Keypad Entry System
Group 4: Gamechangers

Ben Henaghan
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada
Student Number: 96671466

Scott Wang
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada
Student Number: 72573322

Austine Yapp
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada
Student Number: 86705340

Mike Yue
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada
Student Number: 24583156

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A. Addressed Problem

Digital door locks are becoming more and more common, and as they become more common, the vulnerabilities they suffer from become more and more dangerous. While digital locks are more convenient by offering a keyless entry and exit, they are susceptible to a slew of different attacks, all intended to derive the static PIN that unlocks the digital lock.

Assuming a 4 digit access code, simple wear and tear on the keypad means that the attacker only needs to try 4! (24) combinations at most to bruteforce the PIN, cutting down the original keyspace by almost 98% (Figure 1).

More active attacks include shoulder surfing or installing inconspicuous pinhole cameras over the lock, revealing to the attacker exactly what the combination is if the victim is unaware of the attack.

Clearly, this is a security problem, one that hinges on the assumption of a static PIN code. Therefore, we seek to address this problem by dynamically and randomly generating temporary access codes for the user on demand, rather than having the lock accept a preset access code. In other words, our system will have the digital lock, by default, reject every input as invalid. Only when users request a new code through our mobile app will the lock accept the newly generated code.

B. Importance of Problem

It can be hard to quantify or estimate the value of goods protected by electronic keypad locks. We postulate that if the location is important enough to secure with an electronic keypad lock, then the user does not want the location compromised, and thus security is important to the user.

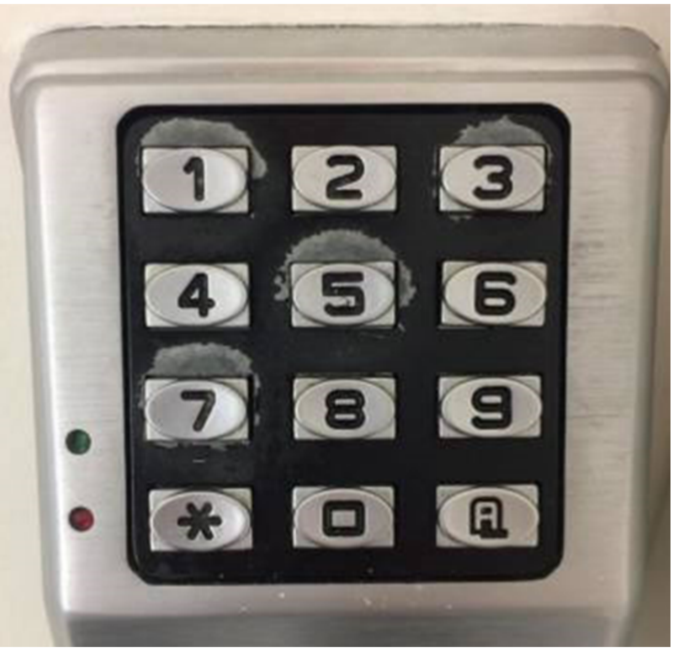


Fig. 1. Worn keypad exposes digits of PIN

However, these electronic keypads allow users to choose their own static access codes, and humans are wildly unsuited to coming up with random PIN codes. Analysis done on leaked 4 digit PINs from breaches or leaks found that 1234 accounted 11% of the 3.4 million leaked codes, and the top 20 most common PINs accounted for 26.83% of the entire 3.4 million PINS [10].

Clearly, users cannot be trusted to come up with secure random access codes, and thus cannot effectively protect themselves, which is an important problem that our design seeks to address.

