2018

Improving the Security of Mobile Devices Through Multi-Dimensional and Analog Authentication

Jonathan Gurary

IMPROVING THE SECURITY OF MOBILE DEVICES THROUGH MULTI-DIMENSIONAL AND ANALOG AUTHENTICATION

JONATHAN GURARY

Bachelor of Computer Engineering

Cleveland State University

2012

Master of Electrical Engineering

Cleveland State University

2013

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

May 2018

We hereby approve the dissertation

of

Jonathan Gurary

Candidate for the Doctor of Engineering degree.

SIGNATURE PAGE ON FILE WITH CLEVELAND STATE UNIVERSITY

This dissertation has been approved for the Department of

ELECTRICAL AND COMPUTER ENGINEERING

and CLEVELAND STATE UNIVERSITY

College of Graduate Studies by

____

Thesis Committee Chairperson, Dr. Wenbing Zhao

____

Department/Date

For my wife, my family, my country, for the Emperor. If the road is easy, the destination is worthless.

ACKNOWLEDGMENTS

Of course, a great thank you to my adviser, Dr. Zhao, for his tremendous help and support. A thank you to my entire committee: Dr. Dong, Dr. Simon, Dr. Wang, and Dr. Wu, for their time and dedication in reviewing this work. And thank you to the EECE department here at Cleveland State, for their financial support and for an overall excellent experience in time I spent working towards this degree. Thank you to Dr. Zhu for getting me started on this journey. Thank you to my collaborating authors from Oakland University for their help. I wish you all the very best.

This work is dedicated to everyone who supported me. I'd like to thank my wife, for being omnipresent in support and bearing with me while I finished this lengthy project. My parents, for all their love and patience as well, even if they have no idea what I'm doing "over there at school". My friends, for distracting me from finishing this sooner, but keeping me entertained in the meantime.

IMPROVING THE SECURITY OF MOBILE DEVICES THROUGH MULTI-DIMENSIONAL AND ANALOG AUTHENTICATION

JONATHAN GURARY

ABSTRACT

Mobile devices are ubiquitous in today's society, and the usage of these devices for secure tasks like corporate email, banking, and stock trading grows by the day. The first, and often only, defense against attackers who get physical access to the device is the lock screen: the authentication task required to gain access to the device. To date mobile devices have languished under insecure authentication scheme offerings like PINs, Pattern Unlock, and biometrics– or slow offerings like alphanumeric passwords. This work addresses the design and creation of five proof-of-concept authentication schemes that seek to increase the security of mobile authentication without compromising memorability or usability. These proof-of-concept schemes demonstrate the concept of Multi-Dimensional Authentication, a method of using data from unrelated dimensions of information, and the concept of Analog Authentication, a method utilizing continuous rather than discrete information. Security analysis will show that these schemes can be designed to exceed the security strength of alphanumeric passwords, resist shoulder-surfing in all but the worst-case scenarios, and offer significantly fewer hotspots than existing approaches. Usability analysis, including data collected from user studies in each of the five schemes, will show promising results for entry times, in some cases on-par with existing PIN or Pattern Unlock approaches, and comparable qualitative ratings with existing approaches. Memorability results will demonstrate that the psychological advantages utilized by these schemes can lead to real-world improvements in recall, in some instances leading to near-perfect recall after two weeks, significantly exceeding the recall rates of similarly secure alphanumeric passwords.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS iv

ABSTRACT v

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTER

I. OVERVIEW AND MOTIVATION 1

1.1 Mobile: An Opportunity for Change 1

1.2 Shortcomings of the Current Paradigm 4

1.3 Statistical Testing 7

1.4 Contributions and Outline 7

II. MULTI-DIMENSIONAL AUTHENTICATION 10

2.1 Outline 10

2.2 Introduction to Multi-Dimensional Authentication 11

2.2.1 An Example of MAPS 12

2.2.2 MAPS vs Traditional Authentication 13

2.3 Related Work: Graphical Passwords 15

2.4 Chess Based MAPS (CMAPS) 21

2.4.1 Graphical Hints 22

2.5 Security Strength of MAPS 24

2.5.1 Security Strength of MAPS 24

2.5.2 Security Strength of CMAPS 26

2.6 Usability Analysis 30

2.7 User Study 31

2.7.1 Overview 31
2.7.2 Apparatus 32
2.7.3 Conditions 33
2.7.4 Participants 33
2.7.5 Memorability 34
2.7.6 Usability 35
2.7.7 Hotspots 41
2.7.8 User Choice in CMAPS Passwords 43
2.7.9 Graphical Hints Generated by Participants 45
2.8 Discussion 46
III. SHOULDER-SURFING RESISTANCE 48
3.1 Outline 48
3.2 Expanding MAPS to Reduce Shoulder-Surfing 49
3.2.1 CMAPS vs Shoulder-Surfing and Smudge Attacks 49
3.2.2 PassGame: Adding Shoulder-Surfing Resistance to MAPS . . 50
3.3 Related Work: Shoulder-Surfing Resistance 51
3.3.1 Testing Shoulder-Surfing 52
3.3.2 Hardware-based Shoulder-Surfing Resistance 53
3.3.3 Challenge-Response 54
3.4 The Design of PassGame 55
3.4.1 Random Board Generation 56
3.4.2 Available Rules 57
3.4.3 Additional rules 60
3.5 Security of PassGame 61
3.6 PassGame User Study 62
3.6.1 Participants 62
3.6.2 Overview 63

3.6.3 Memorability Results 64

3.6.4 Usability Results 65

3.6.5 User Choice in PassGame 68

3.6.6 Shoulder-Surfing Study 69

3.7 PassGame Discussion 71

IV. AUTHENTICATION IN VR 74

4.1 Outline 74

4.2 Expanding MAPS to Virtual Reality 75

4.3 VR Introduction and Related Work 76

4.4 Advantages of a 3D Authentication Scheme 77

4.4.1 Psychological Phenomena 77

4.4.2 Physical Phenomena 79

4.5 Implementation of 3DPass 81

4.5.1 Input Device 83

4.5.2 Design Considerations 85

4.6 Security Strength of 3D Authentication 86

4.6.1 Password Space of 3DPass 87

4.7 3DPass User Study 91

4.7.1 Procedure 92

4.7.2 Memorability Results 94

4.7.3 Usability Results 95

4.7.4 Hotspots 98

4.7.5 User Choice in 3DPasswords 99

4.8 Discussion of 3D Authentication 100

V. BEHAVIORAL PASSIVE AUTHENTICATION 102

5.1 Outline 102

5.2 Introduction to Implicit Authentication 102

5.3 Related Work: Implicit Authentication . . . 103

5.4 Implicit Biometric Authentication Scheme . . . 105

5.4.1 Future Implementation . . . 107

5.5 Experiment . . . 108

5.5.1 Devices Used . . . 108

5.5.2 Experiment Setup . . . 109

5.5.3 Typographical Correction . . . 109

5.5.4 Classification and Analysis . . . 110

5.5.5 Character Independent Classification . . . 111

5.5.6 Character Dependent Classification . . . 112

5.5.7 Order Dependent . . . 114

5.5.8 Future Approaches . . . 116

5.6 Discussion . . . 117

VI. ANALOG AUTHENTICATION . . . 119

6.1 Outline . . . 119

6.2 Introduction to Analog Authentication . . . 120

6.3 Authentication Using Continuous Information . . . 121

6.4 Related Work: Analog Authentication . . . 123

6.5 The Design of PassHue . . . 124

6.5.1 Comparison of Color Values . . . 128

6.6 Security Strength of PassHue . . . 130

6.7 PassHue User Study . . . 131

6.7.1 Data Collection . . . 132

6.7.2 Participants . . . 133

6.7.3 Memorability of PassHue . . . 135

6.7.4 Usability of PassHue . . . 135

6.7.5 Color Selection and Hotspots . . . 140

6.7.6 Shoulder-Surfing Resistance 142

6.8 Discussion 145

6.8.1 Color Blindness and Tetrachromacy 145

6.8.2 Gender Bias 147

6.8.3 Inclusion of Additional Colors 148

VII. CONCLUSION 149

7.1 Summary 149

7.2 Future Work 151

7.2.1 Planned Improvements 151

7.2.2 Upcoming Works 153

BIBLIOGRAPHY 154

LIST OF TABLES

Table Page

I Number of Gestures Required for Different Password Spaces 31

II Recall Rates of CMAPS Passwords. 34

III CMAPS Mean Password Entry Time 36

IV Pairwise Testing on Password Entry Time (Single Correct Attempt) 37

V Average Usability Rating of CMAPS and Other Schemes. 38

VI Statistical Analysis on Usability Data for CMAPS. 39

VII Statistical Analysis, CMAPS vs Other Schemes 40

VIII Password Space of PassGame Rules 61

IX PassGame Recall Rates by Condition 64

X Average Entry Times, New Boards, and Attempts Needed per Successful Authentication 65

XI PassGame and PIN Average Survey Ratings 67

XII Successful Shoulder-Surfing Attempts by Condition 69

XIII Recall Rates of 3DPasswords and Alphanumeric Passwords (one week after initial setup). 94

XIV Presence Survey Results of 3DPass 95

XV Usability Survey Results of 3DPass 97

LIST OF FIGURES

Figure Page

1 Screenshots of the CMAPS Implementation 21

2 Example Graphical Hints 23

3 Password Space Between One and Twenty Gestures 27

4 Password Space at Two, Four, and Eight Gestures 27

5 Visualization of the Password Space of CMAPS 29

6 A CMAPS Password Completed in One Long Gesture 30

7 Survey Results 39

8 Popularity of Tiles 41

9 Popularity of Different Piece Types 42

10 Example Graphical Hints Created by Users. 45

11 A Screenshot of Rule Selection (left), The Rule Selection Prompt (right) . . 55

12 A Screenshot of Authentication 56

13 Usability Survey Results for Convenience (left), Speed (right). 67

14 Frequency of Rule Selection 68

15 An Overhead View of 3DPass Taken in Unity 81

16 Screenshots of the 3DPass Application 82

17 Teleporter Room 85

18 State Diagram for a 3D authentication Scheme 86

19 Number of Possible Passwords Using Various Metrics 90

20 Distribution of Objects in the 3DPass Environment (left) Actual Usage of Environment by Participants (right) 98

21 Screenshot of the Android Keyboard Implementation. 105

22 Touches vs Accuracy and FAR/FRR for Character Independent Data 111

23 Touches vs Accuracy and FAR/FRR for the Character “a” 113

24 Touches vs Accuracy and FAR/FRR for the Character “l” 113

25 Touches vs Accuracy and FAR/FRR for the Character “Space” 113

26 Touches vs Accuracy and FAR/FRR for Multiple Consecutive Touches . . . 115

27 Tutorial Images Shown on the Store Page 125

28 The Password Setup Screen (left), The Login Screen (right) 126

29 Cone Representation of HSV Color Space 127

30 Median Entry Time of PassGame Users Over Time 136

31 Authentication Sessions With Failures 137

32 Authentication Sessions With Failures (Outliers Removed) 137

33 Failed Authentication Attempts per Session Over Time (Outliers Removed) 138

34 User Survey Responses by Condition 139

35 Colors Selected by Participants 140

36 Colors Selected by Male (upper) and Female Participants (lower) 141

37 All PassHues Chosen by Participants 142

38 PassHue Shoulder-Surfing Experiment Start Screen 143

39 Shoulder-Surfing Images 143

40 Shoulder-Surfing Results for PassHues 1-4 at 1 View and 3 Views 144

41 The Passhue Wheel Seen With Minor Deuteranomaly 145

42 Color-Blind Participant’s PassHue 146

CHAPTER I

OVERVIEW AND MOTIVATION

1.1 Mobile: An Opportunity for Change

Alphanumeric passwords for authentication were invented in the early 60’s, a time when keyboards were typically the sole available input device and displays could only handle one color. Since then, the tradition of using alphanumeric passwords for the bulk of authentication has been driven largely by the sentiment of “if it ain’t broke, don’t fix it”, with relatively few changes to the way we do authentication since its inception. Authentication has largely skipped over the invention of the mouse, the gradual improvement of the high resolution color display, and the general advancement of computing power. From the user’s perspective, authentication today is largely the same as it was in the 60’s. Even Fernando Corbato himself, credited with the invention of the alphanumeric password, describes the modern day use of alphanumeric passwords as a “nightmare” [1].

The problems with alphanumeric authentication are numerous and well-known even to the layman [2, 3, 1, 4, 5]: passwords are difficult to remember, frustrating to update or change, tedious to type on anything without a proper hardware keyboard, and often insecure. Passwords are easy to steal by looking over the victim’s shoulder (often called *shoulder-surfing*), so most applications no longer show the password text on the screen, leading to even more difficult and error-prone entry. Short passwords are insecure against brute force attacks, so most applications require eight characters or more, mixing and matching requirements for symbols, capital letters, and various other requirements in an effort to force users to generate secure passwords. Because users often pick poor, easily brute-forced passwords, corporations often require changing passwords every few weeks or months, leading to memory interference and further frustrations. Remembering multiple passwords at once, especially with different rules, is incredibly difficult, encouraging password reuse, password resets, and often costly calls to customer service. Passwords are easy to communicate and write down, leading to the ubiquitous sticky note on the monitor that defeats even the most vigilant IT security efforts.

Despite all the problems associated with alphanumeric passwords, the impetus to replace them has been historically small. Alphanumeric passwords are simple to understand; anyone with knowledge of letters and numbers can easily make one, even literacy isn't necessarily a requirement. Hardware keyboards are a given for any computer system, and even amateur typists can authenticate relatively quickly. For the most part, users are willing to put up with alphanumeric authentication on traditional computers, it's simply not bad enough to overcome inherent resistance to change.

Recent developments such as Single Sign-on, password managers, and secure cookies have alleviated some of the burden of authentication by allowing users to interact less with their passwords, but the authentication process itself remains as archaic as ever. Many of these solutions come with issues of their own, such as reduced memorability from lessened exposure to the password. This work does not address Single Sign-on or other methods that allow the user to avoid entering a password for every application they use, but instead focuses on improving the core authentication experience.

Enter modern mobile devices: smartphones, tablets, phablets, and more. These devices are small computers, unique in many ways, but almost all of them lack one essential item: a hardware keyboard. Entry time on mobile “soft” keyboards is slow and error-prone [6, 7], with average alphanumeric password entry times typically exceeding 20s [8]. An average mobile phone user unlocks their device 48 times a day [9], so using alphanumeric authentication to lock the device would take over two hours a week. Clearly, alphanumeric authentication for mobile devices is completely unacceptable from a usability standpoint. Using alphanumeric passwords on mobile devices can also lead to poor security. Not surprisingly, when faced with annoyingly long entry times, users tend to pick poor, insecure passwords [10] that are easier to enter. Therefore, attempting to apply the alphanumeric paradigm to mobile devices can actually weaken its desktop counterpart.

As mobile devices gain popularity and complexity, users are increasingly likely to use their mobile device for email, banking, and many other secure applications. Increased frustration with traditional passwords has led many developers to utilize alternative, less secure, authentication methods. One example is Credit Karma, an application which stores a person’s financial information, and is secured by 4 digit Personal Identification Number (PIN). Even large banks, such as Chase, have permitted sign in to banking applications using fingerprint authentication.

The advent of mobile devices presents a unique opportunity to revolutionize authentication altogether. For a long time, alphanumeric passwords have been simply good enough, but on mobile devices, alphanumeric authentication doesn’t even reach the good-enough standard. This has prompted a frenzy of authentication development trying to create a robust scheme for mobile devices.

Once it builds familiarity, an authentication scheme designed for mobile can one day spread back to traditional computer environments. We are already seeing the trend of preferring mobile authentication with the rising popularity of two-factor authentication–using the mobile device’s lock mechanism as a type of secondary password by asking for mobile device input in addition to a traditional password. Some desktop applications, for example Microsoft accounts, are transitioning to authentication using only a mobile phone, with a password only as a backup. Furthermore, whatever works on mobile may be applied to smart TVs, wearables, and even VR and AR in the future. In other words, mobile authentication is the frontier, whatever dominates the mobile sphere in the near future will likely dominate authentication for years to come.

1.2 Shortcomings of the Current Paradigm

While biometric authentication is certainly quite popular and subject to rapid development across the industry, it will likely never be a true substitute for knowledge-based authentication. Biometric information can always be stolen, and once it's stolen, it's stolen forever. The 2015 hack of the US Office of Personnel Management [11] resulted in the loss of 5.6 million individual fingerprints. These fingerprint images can easily be used to bypass fingerprint authentication like TouchID, meaning that affected individuals will never truly be secure when using fingerprint authentication. This incident should serve as a chilling warning that biometric data can be stolen even from entities as large as the US government, let alone private organizations and public spaces.

The legality and practicality of biometric authentication as a defense against the state is also an important factor. Many modern mobile devices support total device encryption, unlocked only by the phone's unlock mechanism. Citizens of the United States and many European nations can be legally compelled to provide fingerprints, blood, palm prints, photographs, or various other biometric information as part of a criminal investigation–meaning that biometric security provides effectively zero protection against the state. The debate over whether a person can be compelled to disclose their password is not yet settled [12, 13, 14], however it is clear that law enforcement can attempt to break into a suspect's device [15], meaning that a knowledge-based password's protection against the state is as strong as the authentication scheme. In some cases where the password could be compelled [16], punishment for "forgetting" the password is lesser than the potential punishment for the alleged crime, while other cases have resulted in indefinite detention for refusal to provide the password. If a biometric password is used, refusing is not an option, the state will simply compel the defendant to unlock it.

Biometric schemes are notoriously easy to defeat because the information they use is so easily accessible in the age of ubiquitous cameras and surveillance. Combined with printers or even 3D printers, the information biometric schemes use is often easily reproducible. Most major biometric technologies that ship with mobile phones are successfully defeated within days of their release. Fingerprints are left behind everywhere, and Chaos Computer Club was able to break TouchID [17] using only a high resolution photograph of a fingerprint and a laser printer. Older facial recognition technologies could be hacked with mere photographs of the user's face, while newer technologies like the iPhone X's can be defeated with a 3d printed mask and 2d printouts of portions of the user's face [18]. Iris scanners such as the Samsung S8's have been defeated using a simple high resolution photo of the eyes with rounded contact lenses glued over it [19].

Perhaps the most telling point is that no major manufacturer allows the use of a biometric scheme on its own. Either because of potential hardware failure or as limiter against too many successive bad attempts, all biometric authentication methods require the user to set a knowledge-based backup password, typically a PIN. Attackers are effectively given a choice, they can hack the biometric scheme or the knowledge-based one, whichever is less secure.

While the usability advantages of biometrics are undeniable, and their value as a form of identification or as a tool for authentication is not entirely without merit, biometrics are not necessarily a good first option for users seeking robust security. Indeed there are few, if any, cybersecurity firms that suggest a transition to biometrics as the sole, or even primary method of authentication. While supplementing authentication with biometrics can improve usability and security, for the foreseeable future, it seems that authentication will be based *primarily on knowledge*.

With that in mind, let us consider the current state of knowledge-based authentication on mobile platforms. PIN is still used by the plurality of mobile device owners [20]. PIN, and its *graphical* contemporaries like Pattern Unlock– which we will discuss in more detail later– share one essential shortcoming: they rely on a single unit of repeating information. Alphanumeric passwords rely on letters, numbers, and symbols in sequence, PIN relies on numbers in sequence, and Pattern Unlock relies on a sequence of connected dots.

In existing authentication methods, the user remembers a single piece of information and recalls it back exactly, but this is a poor use of human memory potential. Humans are bad at remembering things, particularly long sequences of information. Our memory is generally limited to seven [21], or perhaps even fewer [22], items in sequence at a time. In general, human memory for "random" strings of letters and numbers is relatively poor, and organized strings are vulnerable to brute force attacks. Multiple passwords are demanded of users, but memory interference is a common occurrence when working with internally similar information like letters and numbers, causing people to confuse one password with another. As we will discuss later, many different types of human cognitive ability go untouched. Authentication today rests firmly in the realm of rote memorization and repetition, one of the weakest kinds of memory.

Most importantly, conventional authentication uses human effort inefficiently. A single touch or gesture on the screen performs at best just one action: a single selection of digit, letter, or other unit of information. On a keyboard, this was an efficient use of effort, a key can only be used to select one unit of information. On modern devices that feature multi-modal inputs, especially precision inputs like touchscreens, relying on one-action, one-unit-of-information is plainly inefficient.

In cases like Pattern Unlock, an entire swipe gesture is needed to communicate a single piece of information, the connection between two dots. In PIN, a tap gesture communicates a digit. PIN and Pattern Unlock are undoubtedly fast, requiring only a handful of touches per session, but they are also insecure by that same virtue. A single gesture offers relatively little information, and a handful of these low-information choices is only a small improvement.

This work presents several approaches to generating usable authentication schemes that are also secure. The chief mechanism for doing so, as we will see, is *improving the amount of information available in a single touch*. The crux of the authentication problem today, to summarize, is simply inefficient use of human memory and inefficient use of human labor. This work will address a few different types of human memory, some untapped by authentication to date, and show how one touch can be used to choose from a much wider array of information than just a handful of letters or digits. This work will present the design and evaluation of five proof-of-concept authentication schemes that may one day be used in some form for mainstream authentication.

1.3 Statistical Testing

In this work, a significance level of .05 is used for hypothesis testing. For omnibus comparisons between categorical and continuous data, Chi-squared (χ^2) and Kruskal-Wallis (KW) analysis are used respectively. If the omnibus test is significant, pairwise testing is done with Chi-squared and Mann-Whitney for categorical data and quantitative data respectively.

1.4 Contributions and Outline

In this section, the contributions and basic structure of each chapter will be briefly summarized. In each chapter, a concept is introduced, followed by the design of a proof-of-concept scheme based on this idea. A user study is presented to study the security, memorability, or usability of the scheme using various relevant metrics.

Chapter 2, Multi-Dimensional Authentication, introduces the concept of a Multi-Dimensional Authentication Scheme (MAPS), a framework that will be used in Chapters

2, 3, 4, and 5 to develop secure authentication schemes. The concept of MAPS itself is a novel one, no other work has formally defined a similar concept for purposes of authentication. CMAPS, a proof-of-concept graphical example of MAPS, is used to demonstrate the potential advantages of a MAPS. CMAPS achieves 8-character-alphanumeric equivalent security strength using just 6 gestures, while maintaining up to 100% memorability over one week and achieving promising early timing results.

Chapter 3, Shoulder-Surfing Resistance, extends MAPS and CMAPS to achieve protection against observation based attacks, typically referred to as shoulder-surfing. This chapter introduces the idea of a challenge-response authentication scheme, a concept that is generally reserved for machine-to-machine communication, and applies this concept to human authentication. PassGame, a challenge-response scheme that utilizes the concept of MAPS and the basic design of CMAPS, proves itself to be extremely resistant to shoulder-surfing, with most participants failing to crack even a medium strength PassGame password after viewing it 30 or more times. Although PassGame does have high entry times, its superb shoulder-surfing resistance and high memorability indicate that PassGame can be a viable secondary password for usage when the user is afraid shoulder-surfing may be a risk.

Chapter 4, Authentication in VR, addresses the design of an authentication scheme for virtual reality or 3D displays. This chapter features a novel breakdown of the physical and psychological advantages of 3D authentication, and a novel analysis of the security of a general 3D authentication scheme. The analysis demonstrates how easily a 3D authentication scheme can achieve high levels of security. Unlike previous works, navigation in the virtual space is used as part of the authentication process. 3DPass, an example of 3D authentication, proves significantly more memorable than its alphanumeric counterpart after a two-week period, and demonstrates excellent results in qualitative user response as well as promising results in entry time. The concept of MAPS is easily applied to 3Dpasswords, where multiple dimensions are already inherently present.

Chapter 5, Behavioral Passive Authentication, addresses the use of typing behavior to identify mobile users. Unlike previous works on this topic, using the concept of MAPS, information is collected from as many dimensions as possible, including timing, location, and acceleration data. User studies show that using all of this information, combined with several novel approaches to classification, can lead to accuracy exceeding 97% in identifying users.

Chapter 6, Analog Authentication, presents another novel concept. In Analog Authentication, continuous information is used instead of discrete information, an idea that is often referenced in works on biometrics and gesture-drawing, but one that has not been generalized for authentication in any other work. PassHue, a proof-of-concept analog authentication scheme, shows that analog schemes can greatly exceed the security strength of similar discrete schemes such as PIN, while offering on-par entry times, near-perfect memorability, reduced hotspots, and some resistance to shoulder-surfing– all demonstrated with an in-the-wild user study.

Chapter 7 summarizes and concludes this work.

CHAPTER II

MULTI-DIMENSIONAL AUTHENTICATION

2.1 Outline

A short, preliminary version of this chapter was published at the Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS 2015) [23].

Section 2.2 introduces the novel idea of a Multi-Dimensional Authentication Scheme (MAPS), presents a short, simple example of MAPS, and briefly addresses potential advantages of MAPS vs traditional authentication. Section 2.3 addresses related works in graphical authentication, current commercial authentication schemes, and existing schemes that use some of the concepts of MAPS. The design of Chess-Based MAPS (CMAPS), a novel proof-of-concept graphical MAPS, is introduced in Section 2.4. The security strength of MAPS in general and CMAPS is analyzed in Section 2.5. The usability of MAPS and CMAPS vs traditional authentication in terms of gestures required for authentication is analyzed in Section 2.6. A user study analyzing memorability, entry times, qualitative user preference, and hotspots of CMAPS is presented in Section 2.7. Future plans for CMAPS are discussed in Section 2.8.

2.2 Introduction to Multi-Dimensional Authentication

There is no so-called “silver bullet” for authentication that can address the issues of usability, security, and memorability at the same time [24]. Improving one almost always comes at the expense of another. Developing a mobile authentication scheme requires careful consideration of these three key elements.

