/>.

26. "Description of Symmetric and Asymmetric Encryption." Microsoft. 2014. <http://support.microsoft.com/kb/246071>.

27. L. O'Gorman. "Comparing passwords, tokens, and biometrics for user authentication." *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2019–2040, December 2003.

28. Toli, A. Zanoni. "An algebraic interpretation of AES-128." in: H. Dobbertin, V. Rijmen, A. Sowa (Eds.), AES Conference, *Lecture Notes in Computer Science*, vol. 3373, Springer (2004), pp. 84–97.

29. W. Mao. "Modern Cryptography Theory and Practice." Prentice Hall PTR, 2004.

30. K. B. Rasmussen and S. Capkun. "Realization of RF Distance Bounding." in *Proc. of the 19th USENIX Security Symposium*, 2010.

31. Bellare, M.; Namprempre, C. "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," *Advances in Cryptology–ASIACRYPT*, 2000. Lecture Notes in Computer Science, vol. 1976, 2000, pp. 531-545.

32. Krawczyk, Hugo. "The Order of Encryption and Authentication for Protecting Communications," Advances in Cryptology–CRYPTO, 2001. Lecture Notes in Computer Science, vol. 2139, 2001, pp. 310-331.

33. Busta, Bruce. "Encryption in Theory and Practice," *The CPA Journal*. NYSSCPA.org. 2002.

APPENDIX A

EVALUATION FRAMEWORK RESULTS

Authentication Technique: Password

Overall: 62.10

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	100		0.05		5.00
Infrequent-Errors	50		0.15		7.50
Easy-to-Change	100		0.05		5.00
Requires-No-User-Intervention	0		0.15		0.00
Time-Efficient-to-Use	20		0.30		6.00
Memorywise-Effortless	20		0.15		3.00
					41.50

Deployability					
	Value	X	Weight	=	Total
Accessible	100		0.35		35.00
Integrates-with-Current-Devices	100		0.35		35.00
No-Additional-Hardware	100		0.30		30.00
					100.00

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	65		0.10		6.50
Resilient-to-Smudge-Attack	70		0.10		7.00
Resilient-to-External-Brute-Force	85		0.25		21.25
Resilient-to-Internal-Brute-Force	60		0.15		9.00
Resilient-to-Theft	100		0.05		5.00
Resilient-to-Social-Engineering	0		0.05		0.00
Resilient-to-False-Positives	100		0.15		15.00
Active-Continuous-Authentication	0		0.15		0.00
					63.75

Overall					
	Value	X	Weight	=	Total
Usability	41.50		0.40		16.60
Deployability	100.00		0.20		20.00
Security	63.75		0.40		25.50
					62.10

Authentication Technique: PIN

Overall: 61.60

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	100		0.05		5.00
Infrequent-Errors	70		0.15		10.50
Easy-to-Change	100		0.05		5.00
Requires-No-User-Intervention	0		0.15		0.00
Time-Efficient-to-Use	55		0.30		16.50
Memorywise-Effortless	45		0.15		6.75
					58.75

Deployability					
	Value	X	Weight	=	Total
Accessible	100		0.35		35.00
Integrates-with-Current-Devices	100		0.35		35.00
No-Additional-Hardware	100		0.30		30.00
					100.00

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	30		0.10		3.00
Resilient-to-Smudge-Attack	45		0.10		4.50
Resilient-to-External-Brute-Force	65		0.25		16.25
Resilient-to-Internal-Brute-Force	10		0.15		1.50
Resilient-to-Theft	100		0.05		5.00
Resilient-to-Social-Engineering	0		0.05		0.00
Resilient-to-False-Positives	100		0.15		15.00
Active-Continuous-Authentication	0		0.15		0.00
					45.25

Overall					
	Value	X	Weight	=	Total
Usability	58.75		0.40		23.50
Deployability	100.00		0.20		20.00
Security	45.25		0.40		18.10
					61.60

Authentication Technique: Gesture

Overal: 62.25

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	90		0.05		4.50
Infrequent-Errors	80		0.15		12.00
Easy-to-Change	100		0.05		5.00
Requires-No-User-Intervention	0		0.15		0.00
Time-Efficient-to-Use	70		0.30		21.00
Memorywise-Effortless	70		0.15		10.50
					68.00

Deployability					
	Value	X	Weight	=	Total
Accessible	75		0.35		26.25
Integrates-with-Current-Devices	100		0.35		35.00
No-Additional-Hardware	100		0.30		30.00
					91.25

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	20		0.10		2.00
Resilient-to-Smudge-Attack	0		0.10		0.00
Resilient-to-External-Brute-Force	55		0.25		13.75
Resilient-to-Internal-Brute-Force	15		0.15		2.25
Resilient-to-Theft	100		0.05		5.00
Resilient-to-Social-Engineering	80		0.05		4.00
Resilient-to-False-Positives	100		0.15		15.00
Active-Continuous-Authentication	0		0.15		0.00
					42.00