C. Proposed Solution

We designed a digital-lock system that employs the use of dynamically generated TOTP access codes instead of conventional static codes. Our system uses fingerprint biometrics for authentication as an alternative to logging in using a username and password. Fingerprint authentication has been proven to be more usable and efficient than entering a username and password [1], minimizing the amount of effort and time required on the user's part. Once the user is authenticated, they can then generate a TOTP for unlocking a particular digital door-lock. Upon unlocking the digital door-lock, the valid TOTP is immediately destroyed and rendered invalid. There has been similar work done by CMU on the Grey system [2], a device-enabled authorization system utilizing smartphones to unify access controls. However, the system still utilizes a static PIN code in order to activate a user's private key for authentication. This leaves the system still highly vulnerable to adversarial attacks such as shoulder surfing. Our proposed system thus hopes to remove the need for static PIN codes through TOTP dynamic PIN codes.

D. Evaluation

We conducted a usability study to evaluate and assess our new lock system in terms of five essential criteria, speed, efficiency, learnability, memorability and user preference as suggested by Jennifer Golbeck [9]. Three undergraduates from the University of British Columbia are selected to perform the user study. A series of tasks are carefully designed to simulate the normal workflow, including login, adding a new lock, generating a code, unlocking the actual lock and etc. After these tasks, users are interviewed with several questions relating to the usability performance of our prototype. From this user study, we collected valuable advice and feedback from some real users' perspectives.

The evaluation results demonstrated our digital lock provides moderately good usability as two users rated 8 out of 10 and one user rate 7 out of 10 in terms of overall usability performance. In addition, users agree our system does provide more security features and gives them more safe feelings when using our digital lock. At the same time, this study also exposes many problems and limitations of our design. For example, it cost a little bit more time (10-15 seconds on average) for a normal unlocking operation compared with the traditional keypad. Plus, some features (e.g. expiry time) are not favored by users, and some features (e.g. showing user identity in trace data) are expected to be added.

E. Summarized Conclusion

F. Contributions

The work on the prototype was split naturally over the three components — The mobile application, server application and code which would run on the actual lock hardware. It was deemed that the android (mobile) application would form the largest amount of work, so that was split between Ben Henaghan and Scott Wang who both had some experience writing mobile applications. Mike Yue developed the server code and Austine Yapp produced the software for the lock hardware.

II. Related Work

A. Access Control System

Access control systems are classified by an electrical control and mechanical door-lock analog according to their operating method. It is possible to classify electronic door-locks based on their recognition technology, namely card recognition, number input method, and methods of bio-information recognition. These categories mainly fall into the three universally recognized authentication factors that exist today: what you know (passwords), what you have (tokens, keys) and what you are (biometrics). Passwords are considered to be one of the easiest targets for hackers. Many companies are therefore searching for more secure methods to protect the information of their customers and employees. On the other hand, biometrics are known to be highly secure and used in specific organizations. However, the hardware and maintenance cost required to upkeep such a system is relatively expensive. One way of bypassing the hardware requirements is to utilize the existing resources on mobile phones such as fingerprint scanners and/or facial recognition technology.

B. Similar Systems

One Time Password (OTP) is a single-use password that is valid for only one login session or transaction. It is often used in various fields such as banking to provide an additional layer of security. OTP challenge-response methods can be broadly classified into event-synchronization, time-synchronization, and a combination of methods [3] [4]. Every challenge-response OTP method requires users to provide a response to a challenge. The new password is traditionally generated using a mathematical algorithm based on a challenge, which is eventually used to authenticate the users. Event-synchronization method relies on the HMAC-based One-Time Password (HOTP) algorithm published in IETF RFC 4226 [5] to generate a new password. Time-synchronization OTP method is an extension of HOTP published in IETF RFC 6238 [6] which uses the current time instead of mathematical algorithms in order to generate a new password that remains valid for the duration of a specified time step. While both OTP methods offer single-use passwords, HOTP remains valid until it is used, or until a subsequent OTP is generated. In contrast, TOTP only has one valid OTP at any given time, with a smaller predefined validation window. HOTPs are thus more susceptible to a brute-force attack due to the larger validation window. One similar system is the Grey system at CMU which proposes the utilization of "smartphones" for unifying access control to both physical (e.g. doors, safes) and virtual (e.g. files, accounts) resources. The Grey system suggests the use of smartphones as the central agent through which access control is managed, namely because of their prevalent adoption in society as well as the hardware capabilities present that would enable applications to take full advantage of the rich computation, communication, and interface capabilities. It employs proof-carrying authorization (PCA) [7] extended with a new distributed proving technique with considerable efficiency improvements [8]. Our designs are similar in the fact that both our systems utilize smartphones for client-side authentication. Also, both designs are intended to conveniently authorize access to other people. One difference in our design from the Grey system is the initial set up of the lock system. In the Grey system, this initial set up is done via either Bluetooth discovery or using the phone's camera to photograph a two-dimensional barcode. Our system instead requires the user to activate the lock via a unique serial number upon first use. In order to do so, the user has to first be logged in to an account associated with our server. Subsequent attempts to unlock the digital lock will require either user account log-in or biometric authorization. Furthermore, our system employs the use of TOTP as the challenge-response OTP mechanism. Users of the Grey system utilize static PIN codes in order to activate their individual private key for authorization. Even though the PIN code entry system is abstracted to the smartphone, it is still highly vulnerable to the same adversarial attacks such as shoulder surfing and/or social engineered attacks.