Security: The scheme should safeguard the user’s device and data against attackers. Security is a combination of many factors, most importantly the number of possible passwords generated by the scheme, often referred to as *password space*. Breaking a password by exhaustively searching through its password space is referred to as a brute force attack. While the theoretical password space is significant, it is more important to consider effective password space, or the number of passwords that would be realistically used in practice. For example, in alphanumeric schemes, a string of 12 unrelated characters and symbols is unlikely to be used by anyone, and the fact that a particular combination of unrelated characters is possible does not necessarily improve security for the majority of users. Attackers are skilled at creating dictionaries to address commonly occurring patterns in passwords, often referred to as hotspots. The mitigation of hotspots is another crucial factor in improving security. The vast majority of users will find that at least part of their password lies in the dictionary of an attacker, be it a word, a year, or any other otherwise ordered sequence of information. A well constructed dictionary can vastly reduce the effective password space, and thus the security strength, of a password scheme. There are also risks associated with password observation. Shoulder-surfing attacks, when the attacker observes a password being entered, are the most common concern, and will be addressed in more detail in the next chapter.

Memorability: The user’s password should be easy to remember, both in the short and long term. Some passwords are designed for daily use, and therefore are not especially concerned with long term memorability. Other passwords, especially those associated with high security applications like banking, may not be used for weeks or months at a time, necessitating high long term memorability.

Usability: The scheme should be fast and easy to use. Usability is king on the mobile platform because mobile devices are used frequently throughout the day and often just for moments at a time. With an average of 48 device unlocks a day [9], a difference of one second between authentication schemes can cost the user hours in the long term. Entry time is therefore the first and foremost concern of mobile device authentication. Cognitive load is also an important factor to consider in usability. Does authentication require the user to divert significant intellectual attention to the device? Even if it's fast, mobile users may not be content to use a scheme that's considered hard.

The *Multi-dimensionAl Password Scheme* (MAPS) seeks to solve the problem of reconciling these three elements by **improving the amount of information communicated in a single action**. MAPS depends on the concept of *dimensions* of information. A dimension is simply a single type of information, for example color, size, shape, or letter. In a MAPS, the choosing of values from multiple dimensions is fused into a single action. Since mobile devices with touch screens are our primary concern, we will use the words action and touch interchangeably.

2.2.1 An Example of MAPS

Consider a simple extension of 4 digit PIN that adds an extra color dimension. The user is presented with the digits 0-9 in red on one side of the screen, and in blue on the other. The user is now able to chose digit and color with a single touch, extending the password space from 10^4 to 20^4, a 16-fold increase. Usability remains largely the same, since the user still has to make just 4 touches. Furthermore, by duplicating single digits and avoiding more complex double-digit numbers, the memorability impact is potentially reduced compared to simply giving the user a choice between the numbers 0-19. By including color, a dimension which is arbitrary relative to the choice of digit, the task of brute forcing a PIN based on numerical patterns is made significantly more complicated. Since the dimensions have no relationship to each other, the attacker needs to create a separate dictionary for patterns in each dimension. A MAPS can also reduce memory interference by altering the type of information available for authentication in each environment. For example, the user's bank account may feature a PIN using the colors red and blue, while the user's stock market account may use the colors green and purple.

Consider the addition of another dimension, for example hold time. The user can touch the digit with a short tap, or a long tap. Usability may not appreciably effected, only 4 touches are required, and a long touch requires only a fraction of a second more than a short touch. On Android for example, a long press is as few as 500ms. If we assume a short tap is 100ms, then the difference between 4 short taps and 4 long taps is roughly 1.5 seconds. The password space is now $(20 * 2)^4$, because there are two hold options for each on-screen digit, a 256-fold increase compared to traditional 4-digit PIN, and a larger password space than traditional PIN can produce with 6 digits (10^6). An attacker would now need to generate a dictionary for numerical patterns, color patterns, and hold time patterns to brute force the password effectively. Note that when calculating security strength, information from different dimensions is treated *multiplicatively*. A more rigorous demonstration on calculating the security strength of MAPS is found in Section 2.5.

2.2.2 MAPS vs Traditional Authentication

We've seen how MAPS, by fusing information from multiple dimensions into a single action, has the potential to improve security with minimal impact on usability and memorability. Traditional passwords are *single dimensional*, they contain a single element, for example characters in alphanumeric passwords, repeated many times. There are several disadvantages to single-dimensional approaches.

To increase security strength, more choices are often made available for the single dimension, for example by allowing special characters in alphanumeric passwords. Users may not be interested in added choices, and indeed, use of capital letters and special characters in alphanumeric passwords is typically low or laughably predictable. In other cases, for example Google's Pattern Unlock, there are practical limits to how large the grid can become before usability becomes an issue. Thus adding more choices to a dimension may not actually result in significantly increased security, and there is often a practical upper limit to how many choices a single dimension can have.

The security strength of a single-dimensional password is heavily dependent on length. To satisfy increased security requirements the user has to chose longer passwords–typically over 8 characters for alphanumeric passwords used for banking and other secure applications. Humans have difficulty remembering sequences of more than 7 items [21], which leads users to pick words and other easily guessable sequences of characters in order to satisfy length requirements while maintaining memorability. Furthermore, long passwords have even poorer usability on mobile platforms, resulting in even worse password choices [10]. In some cases there are upper limits on length, especially with schemes like Google's Pattern Unlock where choices (links between dots) cannot be reused. Both memorability and usability are impacted by length: in general, the more secure a single-dimensional password is, the longer it will take to input, and the harder it will be to remember.

Because length corresponds to security, single-dimensional passwords can only trade security for usability. A shorter password is faster to use, while a longer is one is slower. A multi-dimensional password can increase security without increasing the number of actions required from the user by increasing the number of dimensions in use. The user still has to remember more information, but the same number of actions are needed.

Memory interference can occur between different single-dimensional passwords or within the same password. Because a single-dimensional password is generated by repeating the same type of information several times, the user may have trouble remembering the beginning part of a password when the latter part is being memorized, or conflate different passwords that were set using the same type of information [25]. This is particularly an issue with password expiration policies. Users may confuse current passwords with previous generations of passwords, or worse, use a password with only some minor variation from the previous generation to avoid memory interference.

2.3 Related Work: Graphical Passwords

Because humans primarily engage with visual information, MAPS is envisioned as a *graphical password*. Graphical passwords were originally proposed by Blonder [26] in 1996. Blonder's implementation, intended originally for Personal Digital Assistant (PDA) devices, shows users a number of "tap regions" in a preselected image and asks them to set a password by arranging these regions by location and sequence. For authentication, the regions are hidden from view, leaving only the original reference image, and the user must select the now-hidden regions in the same sequence.

Graphical approaches were assumed to be more memorable than traditional passwords because the human brain is weak at remembering sequences of numbers and letters but good at processing visual data [26, 27]. This phenomenon is often called the *picture superiority effect*, and is well supported in psychology [28, 29]. The picture superiority effect has already revolutionized several other fields, for example advertising [30], which has moved to be far more visual-oriented over time. Mobile devices featuring touchscreens are especially well suited to manipulating visual information. Graphical authentication methods have been shown to have various advantages in memorability [31]. Tullis [32] even shows that some graphical passwords can achieve 96% recall after six years, with no use in the interim.

Graphical authentication schemes are typically grouped into three categories: *recognition, recall, and cued-recall* [33]. These classifications are based on human memory “tasks” outlined in psychology research [34], where recognition is considered the “easiest” task for human memory and recall, sometimes more specifically called *free recall*, is considered the most difficult. In recognition, the subject is tasked with merely identifying if something is familiar, for example asking if a person has seen a certain picture before. Recall requires direct access of information stored in memory, for example asking a person to reproduce a drawing. Cued-recall provides a hint, such as the background of the drawing, but still requires the subject to draw from memory.

Recognition Based

Recognition based schemes, such as Deja Vu [35], prompt the user to identify previously selected images. Users initially create a portfolio of images, taken from a large set of abstract pictures consisting of basic fractal and color patterns. To authenticate themselves, users must pick images from their portfolio out from a number of decoy images. Set up and login times were longer for Deja Vu versus traditional passwords, but users were better at remembering their Deja Vu passwords. Passface [36] is a commercial example of recognition-based authentication built for the open market. Passface works largely in the same way as Deja Vu, except that pictures of human faces are used in place of abstract images. Davis *et al.* [37] concluded that using familiar imagery such as human faces weakens graphical schemes, as it opens them up to various selection biases. Nicholson *et al.* [38] found that Passface users prefer faces from certain groups, for example elderly people remember PassFace passwords better when faces of older people are used. The methods developed in this work seek to use common imagery that should have minimal age, gender, or cultural biases.

Recall Based

Recall based schemes, such as Draw-A-Secret [39], prompt users to recreate a drawing or series of gestures. Users create a Draw-A-Secret password by drawing line gestures on a touch screen PDA, and authenticate themselves by reproducing those lines. Xside [40] is a more recent recall based scheme designed for modern devices that allows users to draw gestures on a separate touchscreen on the back of the device. Recall based schemes tend to have issues with good user password choice; many users tend to draw shapes, letters, and other simple images [41].

Cued-Recall Based

Cued-Recall schemes, such as Passpoints [42], ask users to recreate a drawing or a series of gestures, but provide some sort clue to the user, typically a background image. Users of Passpoints are asked to specify "click-points"– areas that need to be touched in a predefined image. Authentication is achieved by touching all of the click points in the image. The concept is based around a user choosing a personal image, for example a picture of a star, and choosing click points that are memorable or meaningful to the user, for example the points of the star. As one would expect, cued-recall schemes are often prone to hotspots: users are more likely to choose certain parts of an image for authentication, opening up the possibility for guessing attacks [43]. Windows Picture Password follows the same principle as Passpoints, allowing line and circle gestures in addition to taps, but is similarly vulnerable to guessing attacks due to hotspots in images [44, 45]. Perhaps in acknowledgment of this limitation, Windows allows 5 attempts at the Picture Password before forcing the user to enter an alphanumeric password instead, and also does not allow Picture Passwords for remote access.

Commercial Schemes

Early mobile devices such as PDAs relied primarily on Personal Identification Number (PIN) authentication, with some security-conscious users opting to use an alphanumeric password. Because these devices typically did not carry important, sensitive information, security was not a mainstream concern.

The iPhone, first released in 2007 and typically credited with spearheading the design of the modern mobile device, followed the PDA in using the PIN model. Today, PIN is still the default authentication method to unlock most modern mobile devices, typically 4 numbers long. A 4 digit PIN using the digits 0-9 has $10^4 = 10,000$ possible passwords. The default PIN scheme is clearly intended to discourage unmotivated attackers, not to stop serious adversaries. Some operating systems support more secure options for PIN, for example iOS supports an option to wipe the system after a certain number of incorrect attempts, but this can be very inconvenient if the user accidentally uses too many attempts or passes the device to a small child. This wiping mechanism, used by one of the San Bernardino terrorists to secure their iPhone, received a flurry of national media attention before ultimately being defeated by a private contractor for just under one million dollars [46].

Several research schemes have sought to improve on the basic PIN. SwiPin [47] takes advantage of gesture recognition capabilities on mobile devices for input rather than classic button pressing in order reduce shoulder-surfing. ColorPIN [48] adds a color element to each number in the PIN to increase security and reduce shoulder-surfing. The Phone Lock [49] uses a spinning wheel like one would typically find on combination locks instead of buttons to reduce shoulder-surfing. All of these schemes have roughly the same password space as traditional PIN.

Android offers a graphical cued-recall authentication option typically referred to as Pattern Unlock. Users are presented with a 3×3 grid of dots (larger grids are also possible) and asked to create a password by connecting the dots with straight lines that can be contained inside the grid. Some Android devices provide “security ratings” for different authentication methods, and they rate Pattern Unlock above PIN in terms of security, but below alphanumeric. Passwords made using this scheme are predictable and prone to hotspots– a small subset of Android unlock patterns are used by a large portion of users [50] and most users tend to use the same heuristic rules to design their passwords [51].

Pattern Unlock and other schemes built on the same dot-connecting principle (for example TinyLock [52]) offer only 389,112 possible passwords using a 3×3 grid [52].

In 2016, a Pew survey [20] found that 25% of smartphone owners use a PIN, with alphanumeric passwords at 9% and Google's Pattern Unlock at 9%. Fingerprint authentication accounted for 23% of respondents, and is the fastest growing category, however all biometric schemes still require a fallback knowledge-based scheme such as PIN. Among graphical schemes, only Pattern Unlock holds a meaningful share of the market. A number of other graphical authentication methods such as LG's Knock Code, RealUser's PassFace, and Microsoft's Picture Password have failed to capture a significant market share for various reasons.

Multi-Dimensional Schemes

A key distinction between MAPS and traditional authentication is that information from different dimensions is chosen in a single action. PicassoPass [53], for example, asks users to pick information from five different layers (color, image, letter, location, and shape). During authentication, the layers are superimposed over each other and users must touch their chosen pieces of information. Because the user picks items from just one layer at a time, with the other layers fundamentally present as a distraction for the attacker, PicassoPass is not multi-dimensional.

One example of a partial existing MAPS is ColorPIN [48], a PIN-based scheme where three randomly generated, differently colored letters are placed under each digit. Users must remember both the desired digits and their respective colors, then enter the letter that is generated under the correct digit that also bears the correct color. One key difference between ColorPIN and a more direct MAPS is that the input area is still single-dimensional: a keyboard bearing only letters. Although the memory task and stored password are multi-dimensional, user input is still single-dimensional.

Conversely, schemes like SwiPIN [47] utilize multi-dimensional input without multi-dimensional memory or security. The user is tasked to remember a standard 4-digit PIN. During input, digits are assigned to a section of the screen and a gesture direction. Users input the PIN by tapping the correct screen section and swiping in the gesture direction–two dimensions. Users are still recalling a single-dimensional piece of information, the digits in the PIN.

Multi-modal authentication, such as [54, 55, 56], can utilize various forms of feedback such as haptic, audio, or tactile in order to convey or receive some information used in authentication. Bianchi *et al.* [54] uses haptic or audio feedback to send cues to the user that prompt an action. The user must count the number of cues and match the count against their remembered password. A similar mechanism in the real world is unlocking an unlabeled combination lock, using only the clicking of the lock as a guidance for the finding the correct positions. Multi-modal authentication can be multi-dimensional, and indeed Bianchi's ColorLock [54] is multi-dimensional, using color and hold time as its two dimensions, with vibration or audio cues to determine the integer length of a hold.

While multi-modal authentication can also be multi-dimensional, this chapter's introduction to MAPS will focus on a single-modal scheme, using only the touch screen. Multi-dimensionality is often an incidental result of multi-modal authentication, not the primary focus.

2.4 Chess Based MAPS (CMAPS)

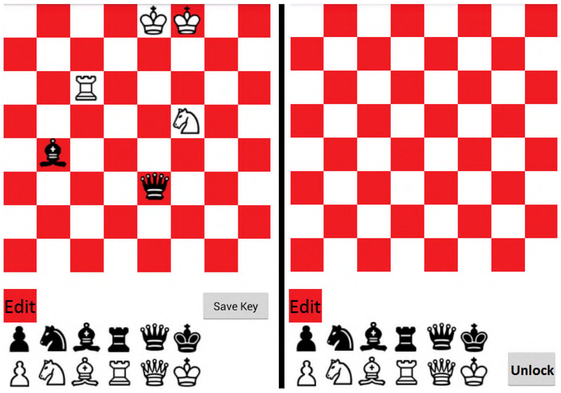


Figure 1: Screenshots of the CMAPS Implementation (An example CMAPS password during setup (left), The unlock page presented to the user before password entry (right))

Figure 1 shows screenshots of Chess Based MAPS (CMAPS), developed for the Android operating system. CMAPS is developed as a proof-of-concept to demonstrate the viability of MAPS. The selection box in the bottom left hand corner shows available piece and color options. Users place chess pieces on the board using either a click-and-drag (more accurately, a touch-and-drag) gesture from the selection box to the desired location, or one tap to select the piece from the selection box and another to place it on the board. Placing 4 pieces on the board can be accomplished by 4 click-and-drag gestures or by a minimum of 5 taps (one to select, and 4 to place, if the piece being placed is the same each time), up to a maximum of 8 taps (if each piece being placed requires a new selection). For simplicity, we will only consider click-and-drag gestures unless otherwise specified. A click-and-drag gesture is roughly equivalent to a gesture connecting two dots in Pattern Unlock, and slightly slower than a single tap as in PIN.

For typographical mistakes, the “Edit” button above the selection box allows a user to empty a tile by tapping the edit button and tapping the desired tile or tiles. The edit button can be considered placing a blank tile. Similarly, the user can overwrite a tile with a different piece by placing the new piece over the old one.

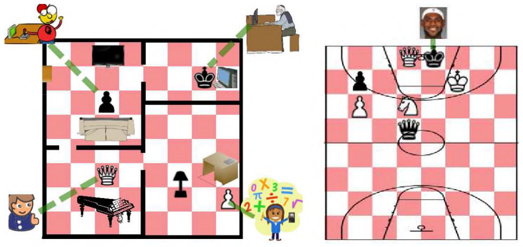
During setup, the user sets a formation of chess pieces. To authenticate later, the user must recreate that formation exactly. The length of a CMAPS password is equal to the number of pieces used in the formation. Each piece placement has 4 dimensions: color (black or white), piece type (king, queen, rook, bishop, knight, or pawn), row (1-8), and column (a-h). Placing a piece on the board fuses all 4 of these dimensions in a single click-and-drag gesture; the user does not select color or row independently, but chooses all 4 dimensions simultaneously when placing a piece on the board. Thus CMAPS fuses information from 4 dimensions into a single gesture or action.

The design of CMAPS does not require any knowledge of chess, allowing CMAPS to be used by anyone. Pieces can be placed on the board in any location and in any quantity, including illegal formations in chess like boards with three kings or pawns in the first row. However, if a user knows how to play chess, they may use certain chess rules or formations in password creation. For example, the user may make a password based on one piece attacking another. The following hypothesis is made based on the design of CMAPS.

H1: Knowledge of chess will improve the memorability of CMAPS. Users who have knowledge of chess will be more likely to remember their CMAPS passwords because they will utilize the rules of chess to assist in forming and memorizing their passwords. H1 is addressed in Section 2.7.5

2.4.1 Graphical Hints

Some users may use patterns or familiar memories to improve the memorability of MAPS. These patterns will be referred to hereon as *graphical hints*. In the user study, some participants were asked to design graphical hints for their CMAPS passwords. The CMAPS implementation does not store those hints– they are kept in memory only– but some users were asked to explain the graphical hints they designed at the end of the experiment.



(a) A family in their home (b) A basketball game

Figure 2: Example Graphical Hints

Figure 2 shows some example graphical hints that were presented to participants in the user study for demonstration purposes. Figure 2(a) shows a home layout, with different member of the family in each room. Location is determined based on the home layout, gender corresponds to color, and the piece type corresponds to age. In Figure 2(b), the chess formation represents two basketball teams playing on a court. The two teams are represented with different colors, and piece type is determined by the player's position. Section 2.7.9 discusses some example hints that participants made during the user study.

Unlike displayed hints used in cued-recall systems such as Windows Picture Password, graphical hints stored in the user's memory will not make the scheme more vulnerable to guessing attacks based on image analysis. Since neither the system nor the attacker has any knowledge of the hint, there is no way to use the hint to improve guessing accuracy, however the mental image of the hint may still have a positive impact on memorability.

Compared to a user generating a password without hints, a hints user will probably chose a more diverse selection of pieces (to represent different elements in the hint), and a more diverse selection of locations (since locations are based on the hint, not just on the board). Hopefully, hints users will pick arbitrary patterns versus predictable patterns. One goal of introducing hints to participants is to mitigate basic shape and pattern drawing that is typical for graphical schemes, such as the behavior found in free-form gesture schemes [41]. Participants in free-form drawing schemes often draw symmetrical geometric shapes like stars, circles, and squares. Another goal of introducing hints is to reduce the popularity of corners– Pattern Unlock demonstrates that corners can be very popular when a grid is used [50].

The following hypotheses are generated for graphical hints.

H2: Presenting users with the idea of graphical hints before password creation will reduce the popularity of hotspots compared to users that were not introduced to graphical hints. Non-hints users may have hotspots particularly around corner tiles. Hypothesis H2 is addressed in Section 2.7.7.

The term "hotspots" refers to frequently selected spots in graphical passwords which enable attackers to run more efficient guessing attacks [43]. Hotspots can also occur in piece type and color if one piece type or color is selected more often than others. H2 refers to hotspots in location, piece type, and color.

H3: Presenting users with the idea of graphical hints before password creation will improve memorability. Hypothesis H3 is addressed in Section 2.7.5.

2.5 Security Strength of MAPS

In this section, the security strength of MAPS and CMAPS is discussed relative to the password space, i.e., the number of possible passwords.

2.5.1 Security Strength of MAPS

Ideally, all dimensions used in a MAPS will be *independent*, that is a choice in one dimension does not limit choices in any other dimension, and does not limit future choices. In CMAPS for example, choosing color does not limit available piece types, choosing column does not limit choice of rows, and so forth. However, CMAPS is still not fully independent, because placing a piece occupies that tile and therefore reduces the options available for the next piece placement. The first piece will have $8 * 8 = 64$ options for locations, the second will have 63, and so forth.

For a MAPS where all dimensions are wholly independent, the number of possible passwords can be derived as follows.

Proposition 1. *For a MAPS with n independent dimensions and m_i possible choices in the ith ($1 \leq i \leq n$) dimension, the number of possible passwords of length l is $\prod_{i=1}^{n} (m_i)^l$.*

The length l can also be considered as the number of times information is fused together from the different dimensions in a single action. Each instance of information fusion can have $\prod_{i=1}^{n} m_i$ possible combinations because each dimension is independent and thus goes into the password space multiplicatively.

Proposition 1, leads to the following corollary.

Corollary 1.1. *The size of the password space generated by adding t possible choices to an existing dimension is no greater than the size of the password space generated by adding a new dimension with t possible choices when $t \geq 2$, and the number of existing choices in each dimension is already greater than or equal to two.*

When $t = 2$ and the dimension to add t possible choices has only two possible choices prior to addition, the resulting password space of both methods is the same.

The proof of Corollary 1.1 can be found at the end of this section.

When t is small, the difference between between the size of the password spaces is also small, but as t increases the ratio between the size of the password space generated by adding a dimension with t choices and adding t choices to an existing dimension grows exponentially with l.

Corollary 1.1 demonstrates the advantage of MAPS over traditional single-dimensional schemes from a security standpoint. Fusing information from multiple dimensions can generate a significantly larger password space than adding choices to a single-dimensional password.

2.5.2 Security Strength of CMAPS

Proposition 2. *With l gestures, CMAPS with a classical chess board consisting of eight rows and eight columns can generate $2^l 6^l \binom{64}{l}$ possible passwords.*

The proof of Proposition 2 can be found at the end of this section.

The results of Proposition 2 are compared against a 4 digit PIN approach and a traditional alphanumeric scheme with 62 options per character (letters and numbers, case-sensitive). Google's Pattern Unlock scheme can support a total of 389,112 passwords on a 3×3 grid [52], approximately the same as 2 gesture CMAPS (290,304). Windows Picture Password supports approximately 2^{30} passwords (exceeded by CMAPS with 4 gestures), though research suggests many passwords can be cracked within 2^{19} attempts [45] (exceeded by CMAPS with 3 gestures).

To make a fair comparison, the password space will be compared against the number of gestures required in different schemes. One gesture selects a digit in a PIN; this may be a tap gesture, like in a traditional PIN scheme, or a swipe gesture in more advanced methods such as SwiPin [47]. We will assume that a single tap can select any character in an alphanumeric password, though in practice many smaller devices require the user to switch to the numeric keyboard in order to enter numbers or to press shift to type a capital letter, which may require an additional tap. In CMAPS, one swiping click-and-drag gesture can place a game piece on its desired tile. A series of two taps, one to select the piece and one to place it, can also be used. The latter approach is likely to be done with two fingers, so both approaches can have potential time benefits for different users. We will assume that a tap, click-and-drag, and two-finger tap have roughly equal input times and can all be considered as one gesture for purposes of making comparisons.

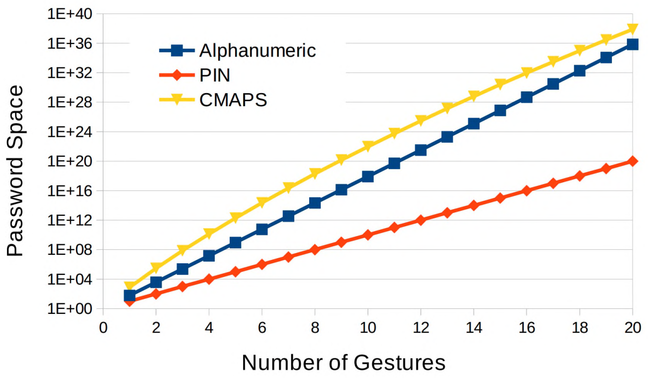


Figure 3: Password Space Between One and Twenty Gestures

Figure 3 compares the security strength of CMAPS, PIN, and alphanumeric passwords with 62 options per character (26 letters, case sensitive, 10 digits). When the number of gestures is less than 20, CMAPS generates significantly more passwords than alphanumeric or PIN approaches. Most passwords used for high security applications such as banking are between 8 and 20 characters long. Because CMAPS has a dependent dimension that offers gradually fewer choices as password length increases, the alphanumeric approach generates more passwords when the number of gestures is larger than 24, but CMAPS still generates significantly more passwords than the PIN based approach.