Overall					
	Value	X	Weight	=	Total
Usability	68.00		0.40		27.20
Deployability	91.25		0.20		18.25
Security	42.00		0.40		16.80
					62.25

Authentication Technique: Picture Gesture

Overall: 60.20

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	80		0.05		4.00
Infrequent-Errors	65		0.15		9.75
Easy-to-Change	85		0.05		4.25
Requires-No-User-Intervention	0		0.15		0.00
Time-Efficient-to-Use	70		0.30		21.00
Memorywise-Effortless	85		0.15		12.75
					66.75

Deployability					
	Value	X	Weight	=	Total
Accessible	85		0.35		29.75
Integrates-with-Current-Devices	85		0.35		29.75
No-Additional-Hardware	100		0.30		30.00
					89.50

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	20		0.10		2.00
Resilient-to-Smudge-Attack	10		0.10		1.00
Resilient-to-External-Brute-Force	20		0.25		5.00
Resilient-to-Internal-Brute-Force	50		0.15		7.50
Resilient-to-Theft	100		0.05		5.00
Resilient-to-Social-Engineering	85		0.05		4.25
Resilient-to-False-Positives	95		0.15		14.25
Active-Continuous-Authentication	0		0.15		0.00
					39.00

Overall					
	Value	X	Weight	=	Total
Usability	66.75		0.40		26.70
Deployability	89.50		0.20		17.90
Security	39.00		0.40		15.60
					60.20

Authentication Technique: Fingerprint

Overall: 77.55

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	80		0.05		4.00
Infrequent-Errors	85		0.15		12.75
Easy-to-Change	15		0.05		0.75
Requires-No-User-Intervention	40		0.15		6.00
Time-Efficient-to-Use	95		0.30		28.50
Memorywise-Effortless	100		0.15		15.00
					82.00

Deployability					
	Value	X	Weight	=	Total
Accessible	100		0.35		35.00
Integrates-with-Current-Devices	65		0.35		22.75
No-Additional-Hardware	50		0.30		15.00
					72.75

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	100		0.10		10.00
Resilient-to-Smudge-Attack	100		0.10		10.00
Resilient-to-External-Brute-Force	100		0.25		25.00
Resilient-to-Internal-Brute-Force	75		0.15		11.25
Resilient-to-Theft	60		0.05		3.00
Resilient-to-Social-Engineering	100		0.05		5.00
Resilient-to-False-Positives	75		0.15		11.25
Active-Continuous-Authentication	0		0.15		0.00
					75.50

Overall					
	Value	X	Weight	=	Total
Usability	82.00		0.40		32.80
Deployability	72.75		0.20		14.55
Security	75.50		0.40		30.20
					77.55

Authentication Technique: 2D Face Unlock

Overall: 59.50

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	100		0.05		5.00
Infrequent-Errors	20		0.15		3.00
Easy-to-Change	0		0.05		0.00
Requires-No-User-Intervention	40		0.15		6.00
Time-Efficient-to-Use	70		0.30		21.00
Memorywise-Effortless	100		0.15		15.00
					65.00

Deployability					
	Value	X	Weight	=	Total
Accessible	60		0.35		21.00
Integrates-with-Current-Devices	100		0.35		35.00
No-Additional-Hardware	100		0.30		30.00
					86.00

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	100		0.10		10.00
Resilient-to-Smudge-Attack	100		0.10		10.00
Resilient-to-External-Brute-Force	10		0.25		2.50
Resilient-to-Internal-Brute-Force	85		0.15		12.75
Resilient-to-Theft	10		0.05		0.50
Resilient-to-Social-Engineering	100		0.05		5.00
Resilient-to-False-Positives	0		0.15		0.00
Active-Continuous-Authentication	0		0.15		0.00
					40.75

Overall					
	Value	X	Weight	=	Total
Usability	65.00		0.40		26.00
Deployability	86.00		0.20		17.20
Security	40.75		0.40		16.30
					59.50

Authentication Technique: 3D Face Unlock

Overall: 56.95

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	100		0.15		15.00
Easy-to-Learn	50		0.05		2.50
Infrequent-Errors	50		0.15		7.50
Easy-to-Change	0		0.05		0.00
Requires-No-User-Intervention	0		0.15		0.00
Time-Efficient-to-Use	20		0.30		6.00
Memorywise-Effortless	100		0.15		15.00
					46.00

Deployability					
	Value	X	Weight	=	Total
Accessible	35		0.35		12.25
Integrates-with-Current-Devices	80		0.35		28.00
No-Additional-Hardware	40		0.30		12.00
					52.25