III. Adversary Model

To truly explore the strengths and limits of our system, we must put ourselves in the shoes of the adversary, and reason about the objective and capabilities of such an adversary.

A. *Objective of Adversary*

The adversary has the primary objective of gaining authorized access to the secured location by entering through the door secured by a digital lock. Forced entries, whether through the secured door or other entrances, are unauthorized accesses and thus outside the scope of this adversary. To achieve this, the adversary seeks the valid PIN code to the digital lock. We assume this adversary is an opportunistic burglar, who does not care about which location they break into and will target the easiest location.

B. *Capabalities*

This adversary can install any arbitrary system on or around the digital lock that will not be noticed by the regular user, including tools such as keyloggers and pinhole cameras. Furthermore, through sleight of hand or other means, the adversary can steal the phone of the user, which has the mobile application that requests new codes installed on it. Furthermore, through social engineering, shoulder surfing, or just bruteforce, the adversary is able to unlock the user's phone and launch the mobile application.

IV. System Design

A. *Overview*

Instead of a static pin code, we use a dynamically-generated TOTP access code. First, the user must authenticate themselves using their mobile device's fingerprint scanner to unlock the device's unique fingerprint secret which was generated when the application was first started. Next, the application sends the fingerprint secret to the server for authentication. The server then sends the list of authorized locks for the user to unlock. Upon selecting a particular lock, the server generates a valid TOTP code associated with that lock, and this code is sent to the user through the mobile app. TOTP is an extension of the HOTP algorithm that uses a secret shared key and the current timestamp as inputs to a cryptographic HMAC hash function. User input in the lock is then sent to the server and verified on the server-side. If the code entered is valid, the server returns a success message and unlocks the lock. Otherwise, the server returns a forbidden message and the lock remains unlocked. Due to the design of the system, a valid code will only be generated and associated with the specific lock on the server database upon request by an authorized user. Without an authorized user's request to the server, no valid code will be associated with the lock. Any code that is entered will thus be denied entry. This essentially eliminates the threat of brute-force attempts by adversaries who attempt to guess the lock's code. Moreover, adversaries may attempt to gain access to the access codes through a shoulder surfing attack. The system employs the use of expiring TOTP codes that are valid only for a single-use. After which, the server immediately deletes the associated valid code. This means that even if an adversary were to successfully identify the user's OTP code, it becomes useless upon use. This effectively nullifies the threat of adversarial replay attacks. Furthermore, an adversary may attempt to gain access to the authorized user's device through a socially engineered attack. Even if the attacker were to have unrestricted access to the user's phone, they would not be able to generate a valid TOTP without first authenticating themselves with their fingerprint biometrics. Communication with the server will only be established if the user's identity is authenticated. This way, any requests made to the server to generate a TOTP can only be done by an authorized user. Finally, the main goal for our design is to provide significantly greater security of traditional digital locks while minimizing the additional inconvenience imposed on users. In order to do so, we kept our system compatible with existing digital lock systems and as small as possible in order to allow for retrofitting of our proposed design. Though more effort and time will be required on the part of the user and their device, we kept this as seamless and effortless as possible.