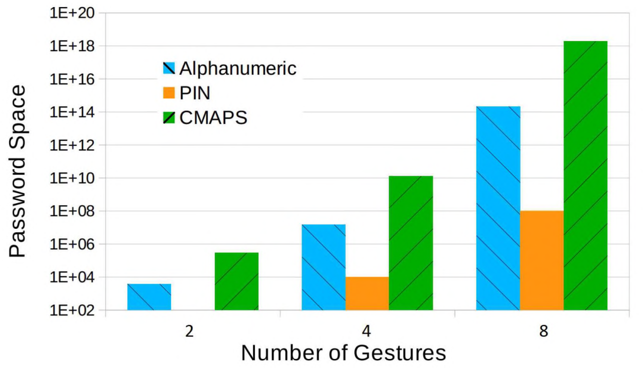


Figure 4: Password Space at Two, Four, and Eight Gestures

CMAPS particularly excels at low gesture counts. Figure 4 shows that two-gesture, four-gesture, and eight-gesture CMAPS passwords can generate about 2900, 1.3×10^6, and 1.9×10^{10} times more passwords than the PIN-based approach respectively and about 75, 890, and 8,700 times more passwords than the alphanumeric approach respectively. A 4 gesture CMAPS password is about 131 times more secure than an 8 digit PIN, recommended by many cybersecurity firms as a minimum for device locking, and a 6 gesture CMAPS password is slightly more secure than an 8 character alphanumeric password, the standard cutoff length for secure applications like banking.

Proof of Corollary 1.1

The number of choices in each dimension m_i has to be greater than or equal to two. If there is only one choice in a dimension, then the dimension has no influence on the password space and it can be removed.

Proof. Let us consider adding t choices to the jth dimension. We denote the number of possible choices in the jth dimension as m_j, where $m_j \geq 2$. Then the size of the password space, denoted as S_1, that results from adding t choices to the jth dimension is

$$S_1 = (\prod_{i=1}^{j-1} m_i^l)(m_j + t)^l(\prod_{i=j+1}^{n} m_i^l) \tag{2.1}$$

where n is the number of dimensions and l is the length or number of times information is fused.

The size of the password space generated by adding another dimension of t choices, denoted as S_2, can be derived according to Proposition 1 as follows.

$$S_2 = (\prod_{i=1}^{j-1} m_i^l)m_j^l(\prod_{i=j+1}^{n} m_i^l)t^l \tag{2.2}$$

where n is again the number of dimensions (before adding t), and l is the length or number of times information is fused.

Since $t \geq 2$ and $m_j \geq 2$, we can derive

$$(t-1)(m_j-1) \geq 1 \ . \tag{2.3}$$

After simplification on Inequality 2.3, we can derive

$$tm_j \geq t + m_j \quad . \tag{2.4}$$

Combining Equation 2.1 and Inequality 2.4, we can derive as follows.

$$S_1 \quad \leq \quad (\prod_{i=1}^{j-1} m_i^l) m_j^l t^l (\prod_{i=j+1}^{n} m_i^l) = S_2 \tag{2.5}$$

We have equality in 2.3, only if $t = 2$ and $m_j = 2$. So the two methods generate password space of the same size only when $t = 2$ and $m_j = 2$, otherwise S_2 is greater. □

Proof of Proposition 2

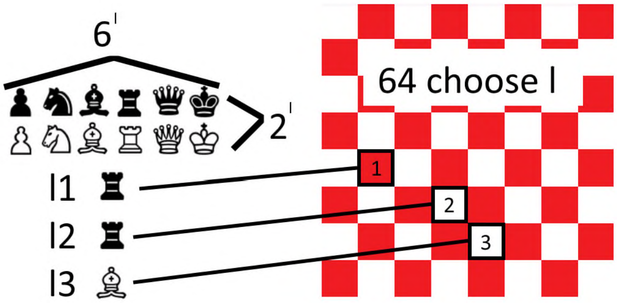


Figure 5: Visualization of the Password Space of CMAPS

Figure 5 demonstrates the choices made in a CMAPS password of length l. One gesture can select a single game piece and place it on the board. We can consider this a single instance of information fusion in a MAPS. Three types of information are selected: (1) location, split into row and column, (2) color, and (3) piece type. The latter two types of information are selected in a straightforward manner. Since there are two choices for color, and six for piece type, and these choices are independent of each other, the password space is $2^l * 6^l$, where l is the length of the password.

Location is accounted for by choosing l tiles from the classic 8 by 8 chess board, which can be expressed simply as $\binom{64}{l}$. Combinations are used because the choice of tiles matters, but the order in which tiles are chosen does not, e.g., if there are two white bishops on tiles a1 and b1, it would not matter which white bishop was placed first. Since this choice is independent of piece color or type, and because duplicate orders were already accounted for by using combinations, the overall password space of CMAPS with length l is $2^l * 6^l * \binom{64}{l}$.

2.6 Usability Analysis

This section discusses the usability of CMAPS in terms of usability requirements and number of gestures used for authentication. Timing information and a survey of user perceptions of usability are presented in Section 2.7.

Because ease of use and speed of use are almost universally recognized as the most important factors on mobile, this section focuses on assessing usability via ease of use and speed of use by examining the number of gestures needed for authentication.

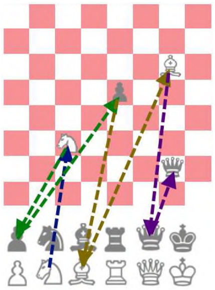


Figure 6: A CMAPS Password Completed in One Long Gesture (The gesture starts from the white knight. For visual clarity, different colors are used to draw segments that place different game pieces.)

Users can place pieces on the board in CMAPS by drawing a line gesture between the desired piece and the desired destination on the game board. A CMAPS password with l pieces requires l click-and-drag type line gestures to complete. Like Pattern Unlock, CMAPS could also be finished in a single long gesture, as demonstrated in Figure 6. CMAPS could also be completed with two fingers, placing 2 pieces at a time to increase speed, though this may be difficult for most users to do accurately.

Table I: Number of Gestures Required for Different Password Spaces

Password Space	$2.2 * 10^{14}$	$1.2 * 10^{21}$	$1.3 * 10^{30}$
PIN	15	22	31
Alphanumeric	8	12	17
CMAPS	6	10	15

Table I compares how many gestures are required to finish a password with a given security strength. The first column represents the commonly accepted bare-minimum security standard afforded by an 8-character alphanumeric password. The second and third columns correspond to 70 bits (2^{70}) and 100 bits (2^{100}), representing a "strong" and "very strong" password respectively. It is clear from the table that CMAPS requires fewer gesture to achieve the same security strength, particularly in the range where most users tend to create passwords. To make the equivalent of an 8-character alphanumeric password, CMAPS requires just six gestures, a savings of 25%. The relative benefit of CMAPS compared to alphanumeric passwords decreases with higher levels of security strength, but passwords meeting those security levels are not typically used on mobile devices.

CMAPS demonstrates an important point: because a MAPS uses dimensions that apply towards security strength multiplicatively, a MAPS will typically perform much better than a single-dimensional scheme at shorter password lengths. Since users prefer to use short passwords, using multiple dimensions can be effective in improving overall security strength.

2.7 User Study

2.7.1 Overview

A user study was conducted to evaluate the memorability and usability of CMAPS. The study consists of two controlled laboratory sessions separated by one week and up to two email responses in the interim.

Demographic data about participants is collected in the first session, then CMAPS is introduced and users are instructed on how to use CMAPS. Before leaving the laboratory, users are asked to generate a CMAPS password on a smartphone which is kept in the lab. Users must recall the password successfully one more time after generating it before they leave the laboratory. The first session takes approximately 20 minutes total.

To simulate regular use of passwords as in previous research [57], an email is sent to participants after two days and again after four days. The email contains a link to an online emulator of CMAPS. The emulator behaves in the same way as the smartphone application, but can be used on any device with web browser access, including a traditional computer. Using the emulator between the first and last session is not mandatory because (1) email response rates may be low since email communication is not always reliable [58], and (2) the following hypothesis is formed for the reminder emails.

H4: Participants who use the reminders will have better memorability after one week than participants who do not. Hypothesis H4 is addressed in Section 2.7.5

One week after the first session, participants return to the laboratory for the second and final session. Participants recall their passwords on the same device they used to create them in the first session. Participants are given at most five minutes to recall their password, with unlimited attempts on the device. At the end of the session, participants fill out a survey comparing CMAPS to their favorite mobile authentication scheme. The second session takes approximately 15 minutes total.

2.7.2 Apparatus

CMAPS was implemented on a Samsung Galaxy S4 smartphone running Jelly Bean (version 4.2) of Android. Two screenshots of the application can be seen in Figure 1.

2.7.3 Conditions

Users are randomly assigned to one of four conditions in order evaluate the memorability and usability of CMAPS at different levels of security strength.

(1) **2g:** Passwords must be generated with exactly two pieces.

(2) **8g:** Passwords must be generated with exactly eight pieces.

(3) **8+g:** Participants were asked to generate CMAPS passwords with “at least eight” pieces.

(4) **8+gh:** Participants were asked to generate CMAPS passwords with “at least eight” pieces. Before generating CMAPS passwords in this condition, participants were shown examples of graphical hints as in Figure 2. Participants assigned to this condition were encouraged to generate their own graphical hints and create their CMAPS passwords based on graphical hints. Graphical hints are stored only in the user’s minds, though some users were asked to describe their hints at the end of the second session.

2.7.4 Participants

Participants were recruited by distributing fliers and leaflet style advertisements, and compensated $10 if they completed both sessions. A total of 66 participants were recruited and 54 completed both sessions, a dropout rate of 18%. Of the 12 dropouts, 6 indicated a schedule conflict and the remainder did not respond– there was no significant difference in the dropouts by condition ($\chi^2 = 5.3$, $p = 0.15$).

Of the 54 participants who completed the experiment, 26 were female, ranging in age from 18-71. Most participants opted to report only their age range, with 26 participants aged 21 to 25 and 20 participants aged 20 and under. Participants were asked “Are you skilled at using Smartphones or mobile devices.” On a scale from Strongly Disagree (1) to Strongly Agree (5), participants rated their skill at using smartphones an average of 4.07, and 81% of participants rated their skill 4 or higher. An “In the Wild” experiment utilizing CMAPS is planned in the future to examine the impact of age and smartphone skill on performance in CMAPS.

2.7.5 Memorability

Table II: Recall Rates of CMAPS Passwords.

Conditions	Participants	Recall	Recall Rate
2g	8	8	100%
8g	18	18	100%
$8+g$	13	13	100%
$8+gh$	15	13	87%

Table II shows the recall rates of CMAPS passwords in each condition after one week. Recall rates did not vary significantly by condition ($\chi^2 = 5.4$, $p = 0.15$), indicating that CMAPS is highly memorable even when using a longer password. Because memorability rates are so high, a planned future extension is to increase the memorization period of CMAPS and remove reminders. A clearer picture of just how long CMAPS can remain memorable, and how memorable CMAPS is at high security strength, is desirable. A future study is also planned with a PIN control group to determine if CMAPS is not only high memorable, but more memorable than its main competitor in secure authentication schemes.

Comparing 8+g and 8+gh shows no significant difference in memorability ($\chi^2 = 5.4$, $p = 0.15$), violating the expectation from hypothesis H3. Questioning of participants during the second session revealed that many participants were using hints even when not instructed to do so. Exact numbers of participants in other conditions using graphical hints were not obtained and further investigation of the impact of user-generated hints without instruction will be left to future work. Notably, many participants who did not receive hints instructions drew shapes. It is possible that the majority of users do not need any instruction in hints and will use them natively, though the caliber of hints may improve with instruction.

Daily use also did not appear to have an impact on memorability. The majority of participants (87%) responded to at least one reminder email by recalling their password via the emulator. Only 33% participated in both reminders. Both participants who failed to recall their passwords responded to only one email. A Chi-squared test on the four conditions (responded to the first email, responded to the second, responded to both, responded to none) reveals no significance ($\chi^2 = 1.68$, $p = 0.64$), violating the expectation of H4. CMAPS passwords appear to remain memorable after one week even without use. A future “In the Wild” experiment may further examine the impact of irregular use in CMAPS.

Impact of Chess Knowledge

Participants indicated whether or not they could play chess by answering yes or no in the demographic survey: 81% answered yes. Of the two users who forgot their passwords, one knew how to play chess and one did not. Chess knowledge does not seem to have an impact on memorability ($\chi^2 = .26$, $p = 0.61$), contrary to the assumption in H1, but there is not sufficient data to make a concrete statement. The current data implies that CMAPS passwords are memorable even to people with no knowledge of chess, and future work will further investigate if knowledge of the game used in a game-based password like CMAPS will improve performance metrics like memorability.

2.7.6 Usability

The usability of CMAPS is evaluated using timing data from the second session and survey data collected from users at the end of the second session. The expectation for CMAPS usability performance is straightforward.

H5: Participants who use more pieces will have longer entry times, make more errors, and have a longer entry time for single attempts.

Password Entry Time

Timing information from each authentication attempt was recorded by the application. Data from participants who were observed to be distracted midway through an authentication attempt was excluded. Roughly 5% of timing data was omitted in this manner. Participants were not instructed to optimize speed when entering passwords. The time clock begins when the screen containing the game board is rendered, and ends when the user hits the unlock button after putting the correct configuration on the board.

Table III: CMAPS Mean Password Entry Time

	Time (seconds)			
	2g	8g	8+g	8+gh
Total	10	21	23	25
First Correct	10	14	14	20

Table III shows the timing results for CMAPS in each condition. Mean total authentication time, including unsuccessful attempts and time spent thinking between attempts, was 10, 21, 23, and 25 seconds for 2g, 8g, 8+g, and 8+gh respectively. A Kruskal-Wallis test using the timing data from the four conditions indicates significance ($H = 10.998, p < 0.0117$). Pairwise Mann-Whitney comparisons between the categories show significant differences between 2g and 8g ($Z = 2.69, p = .007$), and between 2g and 8+gh ($Z = 3.01, p = .002$). Despite 8+g being slower than both 2g and 8g on average, there was no significant difference between 2g and 8+g ($Z = 1.27, p = .20$), which may be attributed to several outliers in 8+g.

The total password entry time for a CMAPS is comparable to other graphical schemes such as Deja Vu (31-36s) [35], CDS (20s) [59], Story (23s), and Draw a Secret (5-12s) [60]. [1]

[1] Deja Vu, CDS, and Story use a mouse for input. Draw a Secret uses a PDA.

Table IV: Pairwise Testing on Password Entry Time (Single Correct Attempt)

Comparison	Z score	P-value
2g vs 8g	1.7	.09
2g vs 8+g	1.04	.30
2g vs 8+gh	2.36	.01
8g vs 8+g	.11	.91
8g vs 8+gh	-1.77	.08
8+g vs 8+gh	-2.03	.04

Time spent on the first successful attempt is calculated as the time from when the screen with the chess board loads to correct authentication, or from the latest unsuccessful authentication to the first successful authentication, whichever is shorter. Thus, if a user makes a mistake but corrects it before hitting the unlock button, this time will include any thinking time or time spent making those corrections. Participants required a mean of 10, 14, 14, and 20 seconds for the first successful authentication attempts in 2g, 8g, 8+g, and 8+gh conditions respectively. A Kruskal-Wallis test using timing data from the four conditions indicates significance ($H = 8.08, p < 0.044$). Table IV shows pairwise comparisons with a two tailed Mann-Whitney test. Both 2g and 8+g show a significant difference with 8+gh, however 8g does not, which may again be attributed to outliers.

The password entry time for a single CMAPS password entry is comparable to other schemes such as CDS (14s) [59], Story (9s), Xside (3-4s) [40], SwiPIN (4-5s) [47], Color-PIN (14s) [48], and TinyLock (2-4s) [52]. [2] There are more 2 gesture CMAPS passwords than there are 1-5 digit PIN or SwiPin passwords combined, and almost as many 2 gesture CMAPS passwords as there are total passwords in Android's Pattern Unlock or TinyLock with a 3×3 grid. The fastest 4 users in 2g required a mean of 7 seconds to authenticate, and the fastest 4 users in the 8+ conditions required 8 seconds, indicating CMAPS can reach competitive authentication times with some practice or user skill. In ColorPIN [48],

[2] SwiPin and TinyLock timings are measured from first touch to last touch rather than from application load to last touch. SwiPin uses total time including preparation time. None of these schemes require tapping an unlock button to indicate that the attempt is finished. CDS and Story are unclear in their methodology. CDS and Story use a mouse for input.

it was found that entry times can be reduced from 14s to 3.5s after just five practice sessions. CMAPS was tested with just one authentication session. Future work will investigate if users can reduce times to 2-5 seconds with light practice and investigate what sorts of authentication times CMAPS can expect when deployed "In the Wild" against traditional authentication schemes.

Participants in 2g, 8g, 8+g, and 8+gh required 1, 1.3, 1.4, and 1.4 average attempts per successful authentication for each condition respectively. A Kruskal-Wallis test on the attempts required shows no significance ($H = 1.144, p = 0.767$), contrary to the expectation from H5.

Usability Survey

At the end of the second session, participants complete a usability survey. The survey asks the following questions and asks users to rate their answer on a scale from Strongly Disagree (1) to Strongly Agree (5).

(1) The authentication scheme in the study is convenient.

(2) The speed of entering a password with the authentication scheme in study is fast.

After answering these questions for CMAPS, participants are asked to pick their favorite existing mobile authentication scheme and answer the questions for that scheme as well.

Table V: Average Usability Rating of CMAPS and Other Schemes.

Scheme	Ratings	Convenience	Speed
CMAPS-2g	8	4.5	3.88
CMAPS-8g	18	4	3.61
CMAPS-8+g	13	4.08	3.54
CMAPS-8+gh	15	3.67	3.6
4-digit PIN	29	4.48	4.52
Google Pattern	7	4	4.29
Fingerprint	11	4.46	4.64

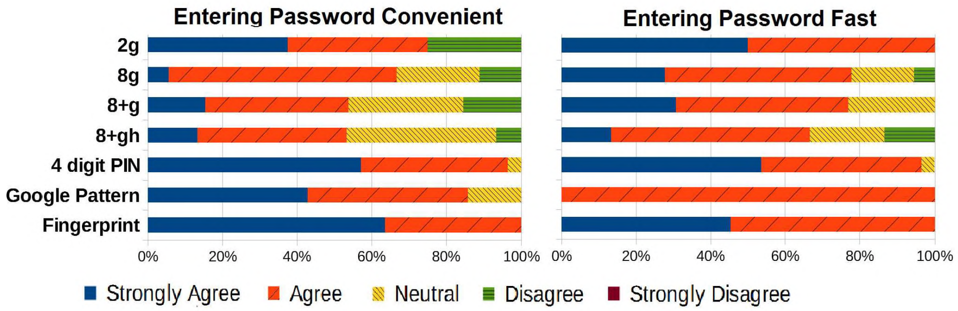


Figure 7: Survey Results

Figure 7 shows the usability survey results. The average usability rating results are shown in Table V. Not included in the table are five participants who did not select a favorite other mobile security scheme and two participants who chose facial recognition and Windows Picture Password.

Table VI: Statistical Analysis on Usability Data for CMAPS.

	Convenience		Speed	
	χ^2	p	χ^2	p
Omnibus	3.36	.399	2.56	.465
Category	Pairwise Test Result			
2g vs 8g	2.10	.147	.181	.671
2g vs 8+g	2.15	.142	.940	.332
2g vs 8+gh	3.41	.065	1.02	.310
8g vs 8+g	.003	.955	.523	.470
8g vs 8+gh	3.41	.065	.609	.465
8+g vs 8+gh	.509	.476	.068	.795

Responses from the survey were further sorted as either unsatisfied (1-3) or satisfied (4-5). Pairwise testing was conducted using these binary categories. Table VI shows that CMAPS has no significant difference in usability between the four conditions.

Table VII: Statistical Analysis, CMAPS vs Other Schemes. (PIN, Pattern Unlock, and Fingerprint Scanner are abbreviated for brevity. Two categories could not be tested because they had perfect ratings. Significant p values are bolded.)

	Convenience		Speed	
	χ^2	p	χ^2	p
2g vs PIN	.284	.594	.284	.594
8g vs PIN	.198	.656	**7.83**	**.005**
8+g vs PIN	**4.01**	**.045**	**11.8**	**.001**
8+gh vs PIN	**7.50**	**.006**	**12.4**	**.001**
2g vs Pattern	NA	NA	.268	.605
8g vs Pattern	1.85	.174	.907	.341
8+g vs Pattern	.359	.549	2.03	.154
8+gh vs Pattern	3.02	.082	2.16	.141
2g vs Finger	NA	NA	3.07	.080
8g vs Finger	2.84	.092	**4.62**	**.032**
8+g vs Finger	2.90	.089	**6.77**	**.009**
8+gh vs Finger	2.90	.089	**7.02**	**.008**

CMAPS is further compared against existing authentication schemes selected by participants. Since each participant is only asked about CMAPS and the authentication scheme they prefer, only pairwise testing is used for analysis. Table VII shows the results for different CMAPS conditions against existing schemes. In terms of convenience, CMAPS 8g was not significantly different from 4 digit PIN, Pattern Unlock, or fingerprint schemes. CMAPS in 8+g and 8+gh was rated less convenient than 4 digit PIN. Speedwise, CMAPS was rated lower than 4 digit PIN and fingerprint for more than 2g, but was not significantly different from Pattern Unlock.

From the survey, 2g CMAPS, which exceeds the security strength of existing mobile unlock schemes, appears to be an acceptable alternative to schemes such as 4 digit PIN. CMAPS with 8g, which exceeds even the requirements of high security applications like banking, is still comparable with existing mobile authentication schemes in terms of usability. Usability survey ratings seem to support hypothesis H5, more gestures reduces the usability of CMAPS.

2.7.7 Hotspots

Frequently selected portions of information in a password scheme, often called hotspots [43], allow attackers to launch more efficient guessing attacks, sometimes called dictionary attacks. Dependence on pictures and images can leave graphical passwords vulnerable to hotspots [61]. In this section, Shannon's entropy [62], an information-theoretical measure of uncertainty, is used to evaluate hotspots. Entropy E is defined as follows.

$$E = -\sum_{i} p_i \log_2 p_i \tag{2.6}$$

where p_i denotes the probability of selecting the ith choice. In CMAPS, hotspots can occur in tiles on the board, in certain piece types, and in color.

Hotspots in Tile Selection

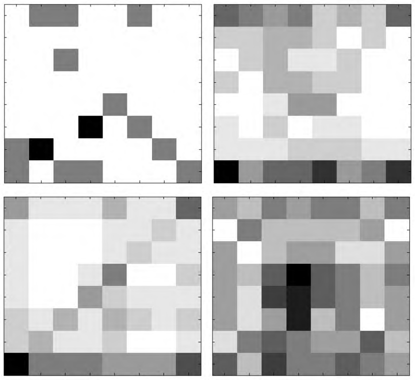


Figure 8: Popularity of Tiles (The gray level of each tile indicates the popularity of each tile. The most popular tile and the least popular tile are colored black and white respectively. From left to right: (top row) 2g, 8g, (bottom row) 8+g, 8+gh.)

Figure 8 shows the popularity of different tiles on the chess board. Some tiles, particularly corner tiles, are chosen more often than others. Assuming a uniform tile distribution, entropy $E^{tile}_{uniform} = 6.00$ according to Equation 2.6 as $p_i = \frac{1}{64}$, $1 \leq i \leq 64$. The entropy of tile selection in Condition C, denoted as E^{tile}_{C}, can be calculated in the same way. Using popularity data from Figure 8, E^{tile}_{2g}, E^{tile}_{8g}, E^{tile}_{8+g}, and E^{tile}_{8+gh} are 3.75, 5.25, 5.26, and

5.76 bits respectively. Hotspots in tile selection can be considered to reduce uncertainty by 2.25, 0.75, 0.74, and 0.24 bits in 2g, 8g, 8+g, and 8+gh conditions respectively.

As the number of pieces used increase, the popularity of hotspots decreases, since more tiles are used. Graphical hints also appear to reduce to reduce the popularity, upholding the assumption from H2. Overall, the entropy of all conditions above 2g is close to the maximum entropy of 6 bits, indicating that when the number of gestures is 8 or greater, the hotspot effect is largely negligible and each tile is roughly evenly popular.

Hotspots in Piece Selection

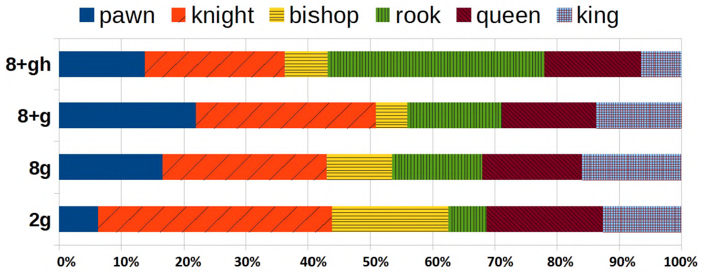


Figure 9: Popularity of Different Piece Types

Figure 9 shows the popularity distribution of different piece types. Ideally each piece will be selected 17% of the time, since there are six choices ($100/6 = 17\%$). The data shows that pawns, rooks, queens, and kings were placed 18%, 14%, 16%, and 15% of the time respectively. Knights were placed 28% of the time, while bishops were placed only 9% of the time.

Assuming a uniform distribution for piece type selection, we can calculate entropy of piece type $E^{type}_{uniform} = 2.59$ bits according to Equation 2.6 as $p_i = \frac{1}{6}$, $1 \leq i \leq 6$. Similarly we can calculate the entropy E^{type}_{C}, denoted as the entropy of piece type selection in Condition C. According to the data from Figure 9, E^{type}_{2g}, E^{type}_{8g}, E^{type}_{8+g}, and E^{type}_{8+gh} are 2.31, 2.52, 2.44, and 2.34 bits respectively. The results show that there is a small hotspot effect in piece type, all four categories are close to the maximum entropy of 2.59 bits.

The results show the number of gestures does not have a predictable effect on hotspots in piece type.