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	100		0.10		10.00
Resilient-to-Smudge-Attack	100		0.10		10.00
Resilient-to-External-Brute-Force	85		0.25		21.25
Resilient-to-Internal-Brute-Force	85		0.15		12.75
Resilient-to-Theft	75		0.05		3.75
Resilient-to-Social-Engineering	100		0.05		5.00
Resilient-to-False-Positives	50		0.15		7.50
Active-Continuous-Authentication	0		0.15		0.00
					70.25

Overall					
	Value	X	Weight	=	Total
Usability	46.00		0.40		18.40
Deployability	52.25		0.20		10.45
Security	70.25		0.40		28.10
					56.95

Authentication Technique: RSA SecurID

Overall: 59.30

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	0		0.15		0.00
Easy-to-Learn	80		0.05		4.00
Infrequent-Errors	80		0.15		12.00
Easy-to-Change	90		0.05		4.50
Requires-No-User-Intervention	0		0.15		0.00
Time-Efficient-to-Use	20		0.30		6.00
Memorywise-Effortless	90		0.15		13.50
					40.00

Deployability					
	Value	X	Weight	=	Total
Accessible	100		0.35		35.00
Integrates-with-Current-Devices	60		0.35		21.00
No-Additional-Hardware	0		0.30		0.00
					56.00

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	100		0.10		10.00
Resilient-to-Smudge-Attack	100		0.10		10.00
Resilient-to-External-Brute-Force	100		0.25		25.00
Resilient-to-Internal-Brute-Force	85		0.15		12.75
Resilient-to-Theft	50		0.05		2.50
Resilient-to-Social-Engineering	100		0.05		5.00
Resilient-to-False-Positives	100		0.15		15.00
Active-Continuous-Authentication	0		0.15		0.00
					80.25

Overall					
	Value	X	Weight	=	Total
Usability	40.00		0.40		16.00
Deployability	56.00		0.20		11.20
Security	80.25		0.40		32.10
					59.30

Authentication Technique: Electronic Physical Key

Overall: 73.90

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	25		0.15		3.75
Easy-to-Learn	100		0.05		5.00
Infrequent-Errors	95		0.15		14.25
Easy-to-Change	10		0.05		0.50
Requires-No-User-Intervention	65		0.15		9.75
Time-Efficient-to-Use	95		0.30		28.50
Memorywise-Effortless	100		0.15		15.00
					76.75

Deployability					
	Value	X	Weight	=	Total
Accessible	100		0.35		35.00
Integrates-with-Current-Devices	60		0.35		21.00
No-Additional-Hardware	0		0.30		0.00
					56.00

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	100		0.10		10.00
Resilient-to-Smudge-Attack	100		0.10		10.00
Resilient-to-External-Brute-Force	100		0.25		25.00
Resilient-to-Internal-Brute-Force	100		0.15		15.00
Resilient-to-Theft	0		0.05		0.00
Resilient-to-Social-Engineering	100		0.05		5.00
Resilient-to-False-Positives	100		0.15		15.00
Active-Continuous-Authentication	0		0.15		0.00
					80.00

Overall					
	Value	X	Weight	=	Total
Usability	76.75		0.40		30.70
Deployability	56.00		0.20		11.20
Security	80.00		0.40		32.00
					73.90

Authentication Technique: Proposed Physical Key

Overall: 84.40

Usability					
	Value	X	Weight	=	Total
Nothing-to-Carry	35		0.15		5.25
Easy-to-Learn	100		0.05		5.00
Infrequent-Errors	95		0.15		14.25
Easy-to-Change	30		0.05		1.50
Requires-No-User-Intervention	100		0.15		15.00
Time-Efficient-to-Use	100		0.30		30.00
Memorywise-Effortless	100		0.15		15.00
					86.00

Deployability					
	Value	X	Weight	=	Total
Accessible	100		0.35		35.00
Integrates-with-Current-Devices	60		0.35		21.00
No-Additional-Hardware	0		0.30		0.00
					56.00

Security					
	Value	X	Weight	=	Total
Resilient-to-Shoulder-Surfing	100		0.10		10.00
Resilient-to-Smudge-Attack	100		0.10		10.00
Resilient-to-External-Brute-Force	100		0.25		25.00
Resilient-to-Internal-Brute-Force	100		0.15		15.00
Resilient-to-Theft	40		0.05		2.00
Resilient-to-Social-Engineering	100		0.05		5.00
Resilient-to-False-Positives	100		0.15		15.00
Active-Continuous-Authentication	100		0.15		15.00
					97.00

Overall					
	Value	X	Weight	=	Total
Usability	86.00		0.40		34.40
Deployability	56.00		0.20		11.20
Security	97.00		0.40		38.80
					84.40