B. *Principles of Secure System Design*

Our design satisfies a number of secure system design principles.

1) Open Design: The security of our system is not based upon the secrecy of our design or implementation. Even if attackers gained access to the complete source code of our project, they would be unable to predict the next PIN code generated for any arbitrary lock from the server, and unable to gain access to any mobile application accounts to generate new PIN codes.

2) Economy of Mechanism: The method with which we generated new PIN codes, as well as the method of communication between server/lock/application have been chosen to be as simple and small as possible.
3) Complete Mediation: Every single request to the server requests authorization, and every new PIN code request on the application requires biometric confirmation before it is carried out.
4) Least Privilege: Basic users can only generate new PIN codes or fetch the current existing code, and cannot add new users to be associated with their locks.
5) Psychological Acceptability: The system was designed to be as seamless as possible for the user. While there will obviously be more friction than if this system did not exist, we attempted to minimize the impact on users.
6) Separation Privilege: Users can only generate new pin codes if they are logged in AND have access to the lock they are trying to generate the pin code for. Thus, new pin codes are granted based on more than one condition.
7) Fail-Safe Defaults: By default, locks do not have any valid access codes. By default, users do not have access to any locks until the lock administrator user grants the user access to the lock. By default, the server does not accept any incoming requests unless they are logged in and authenticated.

V. System Prototype

A. Hardware

We used a RaspberryPi Zero W to build the prototype of our digital lock system. We connected a numeric keypad to simulate a traditional keypad door lock to provide input to the RaspberryPi. Upon entry by a user, the lock system uses the requests Python module to establish a connection with the server via HTTPS. The lock system then retrieves its own serial number and sends it along with the code entered by the user. On the server-side, the code is checked with the lock associated with the serial number. If the code is valid, the server returns a success message. Otherwise, a forbidden message is returned. The digital lock then checks the response from the server and unlocks the lock if and only if the server managed to authenticate the user's code. On the digital lock side, communication with the server is not authenticated. This is intentional in order to allow a third-party to gain access through an authorized user. For example, a domestic cleaner (unauthorized) may be granted one-time access by a homeowner (authorized).

B. Server

The server acts as the central hub, accepting communications from both the digital lock and mobile application. All communications are done via HTTPS, making eavesdropping impossible. Furthermore, any requests to generate new PIN codes or access available locks must be authenticated. To this end, the server requires the mobile application to first log in and provide valid credentials, after which the server will provide a token which must be provided with every subsequent request for authentication. If a valid request is made to generate a new access code, the server uses true random number generatino to generate a new code, tag it with the requested expiry time, and returns the code to the client.

C. Mobile

There are two main activities in the Android application. The first one is login activity. Like most of the login system, the login activity help server authenticates the user by their email and password. In the real system, this activity also provides the functionalities to register, reset password and validate clients' email. For prototype, we only provide a simple login and register interface to simulate this process for the reason that this is a necessary but not the central part of our smart lock system.

Once a user logs in, the token of the response from the server would be stored in users' phones' memory for subsequent API calls. And UI will switch to the main activity, where some lock information will be shown in figure 2.