Graphical hints worsened the distribution of piece type. A total of 204 pieces were placed in passwords based on hints. Rooks were placed 71 times (35%) while bishops and kings were placed only 14 times (7%) and 13 times (6%) respectively. It's likely that rooks were used more because rooks generally move straight horizontally and vertically, so users were inclined to create lines with these pieces. One user built a house using 22 rooks, while another used 16 rooks to build tiers of a pyramid, together accounting for almost half of the overall rook usage by hints users. In both instances the users added other pieces to their password, so the expanded use of rooks by hints users is not necessarily problematic. Bishops and kings may not be attractive for pattern building, but it should be noted that despite their unpopularity overall, kings appear in 40% of all hint based passwords. It's likely that kings were usually placed individually because they are important piece in chess, and each color typically has just one king. While users do avoid pieces like the king for pattern building, they are still regularly included in passwords, making a brute force algorithm based on piece distribution alone quite challenging. If enough user data can be collected, developing an intelligent brute force algorithm for CMAPS based on user choice is one potential future direction of this work.

2.7.8 User Choice in CMAPS Passwords

Black pieces were significantly more popular than white pieces. For non-hints users, there were nearly twice as many black pieces placed as white (184 vs 94). Graphical hints reduced the gap slightly (124 vs 80). Black may simply be the more appealing color choice. Non-hints passwords were 48% monotone, with 15 (38%) all black and 4 (10%) all white. Hints passwords fared better, with 3 monotone passwords (19%)– 2 (13%) passwords that were all black and 1 (6%) password that was all white. A brute force algorithm could fare well against CMAPS by checking combinations of black pieces first. This could be mitigated by changing the default colors of the pieces, for example green vs blue, so that neither color is more appealing, or by requiring at least one choice of each color. Various modifications such as switching the colors will be tested in future work as a way to reduce the hotspot effect in different MAPS applications. It is possible that black is more popular in other applications, not just chess, and a different color palette is more universally desirable to reduce the hotspot effect.

From all non-hint categories, 8 out of 39 (20%) of passwords used only one type of piece. These passwords were based exclusively on pawns (8%) and knights (12%). Half of these monopiece passwords (50%) were also monotone. All monopiece passwords come from the 8g and 8+g conditions, accounting for 2 of 18 (11%) and 6 of 13 (46%) of the passwords in these categories respectively. All monopiece passwords in the 8+g category chose to use exactly 8 pieces. It's possible that using 8 gestures in an 8 by 8 board encourages the filling of a single row, column, diagonal. Half of all monopiece passwords (50%) use a single row, column, or diagonal. If CMAPS were to be fully implemented as a commercial authentication scheme, it should disallow passwords that are both monopiece and monotone, and passwords that are monopiece and fill only a single row, column, or diagonal. This would be analogous to alphanumeric and PIN schemes that disallow a password consisting of just 1 character repeated many times. The impact of enforcing password creation policies that force multiple choices in each dimension of MAPS on user perceptions of usability and the overall password creation time will be studied in future research.

Users in the 8+g condition used an average of 9.01 pieces, with a median of 8. The majority of users in this category (84%) chose to use 8 pieces exactly. If a CMAPS implementation were to enforce a minimum number of gestures, it would be wise for a brute force attack to test all combinations at the minimum first. There are $5.3 * 10^9$ possible CMAPS passwords at 8 gestures exactly, so it would still take a long time to brute force only the 8 gesture passwords. Hints greatly reduced the number of participants using the minimum requirement. Users in the 8+gh condition used an average of 13.6 pieces with a median of 12. Only 20% of users in this category chose to use 8 pieces exactly. This, combined with entry time results, supports hypothesis H5. Entry times are indeed worsened when more pieces are used, but only when the difference is considerable, as there is no significant difference between using 2 and 8 pieces.

The results of user password selection may support H2: graphical hints users tend to make passwords which are less monopiece/monotone and have better tile variety, but they tend to use rooks significantly more than non-hints users. The impact of presenting a brief demonstration of graphical hints before password creation when CMAPS is deployed “In the Wild” may be examined in future research.

2.7.9 Graphical Hints Generated by Participants

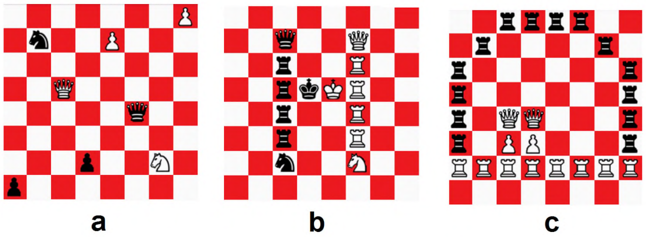


Figure 10: Example Graphical Hints Created by Users.

At the end of the experiment, some 8+gh participants were asked to describe their graphical hints. Several hints generated by users are presented here. Figure 10 shows some example graphical hints created by participants. Password (a) is based on chess. The knights are used as a reference. Each knight is attacking a queen, which is covering a pawn. A pawn of the queen’s color sits in the corner. Password (b) is the letter H, with colors swapped between the two sides. The bulk of the vertical lines are made up of rooks, but the top and bottom of each line is capped with a unique piece, and the horizontal center line is made from kings. Password (c) is a house. The floor or foundation is built from white rooks, and the remainder from black rooks. Two women, their bodies made of pawns and their heads made of queens, sit inside the house. While the password in (c) may appear unusable, that user was able to achieve an entry time of approximately 20 seconds and made no entry mistakes, indicating that even very long CMAPS passwords are potentially usable.

2.8 Discussion

CMAPS has a substantial advantage in raw security over traditional PIN, Pattern Unlock, and even alphanumeric passwords of equivalent length. CMAPS has a large password space and relatively few hotspots, meaning that it should theoretically be very difficult to brute force a CMAPS password. CMAPS is also highly memorable over one week. The usability of CMAPS is roughly on par with existing schemes. User survey indicates that 2g CMAPS is roughly equivalent to PIN, and 8g CMAPS and up are comparably close, although timing data indicates that some practice will be needed before users can achieve PIN-like entry speeds.

Thus CMAPS demonstrates that a MAPS can achieve high security while maintaining high memorability and usability, acting as a solid proof-of-concept for MAPS.

Another aspect where MAPS may have an advantage is in memory interference. One concern is in password expiration, a common policy in corporate environments, which forces users to change their password every few weeks or months. This can create a burden on memorability, or even allow attackers to use information from old passwords to help break new passwords [63], frequently caused by users appending or changing just a small number of digits in the new password. CMAPS can mitigate this problem by adding a *game* dimension. The user's password may be based on Chess initially, but the next password may be based on another game, for example Monopoly. Because the pieces, rules, and board layout of Monopoly are quite different from Chess, the new password will have little relationship to the old one, and it may not suffer from memory interference with the Chess based password. This approach can also be used for managing passwords to different services.

CHAPTER III

SHOULDER-SURFING RESISTANCE

3.1 Outline

This chapter is based on a work presented at the 2017 Conference on Advances in Computer-Human Interactions (IARIA ACHI) [64]. A journal version of this work will appear in the International Journal On Advances in Security, volume 10, circa late December 2017 [65].

Section 3.2 addresses the shortcomings of CMAPS in providing security against common, low-tech observation attacks known as shoulder-surfing, and defines the focus of this chapter as creating a MAPS that resists shoulder-surfing. Section 3.3 describes related works in developing shoulder-surfing resistance, and briefly describes other authors' varied methodologies for testing shoulder-surfing resistance. Section 3.4 introduces the design of PassGame, a novel proof-of-concept challenge-response authentication scheme based on modifying a chess game board to match certain pre-defined rules. Section 3.5 calculates the security strength of PassGame in terms of raw password and effective password space. A user study examining the memorability, usability, and shoulder-surfing resistance of PassGame is presented in Section 3.6. Conclusions and future work are addressed in Section 3.7.

3.2 Expanding MAPS to Reduce Shoulder-Surfing

3.2.1 CMAPS vs Shoulder-Surfing and Smudge Attacks

CMAPS was focused on generating a large password space to reduce brute force attacks, however brute force is a relatively uncommon method of password cracking. A far more common, and significantly less technologically complex method of cracking a password, is simply observing the user entering it over their shoulder, typically referred to as shoulder-surfing. Graphical passwords like CMAPS, because of their visual nature, are significantly easier to observe and therefore easier to shoulder-surf than traditional non-graphical approaches [66, 67].

Another low-tech attack that is frequently used against smartphones and other touch-screen devices is the smudge attack [68], wherein an attacker guesses the password using the smudge pattern left behind on the screen. Pattern Unlock is especially vulnerable to this type of attack [52]. CMAPS may also be exposed since it also relies on click-and-drag gestures, but CMAPS will likely have some resistance to smudge attacks because of over-lapping click-and-drag gestures and the general frequency of use in the piece selection box leading to a less discernible pattern. In general, shoulder-surfing resistance affords smudge attack resistance as well.

Developing a MAPS that is also shoulder-surfing resistant, in turn demonstrating that MAPS can lead to improved security in many ways, is the inspiration for this chapter.

3.2.2 PassGame: Adding Shoulder-Surfing Resistance to MAPS

This chapter of the work is focused on developing a MAPS that is as shoulder-surfing resistant as possible without relying on multi-modal input, biometrics, extra hardware, or anything else besides the touchscreen display. The scheme is called *PassGame*, a reference to Passwords and Games, because it is an authentication scheme again based fundamentally on Chess. Unlike CMAPS, PassGame takes advantage of some of the gameplay rules of Chess. CMAPS, which has already shown itself to acceptable in terms of usability, is used as the basis for PassGame's user interface and input capabilities.

Users typically perceive shoulder-surfing to be a risk in about 17% of their daily device unlocks [9], primarily citing known persons such as children, relatives, and coworkers as the main source of risk. Many users are becoming familiar with the risks associated with shoulder-surfing, and are well aware of when shoulder-surfing is possible. PassGame is intended to protect users against all but the most dedicated attackers, and potentially even against camera surveillance.

In a tradeoff for increased security, PassGame will naturally suffer reduced usability. Because of this inherent tradeoff, PassGame is designed for use only when the user feels like they may be watched. When shoulder-surfing is not a risk, they can use a different, faster, authentication scheme instead. This way, users can have the best of both security and usability, trading off only the memorability requirement to remember two passwords rather than one.

PassGame is designed as a *challenge-response* authentication scheme. Users are presented with a randomly generated chess board, and must make alterations to that board to make it match a predetermined set of rules. PassGame is also still a multi-dimensional password scheme, where rules themselves are one dimension, and each rule also has its own dimensions like color, piece type, and quantity.

3.3 Related Work: Shoulder-Surfing Resistance

Many efforts have been focused on adding shoulder-surfing resistance to existing schemes. In general, there is always a significant tradeoff between the usability of the original scheme and the usability of the shoulder-surfing resistant version. For reference, standard PIN has an average entry time of approximately 1.3s [47, 48].

Roth's Oracle Choice [69] added shoulder-resistance to 4 digit PIN by separating the PIN entry pad into two colors, black and white, with digits assigned to the colors at random. Rather than selecting an individual digit, users pick which color set their desired digit belongs in. This process is repeated several times to select a digit and repeated again until all digits are chosen. Average entry times for Oracle Choice range from 23-26 seconds [69].

SwiPIN [47] splits the digits into two screen sections and assigns each digit a direction. To select a digit, users perform a swipe gesture in the corresponding direction on the corresponding section of the screen, allowing SwiPin to be completed in 4 gestures just like a traditional PIN. SwiPIN has an average entry time of 4-5s, depending on the input configuration.

ColorPIN [48] randomly generates three differently colored letters beneath each digit. Users must remember the color associated with each digit, and enter the letter under the correct digit and with the correct color using a standard alphanumeric keyboard. Just like SwiPin, ColorPIN can be completed with just 4 gestures. ColorPIN has an average entry time of 14s.

Zakaria *et al.* [60] improve Draw-a-Secret's [39] shoulder-surfing resistance by erasing strokes as they are drawn, with minimal impact on usability. Lin *et al.* [70] add a grid to Draw-a-Secret, requiring users to remember the direction in which a stroke passes through the grid in addition to the stroke's shape. Adding the grid requirement had no significant impact on short-term memorability, though usability was not tested.

PicassoPass [53] asks users to choose items from several layers such as color, letter, or shape. Chosen items must be tapped in order while the layers are superimposed over each other to confuse attackers. Usability and memorability were not formally tested.

Convex Hull Click (CHC) [71] and Story [37] are examples of older shoulder-surfing resistant schemes designed for traditional computers.

3.3.1 Testing Shoulder-Surfing

Methods for testing shoulder-surfing resistance vary between authors.

Oracle Choice [69] was tested by showing 8 participants a video recording of various PIN and Oracle Choice entries. Each successful password entry was shown once. All participants were able to guess standard PIN entries (100%), while none of them were able successfully guess Oracle Choice entries after one viewing.

SwiPIN was tested by allowing 3 attackers to view successful entries over the victim's shoulder once. Three guesses were permitted per password. Attackers were able to guess 5/54, 1/54, and 8/54 passwords made with 3 different SwiPIN input designs. SwiPIN was also tested against video attacks, however in authors' words, "none of the designs was significantly secure against video attacks".

ColorPIN was tested using a camera recording of entries with unlimited viewing of the recordings permitted. Camera recordings were taken from a natural angle where obstructing fingers could be an issue, simulating a camera placed over an ATM entry pad. Three guesses were permitted per password. The authors performed the shoulder-surfing attacks themselves, and were able to recover 77% of classic PINs but only 4% of ColorPINs.

Zakaria's [60] improvements of Draw-a-Secret (DAS) were tested by allowing attackers to view a single successful password entry by the experimenter over the experimenter's shoulder. Attackers were given a single guess for each password. A total of 68 attackers were used, and these attackers where able to steal 29/51, 10/51, and 10/51 passwords in the three different input categories, compared against 32/51 standard DAS passwords. Furthermore, 19/51, 34/51 and 23/51 passwords were partially stolen, compared against 17/51 for standard DAS.

Lin's Qualitative Draw-a-Secret (QDAS) [70] was tested by video recording a password using both traditional DAS and QDAS. Ten attackers were permitted to view the video once and make a single guess, resulting in 7/10 guessing the DAS version correctly and 0/10 guessing the QDAS version correctly.

PicassoPass [53] was tested online, by showing participants a single video of a password being entered. The video is recorded by the authors, taken from over the victim's shoulder, and the password is entered on a tablet-style device. Six multiple-choice options were presented for the password, with one guess permitted. Numerical 4-digit PIN, Pattern Unlock, and PicassoPass were tested, with resulting guess rates of 17/18, 13/17, and 0/22 respectively.

3.3.2 Hardware-based Shoulder-Surfing Resistance

Various schemes use hardware to achieve shoulder-surfing resistance. Back-of-Device Shapes (BoD Shapes) [72] utilizes an additional touch screen on the back of the device to authenticate users in a manner similar to Draw-a-Secret. Since the rear of the device is rarely visible, particularly by overhead camera, BoD Shapes has substantial resistance against shoulder-surfing. Glass Unlock [73] uses a near-eye display like Google Glass to communicate information required to the user for authentication, leaving the touchscreen itself as a blank input device. An attacker would need to see the near-eye display to successfully capture the password. EyePassword [74] uses eye-tracking hardware for password entry, so an attacker would need to see the movements of the victim's eyes in order to successfully capture input. Adding hardware can add manufacturing costs and failure points to already complicated and expensive devices.

Some methods rely on existing hardware, for example vibration. Bianchi *et al.* [54] use audio and haptic cues for authentication, requiring the attacker to detect the audio or vibration cues used in addition to capturing visible entry. VibraPass [56] uses haptic vibration cues from a mobile device to assist in accessing an ATM-style bank terminal.

Common built-in biometrics included in many high end devices, such as fingerprint and facial scanners, are resistant to shoulder-surfing but vulnerable to outright theft of biometric data. Chaos Computer Club was able to steal a fingerprint and use it to authenticate using TouchID on the iPhone 5s using only a photograph of the fingerprint [17]. Iris scanners like the Galaxy S8's have been tricked by a simple photo of the eyes with concave lenses glued over them [19]. The iPhone X's FaceID has been fooled by a 3D printed mask [18].

3.3.3 Challenge-Response

Challenge-response mechanisms are typically implemented for computer-to-computer communication, in applications such as encryption and key exchange [75, 76, 77]. Use of challenge-response for human authentication is rare.

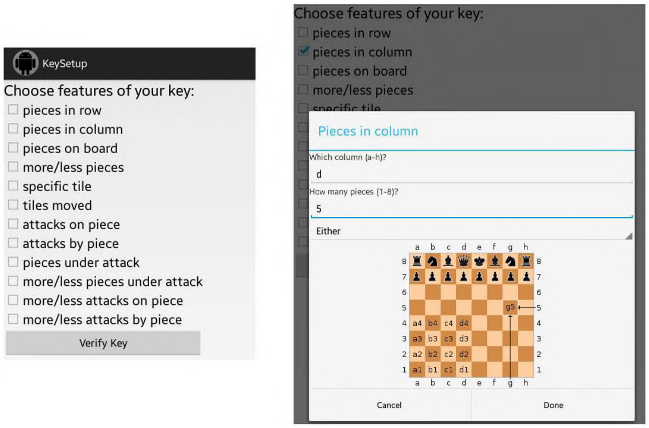
Recognition-based schemes such as RealUser's PassFaces [36] can be considered a type of challenge-response authentication. In PassFaces, users are presented with pictures of faces, and tasked to pick their pre-chosen selection from that group. Deju Vu [35] operates under the same mechanic, using abstract images instead of faces. The recognition cue can be considered a type of basic challenge, where choosing the information from the cue is the correct response. These schemes offer little shoulder-surfing protection, since the attacker can clearly see which image was chosen. In other words, the mapping of the challenge to the response is too predictable.

Another example of human challenge-response authentication is Sasamoto's Undercover [55] scheme. Users are challenged to identify a pre-chosen picture in a group of pictures presented on the screen. Input is conducted using a trackball, which moves or vibrates as a cue to the user. The behavior of input is changed depending on the orientation and vibration of trackball, another aspect of the challenge. Because the trackball is concealed under the user's hand, Undercover offers strong shoulder-surfing resistance, with authentication times averaging from 32-45 seconds.

Turing tests, such as re-CAPTCHA, can also be considered challenge-response schemes, but these are primarily used to determine if a user is human, not for authentication purposes.

3.4 The Design of PassGame

PassGame is designed as a challenge-response authentication scheme based on *rules*. A rule is basically a feature of the Chess board, for example the number of pieces in a given row. Some rules, like the previous example, require no knowledge of Chess, while other rules are based on the attack patterns of certain pieces. A detailed account of the PassGame rules can be found later in this section.



KeySetup

Choose features of your key:

- □ pieces in row
- □ pieces in column
- □ pieces on board
- □ more/less pieces
- □ specific tile
- □ tiles moved
- □ attacks on piece
- □ attacks by piece
- □ pieces under attack
- □ more/less pieces under attack
- □ more/less attacks on piece
- □ more/less attacks by piece

Verify Key

Choose features of your key:

- □ pieces in row
- ☑ pieces in column
- □ pieces on board
- □ more/less pieces
- □ specific tile

Pieces in column

Which column (a-h)?

d

How many pieces (1-8)?

5

Either

Cancel

Done

Figure 11: A Screenshot of Rule Selection (left), The Rule Selection Prompt (right)

At password creation, users pick from the list of available rules. Figure 11 (left) shows an example of the rule selection process, the user selected *pieces in column* as one of their rules. Figure 11 (right) shows the popup that appears after a rule is chosen, allowing users to to set individual features of the rule like column, number of pieces, and color. A reference image is provided to help users understand that rows are labeled 1-8 from the bottom and columns are labeled a-h from the left.

Once the user has chosen all the rules they want, they proceed to a feasibility check or “Verify” step, where the user must satisfy all the chosen rules on an empty board (rules that require removing pieces are exempted from the feasibility check). If a user is able to pass the feasibility check, the password is set. This step is used as a sanity check against otherwise impossible or overly complex passwords. The vast majority of users passed the feasibility check after no more than two tries.

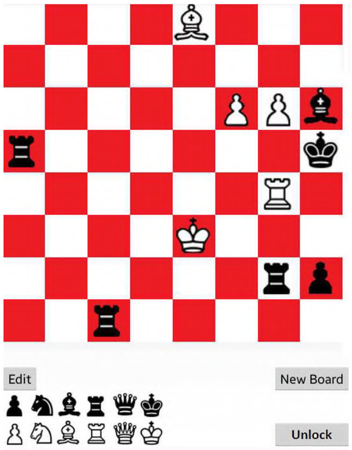


Figure 12: A Screenshot of Authentication

Figure 12 shows an authentication session in PassGame. During authentication, users are challenged with a randomly generated Chess board that they must modify in order to match their chosen rules. Modifying the board can be accomplished by adding new game pieces, removing existing pieces, or moving existing game pieces around. Just like CMAPS, PassGame features a piece selection box for moving pieces onto the board, and an edit button for emptying out tiles. Pieces can be placed or moved anywhere on the board, regardless of the rules of Chess.

3.4.1 Random Board Generation

PassGame presents the user with a randomly generated board for each authentication session. The user can request a new randomly generated board at any time with no penalty.

The random board generation algorithm tries to generate a large variety of boards, but favors placing pieces with roughly the frequency that they typically appear in midgame Chess. For each tile, a number is randomly generated and a piece or empty tile is accordingly assigned. The seeding bias of pieces is static. Pieces are generated at a ratio pawn:knight:bishop:rook:queen:king of 12:8:8:6:4:2, meaning pawns will typically be the most frequently occurring piece and other pieces will appear roughly in ascending order of power, with the king as the rarest piece. Each piece is randomly assigned a color with an even 50:50 chance. The bias of empty tiles is chosen at random, varying from 0-360, so an average board will have a bias of 180, corresponding to a board that is on average 18% populated. That is, if an empty tile seeding bias of 180 is randomly chosen, the odds of any given tile being empty is $180/(180+12+8+8+6+4+2)$, roughly 82%.

Users can request a new random board for a variety of reasons: (1) the password is not possible on the given random board, for example the user needs to remove 5 pieces, but there are only 3 on the board, (2) the user wants an easier board to work with, for example the user's password is easier to enter on a less-crowded board, (3) the user thinks shoulder-surfing will be too easy on the current board, or (4) the user has modified the board too much and no longer remembers their modifications or the initial state of the board.

3.4.2 Available Rules

PassGame was designed with 12 rules, the first 6 of which require no knowledge of how to play Chess. Some rules require users to pick a color: black, white, or either (indicating that the rule can be satisfied with pieces of either color). Some rules require the user to chose a piece type: pawn, knight, bishop, rook, queen, or king. Users will be asked to pick multiple rules at the same time, but for purposes of this experiment, picking the same rule multiple times with different parameters is not permitted. This is to ensure a variety of rules are tested.

Some rules require pieces of a specific color or type to exist on the board. If there is no piece matching the type and color description on the random board, the user can simply add one.

Rule R1 Tile Count: Consider the board as a numbered grid from 1 to 64, where the bottom left tile is 1 and the top right tile is 64. When the randomly generated board is initialized, the user has a tile count of zero. Moving a piece to the right by one tile adds 1 to the total, likewise moving a piece left subtracts 1 from the total. Since there are 8 columns in a row, moving a piece up one row adds 8, and moving a piece down subtracts 8. Placing a new piece simply adds the tile number, and removing an existing piece subtracts the tile number.

The user chooses a tile count, n_{tile}. To satisfy this rule, the user must modify the board so that the tile count matches n_{tile}. For example, if n_{tile} is 8, the user can move a piece up one row, or they can move one piece right 5 tiles and another piece right 3 tiles, or they can add a piece to tile 20 and remove a piece from tile 12. The maximum password space for this rule individually is $[-2080, 2080]$ as $\sum_{i=1}^{64} i = 2080$, but obviously larger tile counts will not be practical for usability purposes.

Rule R2/R3 Pieces in a Row/Column: For these rules, the user chooses a row or column, a color, and a number n from 1-8. The specified row or column must contain n pieces of chosen color. For example, “4 white pieces in row 4” or ”3 pieces of either color in column c”. The password space is $8 \times 3 \times 8 = 192$, since there are 8 choices for row (1-8) or column (a-h), 3 choices for color (black, white, either), and 8 choices for n.

Rule R4 Pieces on Board: For this rule, the user chooses a color and the number n of pieces on the board from 1-64. The password space of this rule is $3 \times 64 = 192$.

Rule R5 More or Less Pieces: For this rule, the user chooses a color and the number n of pieces to be added or removed from the randomly generated board. Only the total needs to match n. For example, if the user picks “2 more white pieces on the board”, the user can remove 4 white pieces, then add 6, for a total of 2 white pieces added to the board. The password space is $3 \times 64 \times 2 = 384$, because there are 3 choices for color (black, white, either), 64 tiles on the board, and the user can choose to add or remove the pieces.

Rule R6 Specific Tile: This rule follows classic CMAPS mechanics; the user selects a piece type, color, row, and column. A piece matching the type and color must be present at the row and column address specified. Like CMAPS, on its own, this rule is not shoulder-surfing resistant. The password space of this rule is $6 \times 3 \times 8 \times 8 = 1152$.

The next 6 rules require basic knowledge of the attack patterns of Chess pieces. For example, bishops can attack along a diagonal line. Note that there are many ways to add or remove attacks in Chess. An attack can be added by removing a piece blocking line-of-sight, by adding an attacking piece, or by adding a defending piece into an existing path of attack. Similarly, attacks can be reduced by blocking attacking pieces, removing attacking pieces, or even removing the defending piece.

Rule R7 Attacks on a Piece: For this rule, the user chooses a piece type, color, and the number n of attacks. *At least one* piece matching the type and color description must have n attackers on it. Note that the maximum number of attacks (n_{max}) on a tile is 16, with 4 diagonal attacks, 2 horizontal attacks, 2 vertical attacks, and 8 attacks by knights, however not all tiles are able to support 16 attackers. Corner tiles for example, can have only 5 attackers maximum, so it may be necessary to move a piece or place a new one to satisfy this rule in some circumstances. The password space of this rule is $6 \times 3 \times 16 = 288$.