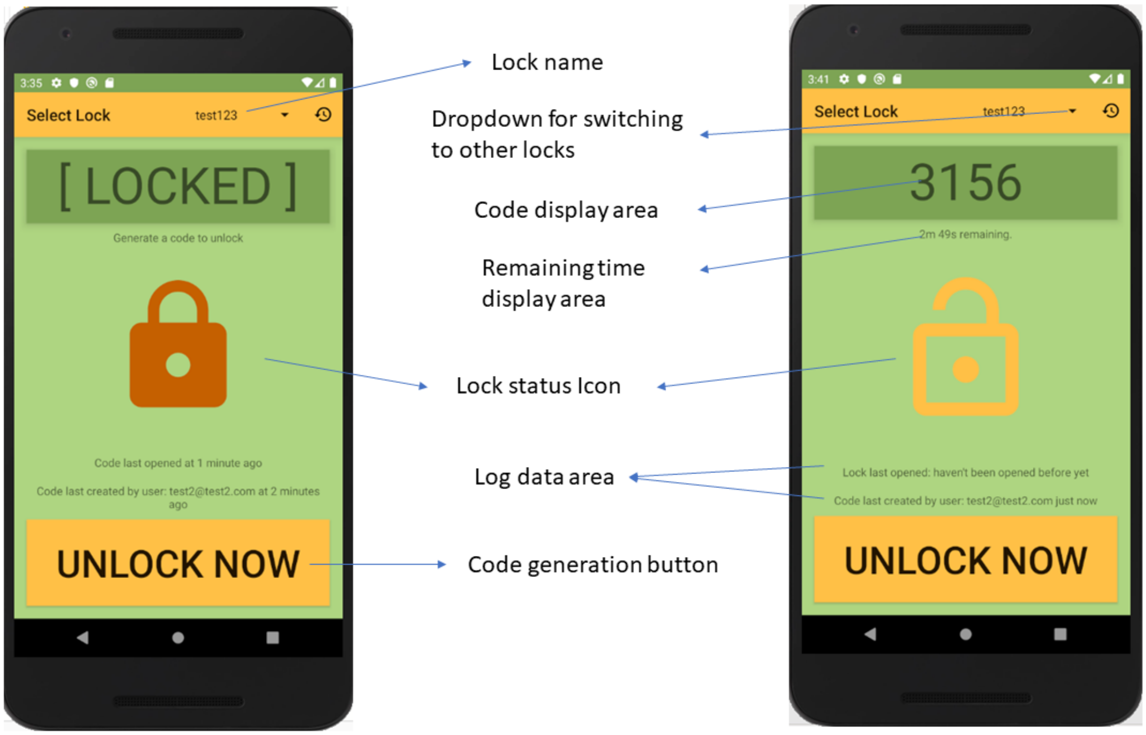
As you can see from this interface, users can easily switch to other locks by clicking the dropdown arrow. All lock names will be shown and the last row item is "Add new Lock" label where users can add a new lock by providing any necessary information to prove his or her ownership for the new lock. Once the user successfully added the new lock by clicking that item, its user-defined name will be shown in the dropdown. For the newly added lock, a closed lock icon is displayed to indicate there is not any valid code for this lock right now. Thus, users are expected to click the code generation button to generate a new code for the selected lock. Once the button is clicked, a dialogue will pop up to let the user pick the expiry date for the code, and the date picker is initiated to be one minute later than the current users' local time by default. When users confirm the expiry date, the biometrics authentication will be triggered. Once users are authenticated, the new code generated by the server will be displayed to users in the code display area. And the remaining time label below will remind users how long this code remains valid.

VI. Evaluation

A. Evaluation Methodology

We perform usability studies to evaluate the usability of our system. First, three UBC undergraduates are invited to perform the study. They all have Computer Science or engineering related backgrounds and had experience of using normal keypad lock. They all learned sufficient knowledge about security and system design during their undergraduate study, which could give us more technical advice from a professional perspective. A series of carefully designed tasks will be assigned to them, and after finishing the tasks, we will interview them individually to get some objective feedback for our smart lock prototype.

All tasks are designed to simulate the real user workflow in the real world. First, each user will download our Android application on their own Android phones (or we gave them a sample Android phone if they do not have one). Then, they



Generate a code to unlock

Code last opened at 1 minute ago

Code last created by user: test2@test2.com at 2 minutes ago

2m 49s remaining.

Lock last opened: haven't been opened before yet

Code last created by user: test2@test2.com just now

Fig. 2. Main activity user interface of Android application

will log in to our system using the test account we give to them. Second, after users have logged in, they are required to register the test lock using the lock information provided to them. In this process, users will figure out how to add locks in our Android application. Third, users are asked to generate the code for the test lock and using the code they generated to unlock the test lock, which is also the core functionalities of our prototype. We also repeated this task three times and recorded the time consumed to assess the speed properties of our system. Finally, we let users generate a code with a short lifetime (e.g. one-minute expiry time) and unlock the lock once the code is expired. After this task, users will have a better understanding of how the design of the expiry time of our system works. The digital lock will only send its own serial number and the code it received to the server. This is the only action on the server that does not require authentication. Upon receiving a valid code, the server immediately marks the code as invalid, rendering any subsequent requests with the same code invalid as well. The only viable attack against this would be to constantly bombard the server with every 4 digit PIN between 0000 and 9999, thus denying everyone access. This could be easily mitigated by limiting the number of requests the server takes from the same IP address. Furthermore, denial of access is against the interest of our adversary model, who wishes to enter the location. After users finished their tasks, several questions will be interviewed to them for assessing our system. First, we let the user rate our prototype in terms of ease of use on a scale of 1 to 10. Then, we asked them to compare and contrast our smart lock with traditional keypad lock. Finally, users are asked to give constructive suggestions for our system.

B. Study Results

All three users successfully completed the tasks, although there were some small issues happened. User A had connection issues during login, but she succeeded in the second try. User C failed to unlock the lock using the generated code in the first time, the error response showed the code is expired, however, he tried to generate a new code and passed the authentication. User B did not have any issues and finished his tasks smoothly. The total time three users used to generate the code and unlock the door is shown in table I.

TABLE I
TOTAL TIME USERS CONSUMED TO UNLOCK THE LOCK

User	A	B	C
1st Attempt	22s	20s	Failed (expired code)
2nd Attempt	17s	15s	19s
3rd Attempt	12s	13s	15s

Regarding the first interviewed question, the results are shown in table II

For the second question, user A believed she felt more secure when using our smart lock system since the code is one-time use with expiry time, while student B thought that although it is more secure than the traditional keypad lock,

TABLE II
RATES OF OVERALL USABILITIES GIVEN BY THREE USERS

User	A	B	C
Rate (out of 10)	8	7	7

our system has reduced usability, because he only needs inputs four-digit number to unlock the door previously, whereas, he has to check his phone and use biometric to generate the code. User C has a similar opinion as user B and he also added that he will prefer to use traditional keypad lock unless our prototype has easier and simple user workflow.

For the third problem, both users A and C think setting expiry time for the code seems unnecessary and redundant, instead, they prefer a default expiry time. User B mentioned he wants to know who consumed the code to open the door other than when the code was used in log area of the main activity of the app.

C. Discussion of the evaluation results

Overall, the results show our prototype is still usable as the rate shown in table 2, two users give 7 out 10 while one user gave 8 points but this study also exposes many problems and limitations of our design. We will talk about the usability of our system according to five factors, speed, efficiency (Security mistake), learnability, memorability and user preference. In terms of the speed properties, as the results listed from the table, users approximately cost 12-15 seconds on average to unlock. While the time length for authentication is not too long, it obviously reduces the speed properties compared with traditional keypad locks. However, some actions can be taken to accelerate this process. For instance, as suggested by user A and user C, we could remove the part of setting expiry time when users try to generate the code and using default expiry time instead. Meanwhile, we sacrificed the flexibility that users could set any expiry time they want.

The efficiency of our system is satisfying overall. All users did not get confused and make too many mistakes when they are performing the tasks, except some small issues do happen. Our Android application could be improved to provide a smoother user experience. For example, when once the user login, our Android app could remember the user's identity so that users do not have to login again for a while. In addition, connection issues can be reduced by improving the performance of the server.

The learnability of our system can be demonstrated in table 1, as you can see users used less time to complete the authentication to unlock when they had a second or third attempt. This proved that our system is straightforward and easy to use, implying memorability also reach the standard. Users do not have to memorize anything except checking their phones to unlock the door.