Rule R8 Attacks by Piece: For this rule, the user chooses a piece type, color, and the number n of attacks. *At least one* piece matching the type and color description must be attacking n pieces. For a king, a queen, or a knight, there are $3 \times 8 = 24$ combinations because a king, a queen, or a knight can attack a maximum of 8 pieces. For a bishop or a rook, there are $3 \times 4 = 12$ combinations because a bishop or a rook can attack up to 4 pieces. For a pawn, there are only $3 \times 2 = 6$ combinations because a pawn can attack just two pieces. The total password space for this rule is $3 \times 24 + 2 \times 12 + 6 = 102$.

Rule R9 Pieces under Attack: For this rule, the user chooses a color and the number n of pieces under attack on the board. The board must contain a total of n pieces of the correct color under attack. The password space of this rule is $3 \times 64 = 192$, since the board can contain up to 64 pieces.

Rule R10 More or Less Attacks on a Piece: For this rule, the user chooses a piece type, color, and the number n of attacks to add or remove. The user must add or remove n attacks on a piece of the chosen type and color. If the specified piece doesn't exist, the user can add it, but of course it is not possible to *remove* attacks from a newly placed piece. Because the maximum number of attacks on one tile is 16, and attacks can be added or removed, the password space of this rule is $3 \times 6 \times 16 \times 2 = 576$.

Rule R11 More or Less Attacks by a Piece: For this rule, the user chooses a piece type, color, and the number n of attacks by a piece to add or remove. The user must add or remove n attacks by a piece of the chosen type and color. As described in Rule R8, different pieces have a different number of a maximum attacks. Since attacks can be added or removed, the password space of this rule is $(3 \times 24 + 2 \times 12 + 6) \times 2 = 204$

Rule R12 More or Less Pieces under Attack: For this rule, the user chooses a color, and the number n more or fewer pieces that should be under attack on the board. The board must be modified to add or remove n pieces of the correct color which are under attack. The password space of this rule is $3 \times 64 \times 2 = 384$

3.4.3 Additional rules

PassGame can be modified to support a theoretically near-infinite number of rules. Rules can be generated based on arbitrary criteria, for example “pieces on a white tile”. Existing rules can also be split into more detailed versions, for example “bishops in column e”, or less detailed rules such as “pieces on the left half of the board”. There are also many more Chess features that can be used, for example “is there a pawn that can en passat?”

Because the number of rules can be increased almost infinitely, the password space is potentially infinite as well. The more rules are made available, the harder it will be for attackers to iterate through all the rules and figure out which ones a user has chosen. Adding or varying rules periodically can confound attackers who program dictionaries or automated brute force tools, causing them to constantly need to update.

While including more rules may impact usability, users do not necessarily need to read through all the rules, they only need to pick a few arbitrarily and make sure that they understand them. Furthermore, changing rules may reduce memory interference, since the user will know that none of their old rules are applicable. A long-term study of applying rule changes and making a very large number of rules available may be a potential task for future work.

3.5 Security of PassGame

Table VIII: Password Space of PassGame Rules

Rule	Overall	Effective
R1	2080	17
R2	192	120
R3	192	120
R4	192	66
R5	384	63
R6	1152	1152
R7	288	72
R8	102	75
R9	192	9
R10	576	108
R11	204	102
R12	384	27
Total	5938	1931

Table VIII shows the overall and effective password space for each of the 12 PassGame rules. The effective password space is calculated from responses given by participants in the user study. Participants were asked, at the end of the experiment, what is the *largest* value they would chose for n in each rule, for example in rule R1, participants indicated they would use no more than 17 tile moves. The total password space for one-rule passwords is the sum of the individual rules, 5938. Since most of the password space of some rules is not usable, for example rule R1 with a tile count of 500, the effective one-rule password space of 1931 is more relevant for password space calculations.

The password space of multiple rules is difficult to calculate because of potential conflicts between rules. For example, "2 white pieces in row 3" cannot be chosen at the same time as "1 white piece on the board".

Let us assume as a pessimistic lower bound that conflicts reduce the available choices in each subsequent rule by half. The two rule password space is then $5938 * 5938/2 = 17,629,922$. The two rule effective password space is $1931 * 1931/2 = 1,864,381$, still 186 times larger than 4-digit PIN and about 5 times larger than Pattern Unlock on a 3 by 3 grid.

The four rule overall password space is $5938 * (5938/2) * (5938/4) * (5938/8) = 1.9 * 10^{13}$, roughly on par with an 8 character alphanumeric password. The four rule effective password space is $1931 * (1931/2) * (1931/4) * (1931/8) = 2.17 * 10^{11}$, roughly on par with a 7 character alphanumeric password.

PassGame has significantly greater security strength at 2 rules than PIN or Pattern Unlock, and offers nearly 8-character standard alphanumeric levels of security at 4 rules using a pessimistic lower bound.

3.6 PassGame User Study

3.6.1 Participants

Participants were recruited using fliers and leaflet advertisements. A $10 cash incentive was given for completing the in-lab sessions of the user study. Thirty seven participants were recruited, of which 36 successfully completed the experiment (13 female). Seven participants were aged 20 or younger, 22 were aged between 21 and 25, 4 were aged between 26 and 30, and 3 participants were aged over 30. Among the 36 participants 17 were undergraduate students, 13 masters students, 5 doctoral students, and 1 staff. Participants were asked the question: "Are you skilled at using smartphones or mobile devices?" On a scale from 1 (strongly disagree) to 5 (strongly agree) participants rated their skill an average of 4.28, with the majority (89%) rating their skill at 4 (agree) or better.

3.6.2 Overview

PassGame was implemented in Android, screenshots of PassGame are available in Figures 11 and 12. The user study is designed similarly to the CMAPS user study, using a one-week memorability study with two reminder emails in the interim.

Initially, participants come to the controlled laboratory environment to fill out demographic information and learn how to set a PassGame password. The tutorial consists of a 15 minute series of videos that covers the basics of PassGame and describes all the available rules. Before leaving the lab, participants generate their own PassGame password and re-authenticate themselves successfully on two different random boards.

As in [57] and the CMAPS study, participants were asked to recall their passwords several times over the experiment to simulate daily use. Reminder emails were sent to participants 3-4 days after the first session and again 5-6 days after the first session. An online emulated version of the PassGame application is used to complete the reminder sessions. Completion of the reminder sessions is optional and must take place within 36 hours of the receipt of the reminder email.

Seven days after the first session, participants return to the controlled laboratory environment to recall their passwords. Up to five minutes are allowed for password recall. At the end of this session, participants fill out a survey rating the usability of PassGame against their favorite authentication scheme.

Conditions

Participants are randomly assigned into one of three conditions.

1R: Participants in this condition set a password with just one rule. Using Rule R6 is not allowed because it is not shoulder-surfing resistant on its own.

2R: Participants in this condition set a password using 2 rules.

4R: Participants in this condition set a password using 4 rules.

Initial pre-experiment testing indicated that passwords with 5 rules or more would require too much effort to create and use, particularly because of contradictions between different rules. If the same rule was allowed multiple times, this was found to be much less of an issue, since users could duplicate easier-to-use rules such as rule R2/R3 which can easily be configured to not contradict. This experiment was limited to a single selection of each rule in order to encourage participants to test different rules, and therefore this experiment was limited to 4 rules or fewer. A future work will include PassGame with no limitations on the number of rules that can be chosen, and no limitations on the number of times each rule can be chosen.

3.6.3 Memorability Results

The following hypotheses are generated for the memorability of PassGame.

H1: PassGame will be less memorable as more rules are used.

H2: PassGame will be more memorable to people who know how to play Chess.

H3: PassGame will be more memorable to people who used the reminder emails.

Table IX: PassGame Recall Rates by Condition

Conditions	Participants	Recall	Recall Rate
1*R*	12	12	100%
2*R*	14	14	100%
4*R*	10	7	70%

Seven-day recall results for PassGame are shown in Table IX. None of the 1R or

2R participants forgot their passwords over the 1 week period. An omnibus chi-squared test shows a significant difference between the three conditions ($\chi^2 = 8.51$, $p = .014$). Hypothesis H1 is supported by the data, PassGame is less memorable with 4 rules than with 1 or 2, though even 4 rule PassGame is quite memorable.

Of the 36 participants, 5 used only the first reminder session, 2 used only the second reminder session, 24 used both reminder sessions, and 5 did not use either reminder session. All three participants who forget their passwords used both reminder sessions, so H2 is not supported. An omnibus chi-squared test reveals no significance ($\chi^2 = 1.64$, $p = .651$).

Participants were asked if they knew how to play Chess, and 31/36 (86%) indicated that they did. Of the 3 participants who forget their passwords, 2 knew how to play chess and 1 did not. An omnibus chi-squared test reveals that there is no significant difference ($\chi^2 = 1.04$, $p = .309$), so H3 is not supported.

3.6.4 Usability Results

The following two hypotheses are generated for the usability of PassGame.

H4: PassGame will have significantly worse entry times with more rules, especially because of the increased cognitive burden of making multiple rules work at the same time.

H5: Users will rate PassGame significantly worse in usability than traditional authentication schemes such as PIN.

Table X: Average Entry Times, New Boards, and Attempts Needed per Successful Authentication

Condition	Total (s)	Correct (s)	Boards	Attempts
1R	33	23	1.6	1.22
2R	110	44	1.9	2.07
4R	143	49	2.1	2.63

Table X shows average total entry time, average entry time for a single successful attempt, average new board requests per authentication session, and average number of authentication attempts per successful session. Total entry time includes time spent thinking, requesting new boards, and making incorrect attempts. A Kruskal Wallis test for total entry between the three conditions finds no significant difference (H=4.996, p=.082) in total entry time.

Correct entry time is measured as time from application load to successful authentication, or from last authentication failure to successful authentication, whichever is lower. A Kruskal Wallis test on the timings for the first correct attempt finds no significant difference between the three conditions (H=3.741, p=.154). Based on both timing results, H4 is not supported, though the trend seems to indicate that H4 would be supported with a larger pool of users.

Timings for a single PassGame successful authentication attempt are not particularly fast, roughly on par with PC-based schemes like Deja Vu (32s) [35], Convex Hill Click (72s) [71], or CDS (20s) [59] and significantly slower than touch-based schemes like SwiPin (4-5s) [47], ColorPIN (14s) [48] and The Phone Lock (11-26s) [49].

The slowness of PassGame can likely be attributed to its challenge-response nature. Most PassGame passwords can be solved in less than 6 gestures regardless of the configuration of the random board. However, PassGame is effectively a simple puzzle solving task; the challenge is presented as a random board and the rules that the user must satisfy can be considered the rules of the puzzle. Most of the time spent authenticating is spent on the cognitive burden of puzzle solving, not on the mechanical actions of input. As with any puzzle solving task, users will improve over time as they gain experience and skill. The rate at which users improve and the absolute limit to an individual's ability will likely vary between users. A focus of future work will be a longer user study that analyzes the improvement of PassGame users over time.

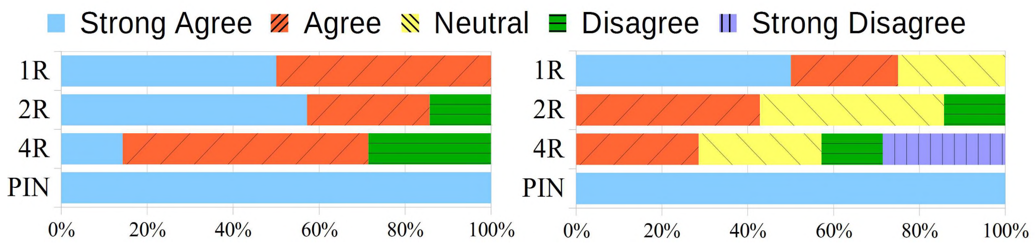


Figure 13: Usability Survey Results for Convenience (left), Speed (right).

Table XI: PassGame and PIN Average Survey Ratings

Scheme	Ratings	Conve.	Speed
PassGame-1R	4	4.50	4.25
PassGame-2R	7	4.29	3.29
PassGame-4R	7	3.75	2.57
PassGame-all	18	4.06	3.22
4-digit PIN	10	5	5

At the end of the second session, participants were asked to rate PassGame and their preferred traditional scheme on a scale from 1 (Strongly Disagree) to 5 (Strongly Agree) for the following statements: (1) It is convenient to enter a password using this scheme, (2) The speed of entering a password with this scheme is fast. Figure 13 shows the usability ratings for PassGame in different conditions as well as PIN. Not enough ratings were collected from other schemes, such as Pattern Unlock and Fingerprint, to include them, however comparable ratings are available in the CMAPS user study in Section 2.7. Table XI shows the average usability ratings collected in this experiment for different conditions of PassGame as well as PIN.

For purposes of statistical analysis, results are grouped as satisfied (4 or higher) or unsatisfied (3 or lower). A chi-squared omnibus test on the three conditions of PassGame and 4-digit PIN shows no significant difference in convenience ($\chi^2 = 4.11$, $p = .25$), however it does show a significant different in speed ($\chi^2 = 11.04$, $p = .01$). Pairwise testing reveals the results are significant between 2R and 4-digit PIN ($\chi^2 = 7.47$, $p < .01$) and between 4R and 4-digit PIN ($\chi^2 = 10.12$, $p < .01$). As expected from hypothesis H5, users rate PassGame below traditional authentication schemes in metrics of usability.

3.6.5 User Choice in PassGame

Because some rules are easier to understand than others, it is likely that hotspots will exist in PassGame. As was the case in CMAPS, certain colors and pieces may also be more popular than others.

H6: Hotspots in rule selection will exist; that is, users will chose the rules which are easiest to understand and implement more frequently.

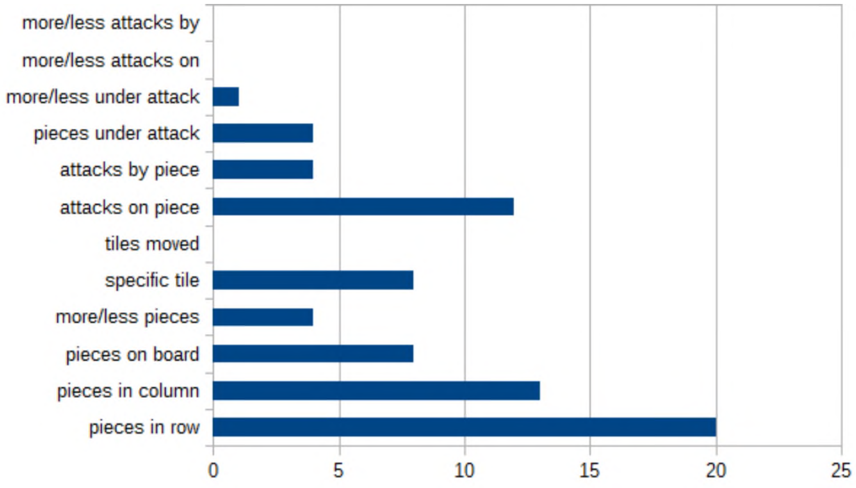


Figure 14: Frequency of Rule Selection

Figure 14 shows the frequency with which each rule was selected in the user study. As expected from hypothesis H6, users were most likely to like select the "easiest" rules, particularly rules R2/R3 (pieces in row/column).

When a number of pieces *n* was required, most users (85%) used 3 pieces or fewer. When a specific piece type was required, users chose the king (46%) and queen (29%) over the rook (13%), bishop (0%), knight (8%), and pawn (4%). This is contrary to CMAPS, where the king was least common piece by occurrence. Notably, pieces were chosen roughly in descending order of importance in Chess, indicating that knowledge of Chess had some impact on user choice.

3.6.6 Shoulder-Surfing Study

After the conclusion of the memorability and usability study, participants were invited back to the laboratory for a shoulder-surfing study. Three different PassGame passwords were recorded using 2, 3, and 4 rules. Only the 4 rule password was formed using rules that require chess knowledge. Recordings are made by screen capture, with touch areas indicated on-screen. For each password, five successful entries are recorded on five different random boards. No modifications unrelated to the password are made, that is, each modification to the board is made purposefully in order to reproduce the password.

Participants were told that the passwords were rated as "easy", "medium", and "hard", that each password had 2-4 rules, and that only the hard password required chess knowledge. Since participants were already familiar with PassGame, only a brief 5 minute recap of the rules was conducted. Participants were provided with a sheet of paper that listed all of the rules, and with sheets of paper with blank chess boards for note taking. A $100 cash prize pool was offered to split between participants who could guess the medium or hard passwords ($20 for medium, $80 for hard).

Table XII: Successful Shoulder-Surfing Attempts by Condition

Strength	1 Viewing	5 Viewings	Unlimited ($\leq$ 1 hr)
Easy	0	5	15
Medium	0	0	3
Hard	0	0	0

Table XII shows the results of the shoulder-surfing study. Participants were initially limited to a single viewing of each password entry as in [53] and [69], simulating a shoulder-surfing who has limited access to viewing the device. Unlike studies such as [53] which allow viewing of just a single password entry, this experiment allows participants to view *all five* successful password entries. None of the fifteen participants were able to recover any of the passwords after viewing all five successful entries once.

Next, PassGame is tested against repeated observation, similar to the approach in

[78], by allowing participants an additional 5 sequential viewings of the 5 password entries, for a total of 6 viewings. Shi *et al.* [79] show that the probability of guessing a password correctly using shoulder-surfing observations is proportional to the *intersection* the attacker observes between successful captured entries. With their own scheme, the probability of correctly guessing a password after viewing just two different successful entries is estimated at 20%. Similarly, Chameleon [80] is considered by its authors to be secure against 3 or fewer captured successful entries. PassGame is tested with 5 captured successful entries, more than any other shoulder-surfing resistant scheme has been tested against. Furthermore, unlike the vast majority of shoulder-surfing experiments, this experiment does not limit how many guesses participants can make on the device.

After viewing the 5 entries an additional 5 times (a total of 30 viewed successful login attempts, including the previous single-viewing), 5 participants (33%) were able to crack the easy password. None of the participants were able to crack the medium password, though some participants were able to partially guess 1-2 rules. None of the participants were confident enough to attempt the medium password on the device. Only 5 participants opted to attempt the hard password, and all 5 were unable to crack it, with several expressing the sentiment that it was “impossible” and “nobody would be able to get that”.

Finally, participants were allowed unlimited viewing of the recordings, including pause/rewind capabilities, simulating a worst-case camera attack. Participants were permitted to work in teams during this stage if they wished, and as before were allowed unlimited attempts on the device. All 15 participants were able to crack the easy password under these conditions. None of the participants were able to crack the medium or hard passwords after 20 minutes, however some participants opted to continue trying, and 3 participants were able to crack the medium password after 40 minutes (2 were grouped as a team). Only 5 participants opted to try the hard password beyond the 20 minute mark, with none able to crack it after 1 hour.

The hard password was eventually cracked by one very dedicated participant after

3.5 hours, using 6 attempts on the device. Although the participant guessed one of the rules incorrectly, they were close enough to pass authentication after several attempts. An exhaustive search was used, as expected, studying the intersection between successful entries to exclude rules one-by-one. In theory, a computer program can perform the same exhaustive search must faster than a human participant. A planned future work is to write a program that analyzes the intersection between successful entries in order to determine how many entries are required on average for a successful guess. Because the challenge is randomly generated, the number of entries required will likely vary even for captures of the same password.

The shoulder-surfing study demonstrates that PassGame has superb shoulder-surfing resistance, far exceeding any of the works discussed in Section 3.3.

3.7 PassGame Discussion

We can safely assume that CMAPS has little to no shoulder-surfing resistance advantage over traditional PIN or Pattern Unlock. With that assumption in mind, PassGame offers a clear victory over CMAPS in terms of shoulder-surfing resistance. Based on initial results, PassGame may offer better shoulder-surfing resistance than any other knowledge-based scheme that doesn't require additional hardware to date. In terms of raw password space, PassGame is already better than PIN and Pattern Unlock, and by adding more rules it can easily be brought up to match CMAPS or longer alphanumeric passwords.

The shoulder-surfing resistance of PassGame clearly comes at a substantial price to usability: entry times on PassGame are significantly worse than CMAPS, and an order of magnitude worse than traditional authentication schemes. In terms of actual gestures required for authentication, PassGame is not especially different from CMAPS, and in many cases may require fewer gestures, particularly when a single piece placement can be used to satisfy many rules at the same time. Consider the PassGame password: "1 white piece in row 3, 1 white piece in column f, 1 white piece under attack, 1 attack by a white bishop". Provided a black piece is randomly generated in line of sight, this password can be completed with a single white bishop placed on tile 3f. In some cases, with particularly simple passwords, PassGame can even be completed with *zero* gestures in some cases. For example, the password "1 white piece in row 3", may very well be satisfied simply by random board generation. However, passwords with this level of simplicity are not very secure.

PassGame's poor usability is largely due to the increased cognitive load of the scheme. As users improve at solving the PassGame puzzle, authentication times may improve, but they will likely never improve over a scheme that requires no puzzle solving, like CMAPS. Additionally, cognitive load is sometimes a difficult price to pay on the mobile platform, especially when the device is used in situations where the user's focus is demanded elsewhere, for example when crossing the street.

The design of PassGame is not as a primary authentication scheme, but rather as a supplementary secure authentication scheme the user can trust when they feel they are being watched. Alternatively, PassGame can be used to secure valuable information, for example a banking application, while a less shoulder-surfing resistant password is used for general device access.

Users of PassGame can continue to trade usability for security by making unrelated adjustments to the game board. PassGame allows users to make adjustments to the game board that have nothing to do with their rules simply for purposes of confusing attackers. A future work will investigate if making unrelated adjustments can trick even a guessing algorithm programmed specifically to crack the intersection between successful entries.

Like CMAPS, PassGame can reduce memory interference between different passwords by utilizing a different game for each account, device, or successive password generation. For example, a user's bank account PassGame may use Chess, while their stock market account may use Backgammon. In corporate environments, where passwords are often set to expire after some time, a user's password may be based first on Chess, then Backgammon once that password expires, and so forth. Since the game board, pieces, and rules of Backgammon are very different from Chess, there will be minimal memory interference between the Chess-based password and the Backgammon-based password.

CHAPTER IV

AUTHENTICATION IN VR

4.1 Outline

This chapter is based on a work published in the International Journal of Communications, Network and System Sciences [81].

Section 4.2 introduces the novel concept of extending MAPS to authentication inside a 3D environment, describing several possible situations where a user may be motivated to conduct authentication inside a 3D virtual environment. Related work on 3D authentication is presented in Section 4.3. The concept of 3D authentication is grounded in several physical and psychological phenomena; a major contribution of this work is to describe these advantages in Section 4.4. The design and implementation of a proof-of-concept for 3D authentication, dubbed 3DPass, is presented in Section 4.5, showing that 3DPass is significantly more greater in scale and complexity than any previously developed work in 3D authentication. The theoretical security advantages of 3D authentication, and a calculation of the security strength of 3DPass, are presented in Section 4.6. A user study demonstrating the superior memorability of 3DPass vs traditional authentication after a period of two weeks, and the promising entry times and qualitative usability results of 3DPass are presented in Section 4.7. Conclusions and future work are addressed in Section 4.8.

4.2 Expanding MAPS to Virtual Reality

In the previous two chapters, MAPS was used to improve the security strength of mobile authentication. This chapter shifts the focus from traditional mobile authentication to a more futuristic domain: Virtual Reality (VR). Today, VR is rapidly gaining market traction [82], with many high-end mobile devices bundling VR add-ons such as the Samsung Gear VR and Google Daydream. In the future, as hardware continues to improve, even low-end devices may be able to run VR. While expensive add-ons such as the aforementioned Gear and Daydream will likely continue to be expensive, simple VR solutions such as Google Cardboard can bring the VR experience to everyone who can afford a mobile device.

This chapter focuses on the development of an authentication system that is specifically designed for a 3D environment, a concept that we'll refer to as a 3D authentication scheme or *3DPassword*. A scheme like this could serve multiple purposes in the future.
(1) Protecting virtual resources in a virtual environment. Consider a virtual environment where some part is restricted, for example a certain room or a safety deposit box. A 3DPassword could grant entrance to a secure virtual asset.
(2) A high security alternative for traditional authentication. As we'll note later on, 3DPasswords may have exceptional advantages in security strength, and are inherently secure against shoulder-surfing when used with a near-eye display.
(3) As authentication when already engaged with VR. As VR gains popularity, transitioning out of VR in order to conduct authentication may become cumbersome. In other words, when the user is already engaged in VR, it may be easier for them to enter a 3DPassword rather than leaving VR to enter a traditional one. Particularly on mobile, as naked-eye 3D

and other technologies gain popularity, users may find themselves spending more time in 3D environments where a native form of authentication is desirable.

This chapter addresses a novel scheme dubbed 3DPass, a proof-of-concept 3D authentication scheme based around fundamental physical and psychological advantages of using 3D environments. We will see that 3DPass, building on existing work in the field, has excellent initial results in memorability, usability, and security.

4.3 VR Introduction and Related Work

3D authentication presents a new paradigm for authentication. To date, passwords have been based on the user's knowledge of facts, information, or secrets. While a traditional authentication scheme asks users: "*what do you know?*", a 3D authentication scheme can ask users to *reproduce an experience*. The user is tasked with recreating an event that happened to them, reproducing a temporal and spatial sequence of events from their own personal history by reliving it inside a virtual environment. If a traditional password is based on *what you know*, a 3Dpassword is based on *what you experienced*.

Technically, any 3D display capable device, even a simple display like a mobile phone or monitor screen, can support a 3D authentication scheme. However, the advantage to using an *experience* over information for authentication is that the user can leverage their level of immersion, which we will discuss in more detail later, to improve the memorability of authentication. The higher the fidelity of the 3D experience, the better the feeling of immersion.

Alsulaiman and Saddik [83] developed the pioneering work in 3D authentication, defining it as a series of interactions with a virtual world. The original concept was envisioned in a manner similar to real-world authentication, where the user will, for example, type a password at a virtual terminal, present a virtual biomteric token, or move a 3D object from one place to another.

Many authors such as [84, 85, 86] have proposed similar ideas, but none have made a complete functional implementation on the level of 3DPass, the scheme proposed in this chapter. The contribution of 3DPass is threefold: (1) A strong basis in psychological and physical advantages available to 3D technologies, (2) A design that directly integrates moving and navigation as a part of the authentication process, and (3) A full-scale proof-of-concept implementation and lab-based user study.

4.4 Advantages of a 3D Authentication Scheme

The 3D authentication scheme is founded in various physical and psychological phenomena, providing advantages in memorability and usability.

4.4.1 Psychological Phenomena

Presence: In psychology, presence is a term that refers to the sense of "being there", inside a virtual environment [87]. Presence is considered the key of virtual reality [88]. Although presence doesn't necessarily improve performance in and of itself, Slater *et al.* conclude that "presence is concerned with how well a person's behavior in the virtual environment matches their behaviors in similar circumstances in real life" [89]. When faced with familiar tasks that emulate real life, it's possible that users with a strong sense of presence will experience improved performance. In other words, 3D authentication can leverage presence to improve performance.

Spatial Memory: Spatial memory, used to navigate the environment and remember the location of places and items, is neurologically distinct from other types of memory like object recognition and factual recall [90, 91, 92]. When compared against passive observers, Attree *et al.* [93] found that active participants in VR navigation had better recall for the spatial layout of the environment. Using active navigation in a 3D authentication scheme can tap into human spatial memory for purposes of authentication.

To date, no authentication method utilizes navigation, so the memorability of a "navigation-based" password remains untapped and unknown. Intuitively, navigation is an extremely basic form of memory, one that even many animals are capable of. Tolman [94] proposes that humans and rats do not just remember paths through their environment one by one, but rather form a high level map of the environment, called a "cognitive map", for navigation. The formation of this map is intuitive, even to animals.

Episodic Memory: According to Tulving [95] memory can be broken into two categories: autobiographical memory of experiences known as *episodic memory*, and fact-based, cognitive reference memory known as *semantic memory*. Episodic memory deals with a person's recollection of personally experienced events, and semantic memory deals with a person's knowledge, such as language and math.

To date, most traditional authentication schemes rely on semantic memory; the user simply recalls some factual knowledge they have remembered earlier. Instead of asking the user for facts, a 3Dpassword asks users to recreate a series of events which the user has already experienced. In other words, rather than being based on *what you know*, a 3Dpassword is based on *what you experienced*. To date, no authentication method leverages autobiographical experience for authentication purposes.

Context: Information tends to be easier to recall when it is recalled in the same environment where it was learned [96]. In a 3D authentication scheme, users have the unique opportunity to return to exactly the same environment, in the exactly the same state, where they first learned the password.

Evidence suggests that words which are more image-arousing improve contextual memory [97]. A more realistic environment may likewise improve contextual learning, and further improve memorability.

Context is tied closely to episodic memory, as temporal and spatial relations between events can be a part of remembering those events. That is, to remember what you did in the kitchen, you may need to recall when you went there, and what preceded the trip to the kitchen.

The 3Dpassword represents a new paradigm in how users remember their passwords. The use of spatial memory for navigation, episodic memory instead of semantic memory, and context, have never before been applied to authentication.

4.4.2 Physical Phenomena

Four depth cues are missing from traditional 2D displays: stereo parallax, motion parallax, convergence, and accommodation [98, 99]. The former three are available in varying degrees on modern 3D displays. Accommodation is not currently available on any commercial product.

Stereo parallax: Because of the space between human eyes, each eye perceives a slightly different image. This difference between what each eye sees is used as a depth cue. Stereo parallax is particularly useful in determining depth of nearby objects, since the difference between each eye's image is more substantial at close range. In displays, true stereo vision is available only in Head-Mounted Displays (HMDs), where a different image is presented to each eye. Glasses-enabled 3D displays and naked eye 3D displays can also take some advantage of stereo vision. Ijsselsteijn *et al.* [100] conclude that adding stereoscopic information to a display improves reported presence.

Head Tracking and Motion Parallax: When a person moves their head, objects which are far away appear to travel less distance than objects which are closer. This effect is known as motion parallax. A number of modern displays, primarily HMDs like the Oculus Rift and HTC Vive, have head-tracking, allowing them to move the on-screen image with the motion of the user's head, proving motion parallax. Ferris [101] demonstrates that deliberate movements of the head can be used to get very accurate estimates of the distance of objects, so motion parallax can be useful in applications where gauging distance is important.

Head tracking also allows users to target objects by simply turning their head to look at them, thus reducing the need for an input device for various aiming tasks and potentially improving usability, depending on the quality of head-tracking versus an alternative input method.

Egocentric vs Exocentric Viewpoints: An egocentric approach emulates a first-person view, while an exocentric view is the more familiar third-person view typically used in television and cinema. Slater *et al.* [89] find that an egocentric approach with an HMD leads to higher reported presence vs an exocentric approach.

Hendrix *et al.* [102] find that the reported level of presence is significantly higher when stereoscopy and head tracking are provided. Barfield *et al.* [103] find that users performed wire-tracing tasks faster when stereo vision was added and with fewer errors when head tracking was added. Hoffman *et al.* [104] conclude that the Oculus Rift HMD, also used in this work, can elicit a strong illusion of presence. Tavanti and Lind [105] found that merely adding 3D depth cues, such as shading and perspective, to an otherwise 2D scheme can improve memory performance.

All of these findings bring us to the following hypotheses on 3D Authentication.

H1: Physical advantages of VR will lead to increased presence ratings and therefore better usability vs a traditional display. In other words, more immersive VR such as an HMD will perform better than a traditional 2D display in usability metrics such as entry time. Hypothesis H1 is addressed in Section 4.7.3.

H2: 3DPasswords will have improved memorability vs traditional authentication, and this advantage will be more significant with approaches that have improved presence. Hypothesis H2 is addressed in Section 4.7.2.

4.5 Implementation of 3DPass

3DPass is developed in Unity and coded in C# using artwork from the Unity Asset Store. Full support for head tracking and stereo vision is available when using 3DPass with the Oculus Rift HMD.

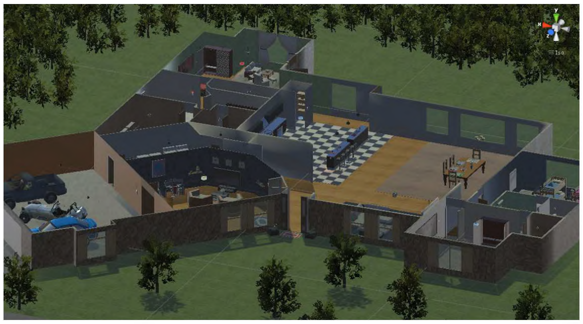


Figure 15: An Overhead View of 3DPass Taken in Unity. The roof has been removed and ambient lighting has been increased for better visibility.

3DPass places the participant's avatar at the entrance to a virtual home. Figure 15 shows a top-down view of our environment from the Unity editor. The home is a tavern-style one story structure with a spacious central loft area containing the kitchen, dining room, and living room. Other rooms, like bedrooms and the study, branch out from the central area. The environment is populated with objects in expected locations. For example, there are appliances and food items in the kitchen, couches and entertainment items in the living room, laundry items in the utility room, and cars in the garage. Users can walk around the environment and interact with most objects around the house.

Small items such as books, fruit, or soap can be picked up and carried around, gently dropped, or thrown. The distance traveled when an object is thrown is proportional to its weight– a banana flies further than a ceramic plate. Stationary items, such as the stove and fireplace, can be interacted with. The stove can be ignited and turned up or down, sinks and bathtubs have running water which fills their corresponding containers, and televisions can be turned on or off and flipped to one of four channels. Lights around the house can be turned on or off, and the default ambient lighting is set for a dawn-like environment where everything is fully visible, but visible more clearly when lights inside the house are turned on.

Doors, drawers, and cabinets can be opened or closed with precision. Objects can be placed inside containers, and when practical the object will temporarily bind to the container. For example, a fruit placed in a bowl will bind to the bowl, so next time the bowl is picked up, both objects will travel together. This allows for indirect actions such as placing something in a bowl, then throwing the bowl as a means from moving the original object from place to place.

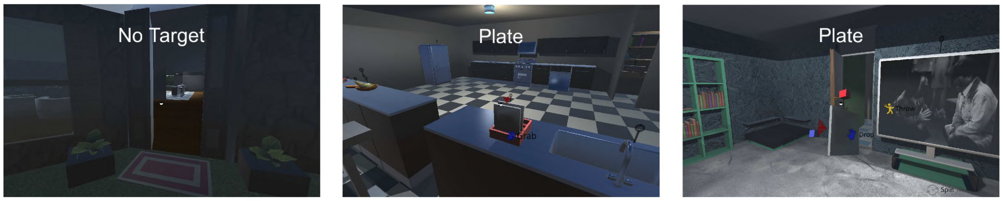


Figure 16: Screenshots of the 3DPass Application. (The entrance to the home, where users start. (left) The kitchen. The user has turned on the lights, and is now targeting a plate to pick it up. (center) The children's bedroom. User has turned on the television and lights and is rotating the held plate. (right))

Figure 16 shows several examples of the 3DPass system. In order to sustain immersion, the GUI of 3DPass is kept minimalistic. A dot in the center of the screen helps the user to target objects by aligning the dot with the object. The object must be within "arm's reach" of the player avatar to be targeted. Small vertical black lines with a circle on top indicate an object can be interacted with. When an object is targeted, two perpendicular red triangles replace this black line, and the name of the currently targeted object is displayed at the top of the screen. A context menu appears when an object is targeted showing the user what actions can be taken, for example drop and throw. The colors and locations of the context menu correspond to colors and locations of buttons on the Xbox 360 controller, for example drop is the left button on the controller, and this button is blue on the controller itself. Held objects are carried in front of the avatar until dropped or thrown. No other objects can be targeted while an object is already held. When rotating an object, colored cubes appear to assist the user in determining what direction they are rotating in. The blue and red cubes correspond to the x and z axis respectively. Users enter rotation mode by holding a button, and rotate objects using an analog stick. The x axis corresponds to horizontal movements on the analog stick, and the z axis is controlled by vertical movements. Movement along the y axis can be achieved by combining movement along the x and z axises.

Users generate a 3Dpassword by performing a set of actions and navigations. Figure 16 can be considered an example 3Dpassword. The user enters the kitchen and turns on the lights. Next the user goes to the children's bedroom and turns on the lights and television. The user returns to pick up a plate from the kitchen and takes it to the children's bedroom. The user stands by the couch in front of the TV and rotates the plate until the angle looks like it matches the door to the left of the TV.

Users authenticate themselves later by repeating the same actions and navigations exactly, within some tolerance for distances and angles. In other words, the user must recreate an experience that they had.

4.5.1 Input Device

Participants interact with 3DPass using an Xbox 360 controller.

While mouse and keyboard may be a viable approach for some users, many users would potentially struggle at using these devices without being able to see them, and finding the mouse and keyboard while effectively blindfolded with an HMD is not an easy task for anyone. Isokoski and Martin [106] find that a 360-style controller performs on the same level as a keyboard+mouse+eye tracker combination at aiming tasks. Davidson [107] finds that the controller performs better than gesture tracking technologies such as the Kinect and on par or better than mouse and keyboard in terms of user enjoyment, mental and physical fatigue incurred, and overall ease of use. Ardito *et al* [108] find that the controller is less error prone for various VR interaction tasks than mouse and keyboard or the WiiMote, roughly on par with mouse and keyboard in terms of input speed, and superior to both mouse and keyboard and the WiiMote in user preference and perceived difficulty.

The Leap Motion was originally considered as an alternative, since it has direct hand tracking. Coelho and Verbeek [109] found that the Leap Motion is slower than mouse and keyboard for various simple input tasks, so by extension technologies like the Leap Motion will likely be slower than the controller.

The controller's primary weakness is aiming, which requires precise use of the analog sticks. HMD users of 3DPass can use head tracking to aim at objects by looking at them with their heads rather than using the analog sticks. The combination of a controller for movement and interaction tasks and an HMD for aiming tasks is hypothesized to be best combination for low entry times, though testing this will be left to future work.

A mobile version of 3DPass is planned for future work, however the current version needs relatively high end hardware to run, requiring a PC with a high end GPU. In a mobile implementation, users could potentially use the phone as the display, for example with a holding device such as Google Cardboard, while using a bluetooth-based controller for input. The iPEGA 9021 is one such controller that is very similar to the Xbox 360 controller in terms of looks and functionality.

Although an attacker could watch the controller during input or intercept bluetooth/usb signals to see what actions a user takes, predicting where the user moves by observing input is still a challenge. Unlike traditional binary input such as keyboard keys, movement with analog sticks depends on how far the stick is pressed in a direction, typically using a potentiometer. The attacker would need to know how long and what angle the analog stick is held during the entire input. Adding minor variations in user start position and object start positions, which may be so minor as to not even be noticeable to the user, can further conflate attempts to intercept the password based on input.

4.5.2 Design Considerations

3DPass utilizes the psychological and physical advantages described in Section 4.4 by adhering to several design considerations.

1) Familiar Environment: The environment is chosen to be familiar to as many participants as possible. Since this scheme is targeted at the general population, a home is used as the environment. The building plan used as the basis of the 3DPass design is the best-selling plan several years running from a popular house plans vendor [110]. A familiar environment allows for faster learning as participants already know what can be expected inside. For example, participants know that in a house, there is a bathroom, with a sink that can be turned on and filled with water. The master bedroom is expected to have a bathroom attached, and the dining room is expected to be near the kitchen. Matching user expectations may improve presence and facilitate faster learning of the environment for navigation. By making the environment familiar, users can focus on what makes it different from other similar environments they've seen before (e.g., other houses), thus allowing users to begin forming episodic memories and establish context in the environment sooner.

2) Multiple Contexts: The environment should have multiple, distinct areas, where users can establish context. 3DPass is split into rooms which have distinctly colored walls, distinctly patterned floors, and other distinguishing features. For example, a user realizes they are in a bathroom context when the floor and walls are tiled, and there are no windows. 3Dpasswords generated in different contexts may suffer less memory interference.

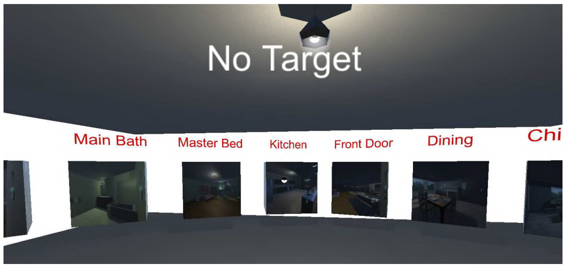


Figure 17: Teleporter Room. (Users are instantly transported to this room by pressing a button. Users can go to any room by walking into the pictured cube that corresponds to the destination.)

3) Quickly Navigable: For usability purposes, the environment should be quick to navigate. 3DPass accomplishes this in three ways: (1) the environment itself is relatively compact, and can be crossed at its longest path in about 15 seconds. All doors between rooms are open by default, (2) Users are able to engage a "speed walk" function by holding a button as they walk, allowing them to pass through any solid object (including walls) and move extremely quickly. The environment can be crossed in about 3 seconds using this function, and (3) Users can press a button to be transported to the "teleporter room", demonstrated in Figure 17, a separate area where all major rooms in the environment are available for fast access.

Humans tend to navigate an environment by remembering layout and landmarks, though landmarks appear to take priority when present, and have a larger impact on navigation performance [111]. 3DPass utilizes landmarks to improve navigation– each room is decorated in a unique way, and has stand-out objects such as large screen television in the living room or a luxury car in the garage.

4.6 Security Strength of 3D Authentication

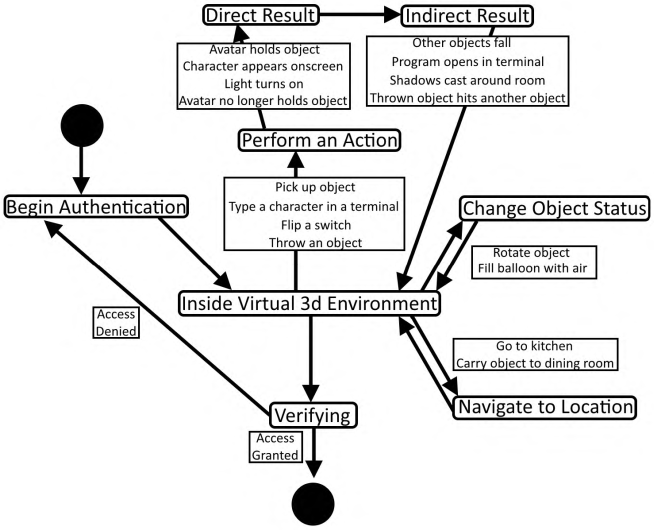


Figure 18: State Diagram for a 3D authentication Scheme. (Statements in square boxes are examples. Rounded boxes indicate states which can be used for authentication.)

As in prior work by Alsulaiman and Saddik [83], it can be useful to express potential actions during the authentication process in the form of a state diagram. Figure 18 presents a modified version of the state diagram for this work's definition of a 3D authentication method. Note that *typing a textual password, performing a graphical password*, and *performing biometrics* in Alsulaiman and Saddik's work in all fall under the category of an "action" by this definition.

An action is any direct input to an object in the environment, including picking up an object, pressing a button, or presenting a biometric token. A direct result is the direct consequence of an action. For example, if a book is dropped, it falls due to gravity, and is no longer held by the user. An indirect result may occur elsewhere in the environment. For example, the book may strike a vase, and the vase may fall. In this chapter, we measure the password space only of direct results. The password space of indirect results is exponentially larger.

The user can also change the state of a held object, for example by rotating it or stretching it. We separate this from an action because the user possess, controls, or is otherwise engaged with the object, and changes to the object's state while it is held do not have any direct or indirect effects on the environment. If a state change effects the environment, it can be considered an action instead.

4.6.1 Password Space of 3DPass

Let us denote the number of objects in the environment which can be picked up and held as N. The number of modifications that can be made to a currently held object, for example rotating it to an angle or stretching it, will be denoted as M. For simplicity we will say M is the same for every object which can be held. We will denote the number of locations a held object can be placed as L, where the size of L is the usable area of the environment divided by some tolerance value. A location can be considered as an xyz coordinate with some boundaries or tolerances, for example $(2 < x < 4)$, $(1 < y < 3)$, and

($3 < z < 5$) for a location at (3, 2, 4) with a tolerance of 1 unit in any direction.

The number of interactions the user can have with objects that does not require picking them up, for example typing a key, turning on a light, or setting a toaster, will be denoted as $N_{interactable}$ * I_N, where $N_{interactable}$ is the number of interactable objects in the environment, and I_N is the number of interactions for each of those objects. Some objects in $N_{interactable}$ may also be in N. Lastly, we denote the the number of locations the user's avatar can move to as $L_{navigable}$. The size of $L_{navigable}$ will be smaller than L, as there will be locations where objects can reach but the user's avatar cannot travel, for example inside a drawer.

At any point, the user has the following choices: (1) Grab one of N objects, (2) If an object is held, modify the object in one of M ways, (3) If an object is held, put it in a new location in L, (4) If nothing is held, perform one of I_N interactions on one of $N_{interactable}$ objects, (5) Move, or *navigate*, to a location in $L_{navigable}$. An attacker will need to try all combinations of those 5 "choices" in order to bruteforce all possible 3Dpasswords.

For simplicity, assume that if the user is already holding an object, they can still pick up another, and that the user is holding an object on startup. If T is length of the password in choices, the password space is then equal to the sum of these choices to the power of T.

$$(N+M+L+(N_{interactable}*I_N)+L_{navigable})^T \tag{4.1}$$

Equation 1 is an upper bound, as we assume that users can pick up objects even when they are already holding one, and that users can modify or relocate objects they haven't yet picked up. If we assume that the user cannot pick up or interact with a new item while already holding one, then in practice, the user has

$$M+L+L_{navigable} \tag{4.2}$$

choices when holding an item. And when not holding an item, the equation becomes

$$N + N_{interactable} * I_N + L_{navigable} \tag{4.3}$$

In 3DPass, there are 327 objects ($N = 327$) that can be picked and 115 interactable objects ($N_{interactable} = 115$) that cannot be picked up. For simplicity, let us assume that $I_N = 1$, though in practice many objects in 3DPass have multiple interactions. For example, most doors can be opened or closed in an approximately 110 degree arc, and the fireplace can be ignited, turned up, or reduced until extinguished.

3DPass simulates approximately 3000 ft^2 (roughly 279 m^2) of indoor living space excluding walls, doors, and other natural barriers. We will use area instead of volume because the player avatar walks on mostly flat terrain and most interactions take place only on one surface of objects (e.g., the top of a table), barring some exceptions such as placing an item on a shelf with multiple levels. Let us assume a generous location tolerance of 3 ft^2 (roughly 1 m^2) for a lower bound of ($L = 300$). That is, there are roughly 300 unique locations in 3DPass. The navigable space is slightly lower than L due to furniture which the user's avatar cannot climb on. Again as a lower bound, let us estimate $L_{navigable}$ is roughly 80% of L, so $L_{navigable}$ is roughly 240.

Held objects can be modified by rotating them. Rotation along the y axis can be accomplished by combining x and z axis rotations. Rotation was used by only 16% of participants in the user study, and no other modifications for held objects are available, so we will set $M = 0$ for purposes of calculating a realistic lower bound.

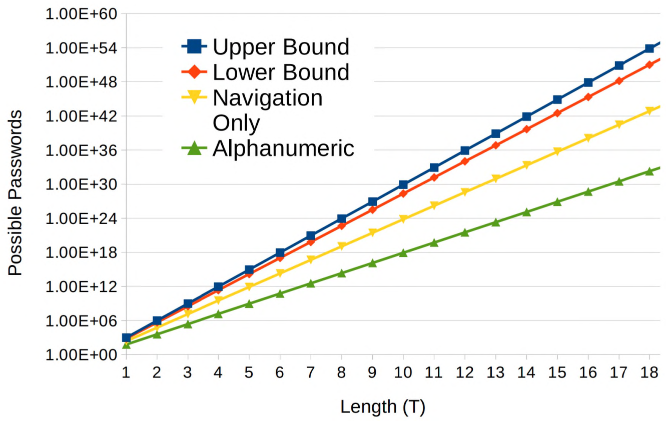


Figure 19: Number of Possible Passwords Using: (1) Equation 4.1, (2) Equation 4.3, (3) Navigation-only, $(L_{navigable})^T$, (4) Traditional Case Sensitive Alphanumeric Password (62^T).

The password space of 3DPass, according to Equations 1 and 3, is plotted in Figure 19. Equation 1 represents an upper bound for 3D authentication schemes, while Equation 3 is a lower bound for 3DPass, since there are less available choices when not holding an item. Equation 3 assumes that picking up an item does not increase available choices, though in practice when an item is picked up, the next choice will follow Equation 2 instead. Both the lower bound and upper bound on the password space of 3DPass is well in excess of the standard alphanumeric approach. Even using very pessimistic lower bound estimates for I_N, L, $(L_{navigable})$, and M, 3DPass has a robust password space. A 3Dpassword with just 5 choices exceeds the password space of an 8 character alphanumeric password. 3DPass with 8 choices has roughly the same password space as a 14 character alphanumeric password.

In the user study, 2 participants made 3Dpasswords using only navigation, with no actions. Though the theoretical password space of a navigation-only password is still quite large, as plotted in Figure 19, but in practice the password may be vulnerable to hotspot analysis. In a full implementation, it would be recommended to enforce at least 1 object grab or interaction minimum per password, resulting in the password space indicated by the lower bound calculation instead. Also, in a 3Dpassword with a series of navigations, order must matter, the user must indicate when they have reached the desired location somehow, and wrong locations must reset the 3Dpassword. Otherwise the attacker merely needs to try $(L_{navigable}) * T$ combinations (going to each location in the environment T times), which is very small. Determining when to reset progress or mark an attempt as invalid, since there are by nature many valid paths to any destination, is a difficult challenge.

4.7 3DPass User Study

The experiment was conducted using a standard 24-inch widescreen monitor with 1920×1080 resolution and an Oculus Rift HMD. Interactions with the environment are recorded by the scheme directly and by video screen recording of the sessions. Participants interact with the environment using an Xbox 360 controller, and using head tracking if assigned to the HMD group.

Participants were recruited using fliers and word of mouth. A ten dollar cash incentive was offered for completing the experiment. To encourage participants to remember their passwords, a $50 prize pool was raffled between participants who remembered all of their passwords. A total of 20 participants were recruited to test the memorability and usability of 3DPass. Participants were 25% female, mean age 23 (stdev=4.5, range 17-32). Most participants (60%) answered yes when asked if they play 3D video games at all, and 20% of participants answered yes when asked if they had ever used VR before. On a scale from 1 (Strongly Disagree) to 5 (Strongly Agree), participants responded to the statement “I am skilled at using an Xbox controller or similar” with an average rating of 3.89. Four participants were completely new to using a controller and had never used one (rating of 1). Excluding those ratings, the average response is 4.53.

In order to evaluate the impact of stereo vision, head tracking, and motion parallax on presence, entry times, and usability ratings, participants were grouped into one of following two conditions.

VR: Participants in this condition used the Oculus Rift HMD for the entire experiment.

Monitor: Participants in this condition used the monitor for the entire experiment.

Participants were grouped at random, except for three participants with large glasses who requested to be placed in the Monitor condition due to potential discomfort wearing the HMD. Two additional participants requested to be placed into the Monitor condition because of concern over nausea associated with VR. In total, there were 11 participants in the VR condition and 9 in the Monitor condition.

4.7.1 Procedure

Unlike CMAPS and PassGame, the design of this user study was particularly interested in determining memorability advantages associated with using 3D versus conventional memory for passwords. This study is conducted using alphanumeric passwords as a control.

Once again, the experiment is conducted in a closed laboratory environment. Participants come in for two sessions, one week apart, taking roughly 20 and 10 minutes respectively. A 10-hour time window is offered for participants to come in for each session, so all 20 participants were not tested at the same time of day.

The first session is conducted according to the following procedure.

1) Participants fill out a form with demographic information such as age and skill with controllers.