Finally, regarding user preference, on the one hand, users believed our system does provide more security features, such as login, biometric authentication and one-time door code, etc. On the other hand, users complain about some other features. Other than the expiry time feature mentioned before, users like to see more traceable data such as those who unlock the door. However, this is a big challenge with our current design because it is hard to identify who unlock the door based on four random-generated digits he or she input but it is easier to tell the user when the lock was unlocked. Some of our members proposed to use four-digit number inserted before the actual code as user identity number. Nevertheless, it is a bad design since users need to remember and press more digits than before.

VII. DISCUSSION

In this section, we will discuss the advantages and benfits of our system prototype and its broader design. We will then examine its limitations and disadvantages. Finally, we will consider the findings of our project overall, with these aforementioned 'pros' and 'cons' in mind.

Our prototype system is complete enough for us to draw useful conclusions about a similar potential product but is by no means a complete system in of itself. The simple design allowed the development team to produce an MVP (Minimum Viable Product) and then iterate on this initial product to build our prototype system. Fundamentally, our system only comprises a subset of the hardware needed for an electronic door lock, namely the authentication and control electronics. Many non-core processes have been omitted from our prototype, namely a secure lock enrollment method and a full logging system. Although our prototype was missing these features, we were still able to collect significant data about the usability of this kind of system, which could easily be incorporated into a future commercial security device.

A. Advantages

User testing showed that users appreciated the clear security benefits of our system but did find it to be a slight inconvenience. The trade-off of security and convenience is a core problem when designing security products for use by the general public; users will be hesitant to opt-in to enhanced security measures if they deem it to be a significant inconvenience. It has been shown that almost two-thirds of users will select the least secure and most convenienent security option if given the choice [17]. Our system seeked to strike a good balance between the extra security benefits of TOTP codes and the convenience of a numeric keypad lock.

All types of shoulder-surfing and replay attacks are completely defeated by our device, meaning that there aren't many possible 'subtle' (ie non-destructive) attacks against our system. However, it is worth noting that the Wi-Fi connected electronics, mobile application and server are all attack surfaces not present in standard keypad-operated access control systems. Overall risk incurred by the device is therefore reduced, as the vulnerability value is significantly decreased.

Given our adversay model, our device successfully resists most known capabilities, such as replay attacks with pinhole cameras. The adversary model does also assume that they would be able to gain access to the user's phone using a social engineering attack, and given access to the phone it is possible that the adverary would be able to comprimise the system. This is because if the threat agent is able to unlock the phone, they should be able to enrol their own biometrics and therefore activate the code generation in the android application. The risk of access in this way is mitigated in a few different ways. Enrolling a fingerprint with android and then generating an unlock code would typically takes upwards of a minute, which would likely arrouse the suspiscion of the owner. The impact of this kind of attack could be further reduced if guidance was added to the android application telling users to be aware of such social engineering attacks and not to share their phone wiht strangers without being aware of what they're doing. The other mitigation of this kind of attack relies on the expiry time of the single-use code, as the adversary would have to get to the door lock in time to use the comprimised code — which is likely take a significant amount of time due phsycial distance. A sophisticated attack could have another adversary located closer to the door, but this level of organization is uncharacteristic for an oppourtunistic threat agent.

B. Limitations

As mentioned before, the phsycial security of our prototype device was out of scope for this project, but would be an important part of the design if we wanted to bring a product like this to market. Our prototype, however, did suffer from a few limitations which impeded both security and usability.

An auto-login feature would greatly improve user effiency, and would not impair security significantly, as biometrics would still be needed to generate a code. This enhancement to the mobile application could easily half the time needed to generate a code in the app, as currently the app will force a user to log in manually if it has not been opened recently (if the activity has not yet been started or has been killed by android). Currently, this limitation significantly impairs the user experience of using our entire system, as typing a username and password is not only a time consuming acitivity, but a fairly difficult and precise process. Most users would have to stand still in order to do this, which is frustrating for a potential user. With an auto-login feature no precise actions are required, meaning a user could generate a code as they walk up to the door and would only have to stop to type in the code — the same as a traditional keypad lock.

Somewhat counter-intuitively, it would increase the overall security of the system to remove the ability of users to manually set the expiry time of codes. The current behavior limits the effectiveness of the TOTP codes and encourages users to generate codes ahead of time, making the system much less secure. This would force codes to always expire after a certain, short, amount of time (eg two minutes); which would force users to only generate codes when they intent to use them immediately. Another benefit of this change would be that the speed of generating a code would improve, as it would be another step that the user would not have to do.

Another significant limitation of our design is that a planned 'failure mode' was not integrated. This is where the lock would accept a more secure (eight digits) long-term code to unlock if the user was to not have access to their phone. This was not implemtned into our prototype due to the additional development effort that would be needed It does, also, pose a signficant security issue — as it subverts a signficant amount of the security design in the core product. We demed that it would be neccesairy in a commercial product but did not resolve the challenges associated with designing such a protocol, essentiall making it easy for the legitimate users while keeping the lock secure. This would be a great avenue for future research.

C. Overall Findings

Overall, we found that security was significantly increased by using our system in place of a traditional keypad lock, when using our defined adversary model. However it is, of course, more inconvenient to use our system over a traditional keypad-based access control system.

We believe that the compromise is likely to be accepted by the majority of security-conscious consumers, as well as many large corporations currently utilizing keypad door locks. Corporations pose signficant potential customers for this kind of system, as the value of assets protected by access control systems can be massive, and the traceability offered by a system such as ours would be very useful when many different people might need access at different times.

VIII. Conclusion

References

[1] G D Lam, "Evaluating the Usability of an Apple Touch ID-Based Access Control System" University of British Columbia, April 2015.

[2] L. Bauer, S. Garriss, J. M. Mccune, M. K. Reiter, J. Rouse, and P. Rutenbar, "Device-Enabled Authorization in the Grey System," Jan. 2005.

[3] D. Choi, W. Kim and D. Won, "One-Time Password Technology Analysis and Standardization Trend," Review of KIISC, vol. 17,no. 3, pp. 12-17, Jun. 2007.

[4] S. Seo and W. Kang, "OTP Technology Status and OTP Introduction Example," Review of KIISC, vol. 17, no. 3, pp. 18-25, Jun. 2007.

[5] "RFC 4226 - HOTP: An HMAC-Based One-Time Password Algorithm", Tools.ietf.org, 2016. [Online]. Available: https://tools.ietf.org/html/rfc4226. [Accessed: 05- Dec- 2019].

[6] "RFC 6238 - TOTP: Time-based One-time Password Algorithm", Tools.ietf.org, 2016. [Online]. Available: https://tools.ietf.org/html/rfc6238. [Accessed: 05- Dec- 2019].

[7] A. W. Appel and E. W. Felten. Proof-carrying authentication. In Proc. 6th ACM Conference on Computer and Communications Security, Nov. 1999.

[8] L. Bauer, S. Garriss, and M. K. Reiter. Distributed proving in access-control systems. In Proc. 2005 IEEE Symposium on Security and Privacy, May 2005.

[9] J. Golbeck, "Usable Security," Open CSF: Open Computer Science Foundations, 01-Aug-2016. [Online]. Available: https://www.youtube.com/watch?v=O4QbOFDUVao. [Accessed: 04-Dec-2019].

[10] Unknown author, "Pin Analyis,", Data Genetics, http://www.datagenetics.com/blog/september32012/.

[11] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[12] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[13] K. Elissa, "Title of paper if known," unpublished.

[14] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[15] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[16] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[17] Weir, C.S., Douglas, G., Carruthers, M. and Jack, M., 2009. User perceptions of security, convenience and usability for ebanking authentication tokens. Computers & Security, 28(1-2), abstract.