2) Participants read a brief description of the 3DPass scheme. The instructions asked them to treat their 3Dpassword as if it were a real password- to try to make it secure, but also fast to enter. Participants were given some text examples of 3Dpasswords using objects, navigation, and rotation, and told that they could make their passwords as absurd or realistic as they wanted.

3) Participants are presented with the 3DPass implementation. Participants are introduced to the controls and take some time to practice using the scheme. Most participants practiced for under five minutes.

4) The environment is reset for participants to set the first 3Dpassword.

5) Participants are asked to generate a standard alphanumeric password with some rules: case-insensitive, at least 8 characters, and not used previously. The alphanumeric password did not have to relate to the 3Dpassword, though several participants chose to do so.

6) Users fill out a questionnaire about their 3Dpassword, and were asked to describe the steps of the 3Dpassword in plain text, with accompanying drawings if desired. These descriptions are used along with video screen recordings of the session to verify that the user correctly remembered the 3Dpassword in the second session.

7) Users repeat steps 4-6 once more, creating an additional 3Dpassword and an additional alphanumeric password.

8) Participants recall their passwords, in the same order they were set. During this time, any discrepancies or vague wording in the user's written description of the password were cleared up. For example, if a participant wrote that their password was to "pick up the bottle", the description was clarified to exactly which bottle was meant. If the intent was to "pick up any bottle in the kitchen", for example, then that was specified. If a user failed to remember an alphanumeric password at this stage, they were asked to return to step 5, count backwards from a random 4 digit number in 3's for 30 seconds, and recall the new password. No users failed to remember a 3Dpassword after setting it.

In the second session, participants returned to the laboratory environment after one week to recall their passwords. Passwords could be recalled in any order, and users could practice in the environment beforehand to relearn the controls if they chose to.

After learning from issues with measuring CMAPS and PassGame entry time data, this experiment bases timing data off of multiple averaged attempts conducted after a small amount of practice. After recalling the 3Dpasswords correctly, participants were asked to re-enter each 3Dpassword an additional 3 times. Timing data was based off these attempts only. Because users already recalled their passwords earlier, time spent thinking and remembering is less impactful on entry time.

4.7.2 Memorability Results

Hypothesis H2 assumes that 3DPass with the HMD will have the greatest memorability, followed by 3DPass with a monitor, and lastly alphanumeric passwords.

Table XIII: Recall Rates of 3DPasswords and Alphanumeric Passwords (one week after initial setup).

Condition	Passwords	Recall	Recall Rate
VR	22	21	96%
$Monitor$	18	18	100%
$Alphanumeric-VR$	22	15	68%
$Alphanumeric-Monitor$	18	18	100%

Table XIII presents recall rates for the two 3DPass conditions and standard alphanumeric passwords. Contrary to the expectation from H2, there was no significant difference in memorability between the VR and Monitor conditions ($\chi^2 = .839$, $p = 0.360$). A larger sample size may be needed. However, as expected, 3Dpasswords are significantly more memorable than alphanumeric passwords. A McNemar Chi-Squared test for VR vs alphanumeric reveals significant difference ($\chi^2 = 10.56$, $p < .001$). Interestingly, all 7 forgotten alphanumeric passwords belonged to the VR group, possibly implying that going from memorizing a VR environment to memorizing a traditional alphanumeric password has some impact on user memory. This phenomena may support the idea that VR authentication should take place in VR context rather than outside it, and the consistency of this phenomena is something planned for investigation in future work. No 3Dpasswords nor alphanumeric passwords were forgotten in the Monitor condition.

Users sometimes added extra steps to their 3Dpasswords, usually caused by interference with their other 3Dpassword. For example, one user had to go to the car in the garage at the end of their 3Dpassword, but ended up going to the car in the garage at the end of both 3Dpasswords. However, since extra actions or navigations do not nullify the correctness of a 3Dpassword, particularly after the 3Dpassword is already complete, they had no impact on memorability. Future work will investigate memory interference and the impact of allowing extra actions in 3Dpasswords.

4.7.3 Usability Results

Presence

Table XIV: Presence Survey Results of 3DPass (Statements are scored on a Likert scale from 1 to 7, **where a higher score always indicates more presence**. VR results are presented first (white), then Monitor in alternate rows (filled). Statements are shortened for length.)

Statement	Mean	SD	Med
How real did virtual world seem	4.73	1.35	5.00
(Monitor)	4.56	0.97	5.00
How consistent with real world	4.09	1.45	4.00
(Monitor)	4.89	1.76	5.00
I did not feel present	5.45	1.63	6.00
(Monitor)	5.11	1.27	6.00
Was not aware of real environment	4.18	1.60	4.00
(Monitor)	3.22	1.56	3.00
Sense of being there	5.73	1.19	6.00
(Monitor)	5.78	0.67	6.00
The virtual world surrounded me	5.36	1.21	5.00
(Monitor)	5.11	0.93	5.00
Captivated by virtual world	4.73	2.00	5.00
(Monitor)	4.33	1.32	5.00

Presence was evaluated using 14 questions from the Igroup Presence Questionnaire (IPQ) [112]. Table XIV show the results of some of the IPQ questions for both conditions. Contrary to initial expectations, the mean scores for both monitor and VR conditions were nearly identical, and Mann-Whitney analysis of the scores showed no significant different in presence between the two conditions. This is consistent with the memorability results for both conditions, and supports the conclusions of hypothesis H2.

Some works have established a link between stereoscopic displays or HMDs anda higher reported level of presence [102]. However, other works have failed to support this relationship, or found that traditional monoscopic displays can evoke even more presence than HMD counterparts [88, 87]. Banos *et. al* [88] demonstrated that older HMDs actually elicited a lower feeling of presence in some contexts than large 2D displays, possibly due to user discomfort with HMDs. Based on the results of the presence questionnaire, this experiment finds similar results. There are several possible explanations, but one possible culprit is the high resolution and larger size available on modern monitors. The large screen used in Banos' work had a resolution of 1024×768, but the monitor in this experiment has a resolution of 1920×1080. Perhaps a large screen is just as immersive as an HMD.

Banos also noted that participants reported more negative effects with the HMD (nausea, dizziness, etc). Participants were asked to rate the statement "I felt ill while in the virtual world" and there was no significant difference between the VR and Monitor conditions (mean response 2.73 and 2.56 respectively, $Z = .038, p = .968$). However, two users who were originally randomly assigned to the VR condition did have to switch to the Monitor condition due to feeling ill almost immediately after putting on the HMD (both users finished the experiment in the Monitor condition and filled out survey responses for the Monitor condition only). If these users had scored for the VR condition instead, that may have resulted in a significant difference, however these users were not permitted to score for the VR condition as they completed the experiment in the Monitor condition.

Entry Time and Usability Survey

Timing data was collected by the application and confirmed by reviewing video screen recordings of the sessions. Time begins counting when the user first moves and ends when the user performs the last action or navigation that makes up the 3Dpassword. In the second session, after verifying that they remembered their 3Dpasswords, users were asked to input their 3Dpasswords an additional 3 times each. Timing data was based on those 3 inputs only.

The average entry time for the VR and Monitor conditions was 20.96 and 25.93 seconds respectively. A Mann-Whitney test indicates there was a significant difference between the conditions ($Z = -2.05, p = .040$), supporting hypothesis H1. The likely explanation is that head tracking allowed for significantly faster aim than using the analog sticks of the controller, allowing some users to shave several seconds off their authentication times. Because presence results indicate that both conditions were equal, it is unlikely that presence was a major contributor to the difference in entry times. The fastest users in each condition required 8.67 and 11.00 seconds respectively.

Four participants (two in each condition) were not included in timing results because they had never used a game controller before and were still training to use the controller as an input device during the experiment. The average entry time for these participants in the VR and Monitor conditions was 56.08 and 63.25 seconds respectively.

Table XV: Usability Survey Results of 3DPass (Statements scored on a Likert scale from 1-Strongly Disagree to 10-Strongly Agree. Some statements shortened for length.)

	VR			Monitor		
Statement	Mean	SD	Med	Mean	SD	Med
Creating a password was easy	7.91	2.17	8	9	1.32	10
Logging in was easy	7.55	2.38	8	8.56	1.67	9
Remembering password was easy	8.73	1.62	10	8.78	1.39	9
Faster than alphanumeric	5.27	2.87	5	4.00	2.69	4
With practice, would be fast	8.82	1.99	10	9.44	0.73	10
The scheme was fun	9.27	1.10	10	10	0	8
Prefer the scheme to alphanumeric	6.91	3.21	7	7.56	2.40	8

The results of the Likert survey for usability are presented in Table XV. Mann-Whitney comparisons between the two conditions found no significant difference between the two conditions for any survey response. Again, this matches expectations since no different was found in presence between the two conditions. About half of users agreed that the 3DPass scheme was faster than a traditional alphanumeric password, and about 70% prefer 3DPass to conventional passwords. Almost all users agreed that 3DPass was fun. The survey results indicate that many users enjoy 3DPass and would consider it as an alternative to traditional passwords.

4.7.4 Hotspots

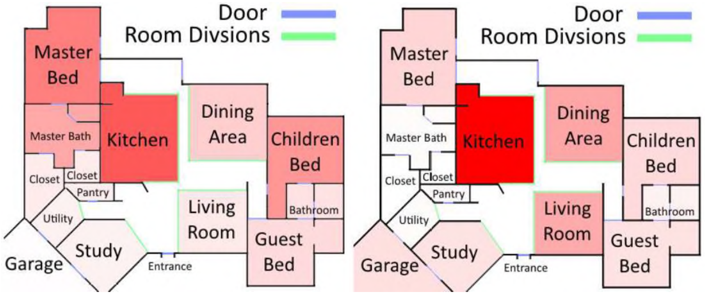


Figure 20: Distribution of Objects in the 3DPass Environment (left) Actual Usage of Environment by Participants (right). (One use was counted if an object was picked up, dropped, thrown, or interacted with at that location, or if the user specifies navigation to that location. Doorways separating rooms are marked with blue lines. Green lines indicate the logical end of a room that has no physical barrier. Due to the “loft style” arrangement of the center of the house, the kitchen, living room, and dining area are all actually one continuous space.)

Figure 20 shows the distribution of objects in the environment and the distribution of objects and locations utilized by participants in the user study. As expected, the 3DPass environment itself has several areas of clustered objects, for example the kitchen, where it was natural to expect more objects than other rooms in the house. Participants were more likely to use objects near the entrance to the home, but locations all over the environment were utilized. Despite the garage having just one interactable object (the light switch), mostly just decor (e.g., cars which were not interactable), many participants chose to use this room in their 3Dpasswords. An attacker attempting to brute force 3DPass may have difficulty determining which areas are most popular. The kitchen was used over three times as much as the dining area, the next most used room, (118 uses for the kitchen vs 35 for the dining area). However, 74% of participants used 2 rooms or greater, and the other rooms used were significantly less predictable.

4.7.5 User Choice in 3DPasswords

To better understand what kind of 3Dpasswords users would set, the 3DPass scheme does not impose any requirements, and successful authentication is determined manually by examining the user's explanation of their password and video of the initial session against data logged by the scheme and video recordings of the user's login sessions. In the user study, after setting a 3Dpassword, users write down a description of the 3Dpassword and answer several questions about their 3Dpassword.

From password descriptions, 84% of participants indicate that order matters. Objects are used by 95% of participants, and locations are used by 61% of participants. Among those who use locations, 40% use locations that can be defined statically (e.g., xyz coordinates), while the majority prefer locations that can be defined in relative terms (e.g., in the pan, on the counter). Rotation was used by 16% of participants, though all but one of these involve rotating the player avatar to face some angle rather than rotating an object. For example, one participant grabbed a soda, shook it by shaking their head up and down with the HMD, and released it at what they described as "a 40 degree angle", which was interpreted as 40 degrees up on the vertical axis from origin. Time, i.e., waiting a real-world amount of time for some action, is used in only two 3Dpasswords (5%). Based on user feedback, a full release of 3DPass should enforce order, allow (and encourage) both use of objects and navigation, and allow rotation for a still-substantial minority of users who may use it.

Participants created 3Dpasswords with an average of 1.85 objects grabbed, 3.89 interactions (including drop/throw), and 3.11 locations which were part of the 3Dpassword. Thus the average password length is about 9 choices, the equivalent of a 15 character alphanumeric password using Figure 19. Only 21% of locations are entirely independent from any action in the 3Dpassword. For example, a 3Dpassword may have "go to the kitchen" followed by "open the fridge", but the kitchen location is irrelevant here because travel to the kitchen is required to open the fridge. On the contrary "open the fridge" followed by "go to the garage" makes the location an independent part of the 3Dpassword.

Most frequently, a location corresponds to a drop or throw, for example “go to the garage” is followed by “throw the book into the car”, in which case the location is mostly independent, since it is relevant to where the action takes place. Omitting locations entirely, the average 3Dpassword is still about 6 choices long. With 6 choices, 3DPass still exceeds the password space of a 10 character alphanumeric password.

4.8 Discussion of 3D Authentication

Based on the results of the user study, 3DPass is comparably better in terms of memorability than a traditional alphanumeric password. In security, 3DPass appears to have a clear advantage, but several factors may complicate accurate calculation of password space. For example, objects in N may be destroyed so that N shrinks over time. In 3DPass, the world has a discrete edge, and users could throw off objects off the edge of the world where they would no longer be recoverable. The size of N could also increase, if for example, one loaf of bread is broken into two halves. Interactions can vanish or appear as well, for example if the batteries are removed or replaced from a remote control. In total, none of these factors should have any significant impact on password space, since a change in one or two objects has relatively little impact on the overall total.

Certain tasks where precision was assumed be helpful, for example opening doors, were merely time sinks. Participants either opened doors, drawers, and cabinets enough to use them, or shut them fully. Adding a button which instantly fully opens or shuts a door, drawer, or cabinet could improve average entry time by several seconds, as some users that utilized doors in their 3Dpasswords spent up to 15 seconds just opening and closing doors. Since we assumed that the number of interactions per object was 1, the ability to open doors to different levels had no bearing on password space calculations.

Using rotation may present a challenge for certain objects which have internal symmetry. For example, the plate shown in Figure 16 can be considered as a square which looks identical at 0, 90, 180, and 270 degree angles. A full implementation of 3DPass would need to recognize when objects with internal symmetry are at rotations that result in an identical image or avoid objects with symmetry altogether, otherwise a password may be rejected if an object is at, for example, 270 degrees instead of 90, even though both angles look identical to the user.

Surprisingly, there was no detectable advantage in using an HMD vs a monitor in terms of presence, memorability, or reported usability. A plan for future work is to find a VR mechanism with superior presence, in order to examine the impact of increased presence on memorability and usability.

Several technological advancements may make 3DPass entry times even more favorable. Hand/body tracking and naked-eye 3D technologies may simplify user input and allow users to navigate the environment faster and more intuitively. Presence and entry time will likely improve when users can pick up an object using a gesture rather than pressing a button, though current hand tracking technologies such as Leap Motion may not have enough precision [109, 113]. Rotation will likely be more favored with these input devices than with a controller, as achieving 3-axis rotation with two analog sticks is difficult. Eye tracking inside VR, provided by some manufacturers such as Tobii or FOVE, may improve aim speed even further, allowing users to target objects without even moving their heads.

Although entry times were improved by using the HMD, only a few VR participants actually used head tracking to aim, with most opting instead to keep their heads in roughly one place for the entire experiment. Greater utilization of head tracking can lead to improved entry times (via improved aiming due to head tracking), and improved depth perception (via motion parallax). As users become more familiar with HMDs, and in turn actually utilize head tracking, entry times and presence scores may improve. A future work will be to repeat the experiment with experienced HMD owners. The few participants who utilized head tracking most were also among the fastest participants.

CHAPTER V

BEHAVIORAL PASSIVE AUTHENTICATION

5.1 Outline

This chapter is based on a work presented at the 2016 International Conference on Human Aspects of Information Security, Privacy, and Trust (HCII 2016) [114].

The concept of passive behavioral authentication is introduced in Section 5.2. Related works in authentication via typing behavior are discussed in Section 5.3. A novel passive behavioral authentication method based on typing, utilizing a wider variety of data than previous schemes, is introduced in Section 5.4. A user study is conducted, and the results of using several novel classification methods are shown in Section 5.5. Conclusions and future work are presented in Section 5.6.

5.2 Introduction to Implicit Authentication

Previously, the proposed authentication mechanisms have all been active, securing the device at initial point-of-entry. In this chapter, the consideration is shifted to *passive authentication*, a constant in-the-background process of authentication that seeks to capture attackers which have already broken through the primary authentication method.

The scheme proposed in this chapter is likely the least “novel” of all those proposed in this work. Keystroke dynamics, the mechanism of using typing behavior to identify a user, has been tried and done many times before, including in a number of commercial applications, and even on the mobile platform. In fact, keystroke dynamics traces its origins all the way back to World War II, when telegraph typists were identified based on the manner in which they typed simple dots, dashes, and stops.

The novelty of the passive authentication scheme in this chapter is twofold: (1) Using the concept of MAPS, the authentication method in this chapter combines not just traditional dimensions of typing like timing information, but also other dimensions such as location and acceleration, and (2) the results of using this larger amount of information in a pilot user study are extremely encouraging in terms of accuracy.

5.3 Related Work: Implicit Authentication

Once an attacker breaches the unlock authentication mechanism on a mobile device, they are usually free to read sensitive information and use the device as their own. Although many users lock their devices, many still do not. As a result, a large market exists for the theft and resale of mobile devices. This work proposes a hassle-free second line of authentication that secures a mobile device against intruders who have already unlocked the device and started to use it.

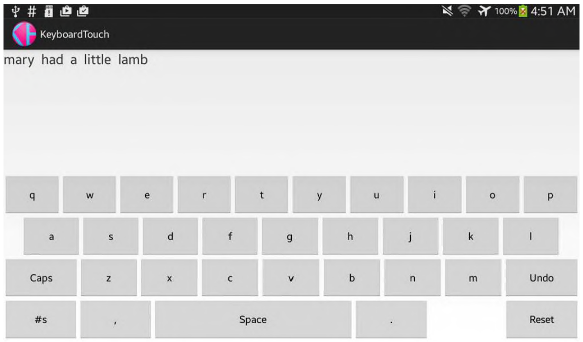
Implicit authentication identifies users passively without any explicit input from the user such as a password entry. Physical biometrics such as gait [115] can be used as implicit authentication, however the attacker can merely refrain from walking while the device is powered on. Likewise voice recognition schemes, such as [116, 117], can be thwarted by staying silent. The only task that’s absolutely necessary to the use of a mobile device is interaction with the touchscreen, which is required to do anything from placing calls, to writing text messages, to accessing secure accounts. A *behavioral* biometric based on some aspect of the user's touchscreen behavior rather than physical characteristics, will be difficult for the attacker to avoid if they wish to use the device.

Various authors have proposed authentication schemes based on touchscreen gestures and readings from on-device sensors during a touch [118, 119, 120, 121, 122, 123, 124, 125]. Unlike previous works such as Feng *et al.* [125], which identify users based on touch screen gestures, this work focuses specifically on identifying users based on typing patterns on the soft keyboard. Most users regularly type small amounts of characters on their phones, for example text messages, emails, and phone numbers. An attacker that steals the device and attempts to use it for any of these tasks will be identified and locked out, potentially triggering a warning to location schemes like Apple's "Find my iPhone." The scheme can also provide a secondary line of security for passwords input on the mobile device. If an attacker knows the user's password and attempts to input it[1], the entry may be detected as unauthorized, triggering an account lockdown or some verification step.

Other works such as Draffin *et al.* [123] have utilized touch behavior in the past to authenticate users. The scheme presented in this work utilizes all features that can collected on modern devices, including acceleration data, for a higher degree of accuracy than previous works. Several approaches for classifying touches are also presented, all based on simple statistical classifiers than can be run on the limited hardware of a mobile device. The results show that some approaches can achieve a 97% rate of accuracy identifying one user out of a pool of 15 after only 15 touches.

[1] We assume that the account utilizes trusted devices, and the attacker cannot simply enter the account information on another device

5.4 Implicit Biometric Authentication Scheme



mary had a little lamb

Figure 21: Screenshot of the Android Keyboard Implementation.

The built in soft keyboard on Android systems intentionally disallows recording of touch information to avoid various misuse such as keylogging. An application was created to emulate the soft keyboard rather than circumventing this security feature. A screenshot of the application is presented in Figure 21. The Android class MotionEvent is used to collect data from touches on the keyboard buttons, and the SensorEvent class is used to collect accelerometer data. For each touch, the following data is collected.

Duration of touch and time since last touch: The use of these values by themselves is sometimes called *keystroke dynamics*. The duration of touches and time between touches is recorded in milliseconds. The duration of a touch is considered the time between the press and release of a button (eventtime-downtime in the MotionEvent class). The time since the last touch is considered the time from one press to the next (downtime-previous downtime in the MotionEvent class). For the first touch in every trace, the time since last touch is set to zero.

Relative x and y location of press: The location of the center of the touch at the time the button was pressed, relative to the button, is recorded in pixel units. The top left corner of any button is (0,0) and the bottom right corner is the maximum, which varies by device.

Size of touch on press: The size of the touch is recorded on a scale from 0 to 1, where 1 is the maximum touch size the system will recognize, which varies by device. The system interprets all touches as a circle where size determines the radius of that circle. Size of a touch roughly correlates with finger size and touch pressure, which can be used to identify a user. The number of sizes supported by each device is not infinite; most devices support between 20 and 100 discrete touch sizes.

Magnitude of acceleration on press: The magnitude of acceleration is read in m/s^2 from the accelerometer, a sensor which the vast majority of devices on the market have today. Different devices update their sensors at different rates. We take the last known accelerometer reading before the press occurs, which may be several milliseconds before the touch itself.

Relative x and y location of release: The location of the center of the touch as the the button is let go. By taking the x and y locations of the press and release, we can determine how far, and in which direction, the user moves their finger during a touch. Touch distance or direction are not used directly because we are seeking to minimize computation requirements, though an intelligent classifier may utilize these features indirectly.

Size of touch on release: The size of the touch as the button is let go. Pressure can be inferred from the difference in sizes between press and release. If the press size is very large, and the release size is much smaller, then it is likely the user pushed their finger down hard on the device and increased the surface area that was making contact with the screen. Pressure is not calculated directly in this work, though a classifier may indirectly take advantage of this relationship. Physical properties of the finger may also play into this feature, for example the amount an individual's finger yields when making contact with a hard surface.

Magnitude of acceleration on release: As with a press, the magnitude of acceleration on release is the last known accelerometer reading at the time the user releases the button. If a touch is short enough and the device's sensor is slow to update, this value may be the same as the value for the press. The difference in acceleration between press and release can be used to infer how hard the device was touched or how steady the user's grip is. A large difference can indicate a hard touch which pushed the device down and and caused it to recoil back to its starting position. No effort is made to process acceleration in this work, however the classifier may make such inferences indirectly.

Maximum, minimum, and average acceleration during the touch: Acceleration is read as many times as possible between the press and release of a touch, recording the max, min, and average of the magnitudes of acceleration. Depending on the speed of the touch and the rate at which the accelerometer updates on that particular device, it is possible to obtain zero readings between the press and release, in which case we set all three values to some default value.

Pressure, although reported by the MotionEvent class, is either 1) set to a default value which does not change or 2) scaled linearly with touch size. Capacitive touch screens used by most modern mobile devices are not able to sense pressure directly, but instead infer it from the touch size. Some devices can read pressure using a pen device with a pressure sensitive tip, and manufacturers like Apple and Huaewi are developing phones that have pressure sensors behind the capacitive glass. Many previous works that utilize pressure are actually double-counting size.

5.4.1 Future Implementation

In an ideal case, the scheme collects touch data anonymously from a large pool of users and distributes the anonymous data between the users somehow, possibly with a peer-to-peer approach. Periodically, each user updates the touch data they compare themselves against with data from other users. Typing behavior is analyzed by the classifier and locks the device or prompts for additional verification if the touches do not appear to belong to the user for a certain length of touches. The user will have a high probability of matching their own touch behavior, while the attacker has an equal chance of matching any of the other users in the classification pool.

For example, with 15 users, an attacker who has an equal chance to be behaviorally matched to any of those users has a $1/15 = 7\%$ chance to be authenticated as the user per attempt. If three consecutive failures cause a lockout, the attacker has a $14/15^3 = 81\%$ chance to be locked out in the first three authentication attempts.

The scheme should change the data that it compares against the user frequently in order to reduce the odds that the user will be mistook for another typist with similar behavior. Future work in this scheme will be focused on implementing a method for collecting and exchanging anonymous touch data between users and classifying touches on-device in real time.

5.5 Experiment

5.5.1 Devices Used

The user study was conducted on a Galaxy Tab 3 and Google Nexus 4 smartphone in order to ensure results are consistent across different devices. The Galaxy Tab 3 has an 8 inch screen with a resolution of 1280 by 800, and the Nexus 4 has a 4.7 inch screen with a similar resolution of 1280 by 768.

There are several distinctions between the devices: (1) The accelerometer on the Tab 3 reports significantly slower, so most touches on the Tab 3 do not have an acceleration value, except for slow typists. For faster typists, as little as 5% of touches report all the acceleration values described in Section 5.4. (2) The Tab 3 reports the x and y location of touches to a precision of five decimal places, while the Nexus 4 uses only whole numbers. It is not clear to what digit the Tab 3's values are significant. (3) The Tab 3 reports size on a scale from 0 to 1, while the Nexus 4 reports sizes on a range from 9 to 11. In conclusion, any implementation of the proposed authentication scheme will need to be device specific, or make adjustments between device values appropriately. This may confound future plans to collect information from a pool of users and distribute it between them because users may also have to be on the same device type in order to participate.

5.5.2 Experiment Setup

Participants were recruited via word of mouth and required roughly 15 minutes to complete the experiment. A total of 30 participants were recruited, 15 to type the phrase “mary had a little lamb” on the tablet device, and an additional 15 to type the phrase “maryhadalittlelamb” on the smartphone device. The space character is omitted for the second set of participants to examine if the using space has a detrimental impact on identification accuracy. Touches consecutive to space may have a consistently low time since last touch because the space key is easy to find and reach. This may render the time since last touch data point less useful in classification.

Participants typed the phrase a minimum of 20 times, though some participants chose to type the phrase up to 25 times. To ensure consistent acceleration data, participants were asked to sit in a stationary chair while typing, holding the device in landscape mode with their non-dominant hand and typing with their dominant hand. The particular grip participants used and the manner in which they typed were not specified, but participants could not use the hand holding the device to type or rest the device against any stationary surface. Participants were allowed to choose their stance and adjust it as they typed, for example they could lean forward or back in the chair.

5.5.3 Typographical Correction

Some typological correction was employed similar to Draffin’s approach [123]. Typographical mistakes were treated as follows: 1) if more than three typographical errors were present, the trace was discarded, 2) if a character was typed incorrectly, e.g. “msry had a little lamb,” the incorrect letter was treated as correct (treating “s” as an “a” in the example), 3) if a character was missing, e.g. “mry had a little lamb,” the previous character would be duplicated, e.g. “mmry had a little lamb,” and the typo ignored as in (2) and 4) if an extra character was typed, e.g. “masry had a little lamb,” the extra character was simply removed.

Typographical mistakes, made consistently, may actually help to identify some users. For example, if the user attempts to press “a” and consistently hits “s,” it’s expected that the x-coordinate of the mistaken touch on the “s” key, after correction to an “a,” will be more leftwards than users who touched “a” successfully. Analyzing the feasibility of identifying these mistakes in real time and the impact of keeping these mistakes on accuracy is left to future work. Users made an average of 10 typos in their 20 traces combined. Due to typo correction, some participants ended up with fewer than 20 traces, to a minimum of 15 traces per participant.

5.5.4 Classification and Analysis

Data was processed using computationally efficient classifiers- K nearest neighbors (KNN), binary decision tree, and naive Bayes. The goal of using simple classifiers is to eventually move the classification on-device, so the mobile device can build and use the classifier without connecting to some trusted server (for classification) in a future work.

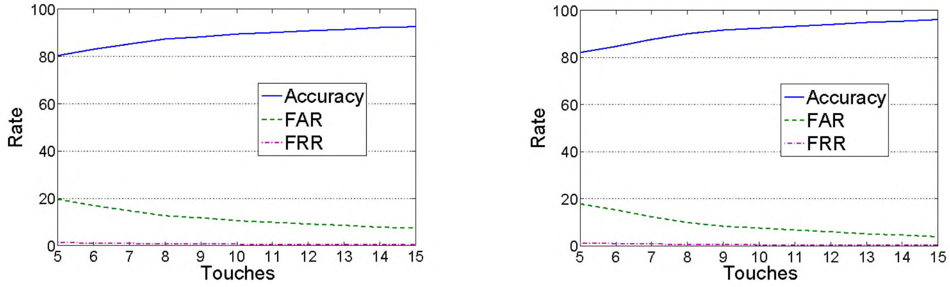
In this work only the binary decision tree results are presented, as they were the most favorable.

Classification success is measured with Accuracy, False Acceptance Rate (FAR), and False Rejection Rate (FRR). Accuracy is simply the percentage of authentication attempts from a user that were correctly matched to that user. FAR is the percentage of authentication attempts that do not belong to a user but were matched to them incorrectly. FRR is the percentage of authentication attempts rejected outright or incorrectly matched from the user to someone else. An authentication attempt will be defined for each approach later on.

5.5.5 Character Independent Classification

In this approach, touches are grouped together and classified without considering the character being touched. The touches are split evenly into testing and training sets at random, with approximately 200 touches per user in each set. All character information is stripped, that is an “a” is treated the same as a “b” or any other character. Training sets from all users are combined and fed to the classifier. Note that for participants typing on the Tab 3, this also means that the “space” character is treated the same as any other character, even though the space bar is significantly larger than other buttons and thus has a wider range of position values. The following hypothesis is formed for the two devices.

H1: Including space will reduce accuracy in the Tab 3’s results, because space has more potential positions and is more rhythmic in touch patterns.



(a) Galaxy Tab 3 (b) Nexus 4

Figure 22: Touches vs Accuracy and FAR/FRR for Character Independent Data

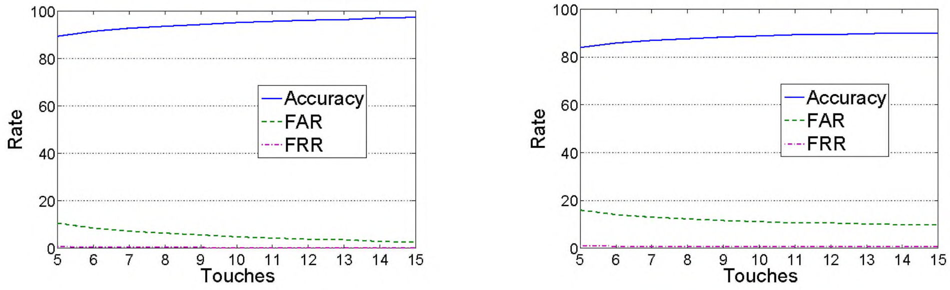
An authentication attempt is begun by taking n touches for each user from the testing set at random and applying the classifier individually to each touch. The identity of the user is determined by taking the plurality of the n chosen touches. For example, n is equal to five touches and user a’s trace contains two touches identified as user a, one as user b, one as user c, and one as user d. The authentication attempt is marked as successful for user a, even though the majority of the touches were attributed to other users, because the plurality of touches were identified as user a’s. An authentication attempt using *n* touches is taken from users b, c, and d in the same manner. A tie does not authenticate any user and is automatically considered a false reject. To ensure consistency, 2000 samples are taken from each user for each value of n, and the overall accuracy, FAR, and FRR are calculated by averaging the results for all users. Figure 22 demonstrates results from $n = 5$ to $n = 15$.

Figure 22 shows an acceptance rate of 93% after 15 touches with an FAR of 7% and an FRR of .5% for the Tab 3 and an acceptance rate of 96% with an FAR of 4% and an FRR of .25% for the Nexus 4. Thus a user can be authenticated after typing a short text message, using a training set that can easily be built in as little as two or three text messages. The results are consistent with the expectation in H1, the Galaxy Tab had slightly lower accuracy than the Nexus 4. Future work will examine the possibility of removing spaces, and potentially any characters that precede or follow space when applying this approach to text that does contain spaces.

This approach could be used to dynamically monitor all typing on the device. Assuming a generous allowance of three incorrect authentication attempts before device lockout, a legitimate user will have a near zero chance of lockout ($7\%^3 = .0343\%, 4\%^3 = .0064\%$), while an attacker will face a substantial chance of lockout after only 45 touches. Because all characters are treated identically, classification data from other users utilizing the authentication application can easily be anonymously collected and distributed, allowing each user of the application to be compared against other anonymous users, potentially against different users for each authentication attempt. This further reduces the chance of a legitimate user being mismatched with another user with similar typing behavior but serves no advantage to an attacker. Future work that authenticates users in real-time will investigate the possibility of swapping which other users are used for classification every few touches.

5.5.6 Character Dependent Classification

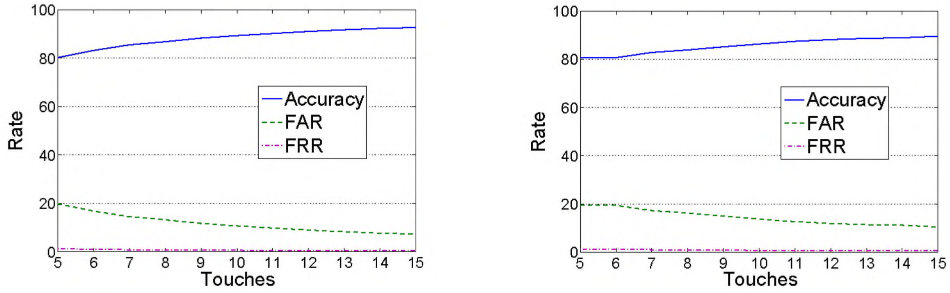
For this experiment, each character receives its own separate classification. The characters “a,” “space,” and “l” are classified as these are the most common characters in the typed phrase. Touches are split evenly into testing and training sets at random, with approximately 40 touches for “a” and “space” and 30 touches for “l” in each set. An authentication attempt for this experiment is defined the same way as in the previous experiment, with n touches taken at random from each user and a plurality-wins model for each authentication attempt. Once again 2000 samples are taken from each user for each value of n.



(a) Galaxy Tab 3

(b) Nexus 4

Figure 23: Touches vs Accuracy and FAR/FRR for the Character “a”



(a) Galaxy Tab 3

(b) Nexus 4

Figure 24: Touches vs Accuracy and FAR/FRR for the Character “l”

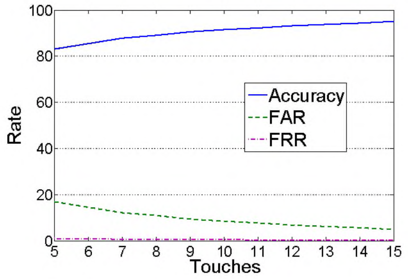


Figure 25: Touches vs Accuracy and FAR/FRR for the Character “Space”

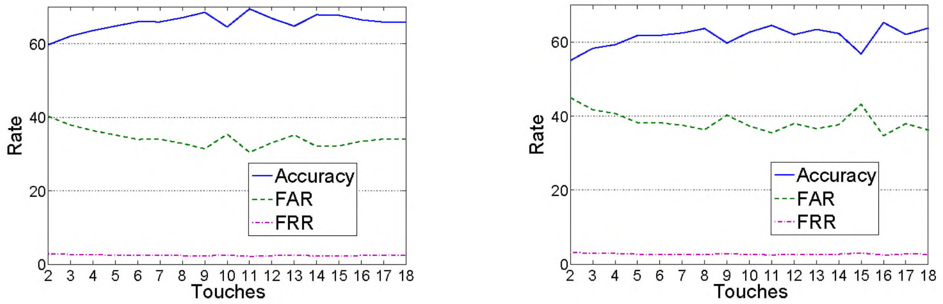
Figures 23, 24, and 25 show results for each character from $n = 5$ to $n = 15$. The approach achieves an accuracy of 97% after 15 touches of the letter “a,” with an FAR of 2.6% and an FRR of .2% for the Tab 3 and an accuracy of 90%, with an FAR of 10% and an FRR of .7% for the Nexus 4. For the letter “l,” the approach achieves an accuracy of 92% after 15 touches with an FAR of 7% and an FRR of .4% for the Tab 3 and an accuracy of 90% with an FAR of 11% and an FRR of .6% for the Nexus 4. Note that for both characters, the Nexus 4 results are confounded by a single user who was consistently misidentified as one other user. Excluding this user puts the accuracy of the Nexus 4 above that of the Tab 3. Comparing against different users for each authentication attempt can reduce the probability of two users with very similar touch behavior getting confused with each other in this manner. Results for the “space” character on the Tab 3 are in line with other characters, achieving an accuracy of 95% after 15 touches, with an FAR of 5% and an FRR of .3%.

This approach can be applied to frequent characters like vowels and space for reliable authentication with reduced overhead. As with the previous approach, the content and order of an individual’s typed text do not matter, so anonymous classification data can easily be collected for different users. Comparing to different anonymous users for each authentication attempt can reduce the chances of a consistent misidentification such as the one encountered with the Nexus 4. While the success metrics for this approach are similar to the previous approach, applying classification only to popular characters can reduce the processing and memory overhead of the scheme. A comparison of the overhead between this approach and the former approach is reserved for the real-time application in future work.

5.5.7 Order Dependent

In the final approach, consideration is given to how a user’s typing behavior may change between different characters. In other words, a user may type the character “a” in a different way if “m” precedes it rather than “h,” and this logic can further be extended to groups of 3, 4, or more characters. This approach can be used for additional security on static text such as passwords. Order of all touches is maintained and touches are grouped into pairs, threes, fours, and so forth, where n is the size of the group. The number of available traces depends on the size of n, for example there are 13 possible ordered letter pairs in the 22 character long phrase, and each user has approximately 20 traces, for a total of $13 * 20 = 260$ traces per user. Although success metrics may be different for certain letter combinations, e.g., for “ma” the results may be worse than for “ar,” results are combined and averaged for the purpose of condensing the results. Since there are significantly fewer traces in this approach, 60% of the traces are selected for training and 40% for testing.

An authentication attempt is considered a set of n correctly ordered characters. Data is merged from consecutive characters into a single row of data for purposes of classification. The entire row, containing data for each character in the sequence, is classified individually, so the authentication attempt is based on a single decision in this approach. This would be the only practical approach for a scheme designed to supplement password entry, since the user will generally enter their password only once per session.



(a) Galaxy Tab 3

(b) Nexus 4

Figure 26: Touches vs Accuracy and FAR/FRR for Multiple Consecutive Touches

From Figure 26, it is clear that the increase in acceptance rate is quite minimal by using more consecutive touches. On a password of five of more characters, an accuracy of approximately 65% is possible. Collecting classification data for an approach such as this one may be problematic, since other users will need to type precisely the same text. This approach may be applied to the entry of phone numbers rather than passwords, since phone numbers are also static. The number would not need to identical as the authentication can be done in three parts based on the area code, first set of digits, and final set of digits. Collecting classification data for phone numbers would be easier as many users will enter, at least, the same area code. This approach will likely not be tested further in future work as the data collection would require a very large pool of users and an implementation that was able to hook into password or telephone number entry, a serious trust issue for mobile users.

5.5.8 Future Approaches

Some other approaches are considered for future work, especially in a real-time application that uses dynamic text generated by users in the wild.

(1) Common phrase only - Analyze only messages that are commonly typed, based on data collected from users over time or publicly available sources for common phrases, for example: "Coming home soon" or "See you later." May substantially reduce overhead, but also may decrease the odds an attacker will be identified.

(2) Fast-typed only - Analyze the user's typing behavior and only classify when typing is fast enough to indicate the user is focused. May reduce the odds of misclassification when the user pauses to think about a message, but also may decrease the odds an attacker will be identified.

(3) Language-independent - The first and second approaches may be able to use data from languages that share the same alphabet without significant drop in accuracy. This could reduce the difficulty of generating a large enough pool of participants to exchange touch data.

(4) Complete word only - Analyze whole words, or compare characters individually but group them by word. Data collection would be more difficult in this approach, as would

retaining anonymity, but typing behavior may be more distinct when typing specific words.

5.6 Discussion

The scheme faces two significant issues before it can be applied to commercial use.

First, smartphones are rarely stationary. The experiment placed all users in a similar stance and asked them to remain mostly stationary. In practice, different stances, for example walking or sitting, must be identified, each with their own corresponding classification. Typing behavior will be different when sitting down compared to standing up, and actions like walking will influence acceleration data. Moving objects, such as cars, will further affect acceleration data, so the scheme should detect when it is inside a moving object and switch to a classification routine that is not based on acceleration data. Future work will focus on blending stance detection and acceleration detection with the scheme.

Second, user behavior changes over time in response to mood, injury, time of day, and many other factors. The experiment collected test and training data in the same session. Future work will collect samples at different times to verify that the scheme maintains accuracy despite day to day behavioral changes.

Data from gyroscopes may be included in future versions of the scheme as gyroscopes become more common on low to mid-end mobile devices. Pressure may also be included if pressure sensitive screens or pens become more prolific.

The scheme should be tested on user generated text input rather than preassigned text. Users may make erratic pauses to think about the text they are typing during normal behavior, and this may lead to some degradation in performance. The degradation may be counterbalanced by differences in text compared to the lab experiment. In the lab experiment, it is possible that users were more likely to behave in similar ways because the text they were typing was identical. Different texts may lead to more different typing behavior, and thus better identification rates. Typo detection will be a significantly greater challenge on user generated text and will also be studied in future work.

In the first two approaches, since each touch is treated independently, the time since last touch feature may be considered as noise. Recreating the classification experiments without the time since last variable touch worsened the results by about 3% across the board. Another future task is to evaluate why time since touch last assists with classification. It's possible that time since last touch indirectly infers typing speed, since faster typists will generally have lower values for this data point.

Lastly, the data used in the experiments required about 1.6MB and 375kB for the first and second approaches respectively. Simple fast zip compression can reduce the file size to 475kB and 90kB respectively, so the scheme can have negligible impact on network use and device storage. Future work will examine the overhead of the scheme on network, memory, and processing resources.

CHAPTER VI

ANALOG AUTHENTICATION

6.1 Outline

This chapter is based on a work awaiting publication at the July 2018 International Conference on Human Aspects of Information Security, Privacy, and Trust (HCII 2018) [126].

Section 6.2 introduces the novel concept of analog authentication and discusses the motivation for developing it compared to the MAPS approach used in previous chapters. Related work on authentication using continuous information is presented in Section 6.3. The implementation of PassHue, a novel proof-of-concept analog authentication scheme, is presented in Section 6.5. The security strength of analog authentication, using PassHue as an example, is addressed in Section 6.6. A user study examining the memorability, usability, hotspots, and shoulder-surfing resistance of PassHue, conducted in-the-wild, is presented in Section 6.7, demonstrating that PassHue excels in all categories. Color-blindness, gender biases, and future work are discussed in Section 6.8

6.2 Introduction to Analog Authentication

In the previous chapters, the security of a password scheme was enhanced primarily by using the concept of multiple dimensions, under the premise that adding an additional dimension is better than making a single dimension larger. While the general principle of this idea works, as demonstrated in the previous chapters, this chapter presents an alternative method for enhancing security strength without necessitating any significant tradeoff in memorability or usability.

This chapter presents the concept of *Analog Authentication*, the idea of using continuous rather than discrete information for authentication. In essence, by using continuous information instead of discrete information, analog authentication is able to make an incredibly large single dimension, facilitating high security strength. In a multidimensional approach, the user has to remember information from different dimensions, which can carry a memorability burden. In an analog approach, a user is tasked to remember a single piece of continuous information– this is largely the same memory burden a user faces in traditional authentication. Likewise, an analog approach can be completed in largely the same way as a similar traditional authentication scheme, meaning little to no difference in entry time, the most important metric of usability.

To demonstrate the power of analog authentication, this chapter presents *PassHue*, an analog authentication scheme based on color that emulates the functionality of 4-digit PIN. PassHue is evaluated using an "in-the-wild" user study that simulates real-world use over a period of two weeks. PassHue offers greatly increased password space compared to classic PIN, and comes with some inherent resistance against shoulder-surfing. Memorability-wise, PassHue has several psychological advantages that may lead it to be more memorable than traditional PIN, and preliminary results support this possibility. PassHue is very similar to PIN in terms of usability. In total, PassHue appears to be a viable alternative to PIN with a number of concrete advantages. As an example of analog authentication, PassHue demonstrates the unique power of using continuous information for authentication purposes.

6.3 Authentication Using Continuous Information

Most traditional authentication methods ask users to remember information which is discrete, such as letters, numbers, or an ordered pattern like Pattern Unlock. Users remember a sequence of discrete information and *recall* that information back exactly. Some methods, such as RealUsers's PassFaces [36], ask users to remember discrete items such as faces or patterns and *recognize* them from a larger set of items later.

By discrete, in this chapter, we mean that the information being remembered can be divided easily into a whole number of choices: there are 26 letters in the alphabet, 10 digits, and 3-8 possible directions that a user can pick from any given dot in a pattern. In practice, many sets which are treated as continuous may also be considered discrete, for example 3D space is sometimes argued to be discrete in terms of the plank length. The discussion of which sets are discrete vs continuous is outside the scope of this work; any sufficiently large set which appears to be continuously variable to an average human will be considered continuous.

Analog authentication asks users to remember information from a continuum. That is, given a continuum such as loudness, an analog authentication scheme would ask the user to reproduce a specific volume or volumes. The memory task is effectively extended from *recall* to *estimation*, as the user must now not only remember the volume that was previously set but also estimate it accurately. By necessity, a tolerance must be given to the user for error. The password space of an analog authentication scheme is proportional to the size of the continuum divided by the tolerance. Thus, analog authentication has a direct tradeoff between usability (a larger tolerance so the user can authenticate more easily), and security (a smaller tolerance to increase the size of the password space).

The foundation for any analog scheme is simple: for a continuous data set, allow the user to chose some number of items as a password, then compare future authentication attempts against those items for similarity. If the similarity score is sufficient, the user is authenticated.

We will use the term analog authentication to convey that a continuum of information is being used rather than discrete information. The concept is not to be confused with continuous authentication, which refers to authentication that functions by analyzing user behavior in the background while the user is interacting with a device, for example the scheme discussed in the previous chapter.

Intuitively, it is tempting to assume that humans will perform at vastly varying abilities depending on the estimation task. The well-known “seven, plus or minus two” rule [21] dictates that the average person can distinguish between about 7 pieces of continuous information at a time. When a continuous set is broken into n items, humans start having trouble discerning between items when n is larger than 7. The general rule holds for continuous sets such as pitch, saltiness, loudness, or points in a square. Humans are generally able to discern between no more than 7 unique items before accuracy beings to suffer considerably. Splitting the continuum any more finely leads to errors with rapidly increasing frequency.

Cowan [22] describes the limit as “The Magical Mystery Four” instead, arguing that working memory for the average young adult is limited more closely to 3-5 items.

It follows that an important concern in the design of an analog authentication scheme is to ensure that the user does not have to break the continuum down into more than 7 pieces, and fewer is better. Beyond that, the memorability of a particular continuum must be justified individually; there is no research suggesting that continuous data is always more memorable than discrete data or vice-versa. As Miller [21] demonstrates, even though memorability is similar between various types of continuous information, some are still more memorable than others.

Discrete information like letters is often bundled into higher-order information like words or sentences to improve the number of items a person can remember. Similarly, continuous information like pitch and tone can be bundled into higher order information like notes and songs, though this may also have the effect of making the information discrete.

6.4 Related Work: Analog Authentication

Currently, continuous information is seldom used for authentication. Even when potentially continuous information is used, it is often presented in a discrete manner. For example, if the user is asked to pick a color, for example as a banking security question, they are typically presented with a short list of options or asked to use a standard language-based description such as “blue” or “silver”.

An exception is free-form gesture drawing, such as the work by Sherman *et al.* [127] and by Clark and Lindqvist [128]. Free-from gesture drawing uses a continuous 2D drawing to authenticate the user, placing it firmly in the realm of analog authentication. While these works have discussed the implications and advantages of utilizing continuous information as opposed to discrete information, none have formalized the concept of using continuous information outside the scope of free-form gesture drawing. Free-form gesture drawing can be considered just one type of analog authentication. Additionally, free-form drawing can be considered as an example of analog authentication which bundles low-level continuous information (2D positions), into higher order information (lines and shapes), while still preserving the analog nature of the method.

On the contrary, Google’s Pattern Unlock, and in fact any touch-based authentication on the mobile platform, can be considered examples of turning analog information (2D positions) into discrete information (connections between points, digits, etc). Buttons that users touch to input a PIN or password can be considered a type of tolerance: any 2D positions that fall inside the button count as the same input. We will not count these methods as analog because the memory task facing the user is discrete, only the input method is continuous. In other words, a scheme can only be considered analog authentication if the input method, password, and memory task are all based on continuous information.

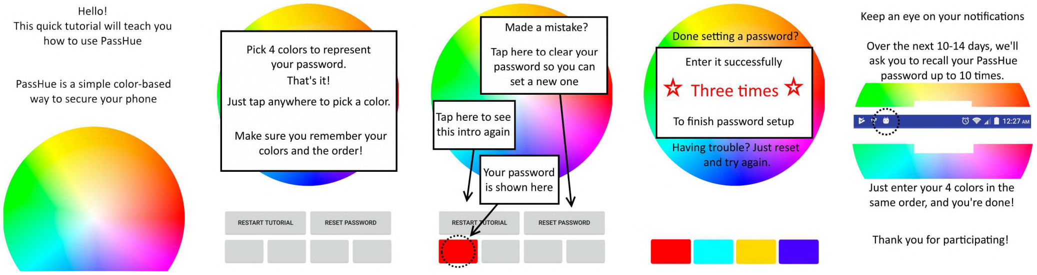
Bianchi's [54] works falls into a similar category: continuous information such as vibration, beat, and hold time is ultimately used as a cue for discrete information like numbers. Remembering the cue is part of the memory task, so Bianchi's approaches can be considered partly analog, however analog cues such as vibration are broken down into discrete functions like "number of vibrations that have elapsed", an integer value which is plainly discrete.

In biometrics, analog authentication is the norm, utilizing continuous data such as gait and typing speed. Biometric methods are outside the scope of this chapter, we will be specifically focused on knowledge-based methods.

This chapter introduces the concept of analog authentication, the idea of using continuous data for authentication. As an example of analog authentication, PassHue, a mobile authentication scheme that uses a color continuum, demonstrates the potential advantages of analog authentication. PassHue follows a PIN-like approach similar to the classic numerical PIN, SwiPIN [47], or ColorPIN [48]. It is designed to be immediately familiar to end users and offer login times and memorability on par with existing PIN-based approaches. PassHue improves on existing mechanisms by providing a much larger password space and moderate protection from shoulder-surfing. As an example of analog authentication, PassHue demonstrates that continuous information can be used for authentication just as well as discrete information. An in-the-wild user study demonstrates that PassHue can achieve high usability and remain memorable over a period of 2 weeks.

6.5 The Design of PassHue

This section addresses the design of PassHue, an example of analog authentication that utilizes color.



Hello!
This quick tutorial will teach you how to use PassHue

PassHue is a simple color-based way to secure your phone

Pick 4 colors to represent your password.
That's it!
Just tap anywhere to pick a color.

Make sure you remember your colors and the order!

Made a mistake?

Tap here to clear your password so you can set a new one

Tap here to see this intro again

Your password is shown here

Done setting a password?

Enter it successfully
☆ Three times ☆
To finish password setup

Having trouble? Just reset and try again.

Keep an eye on your notifications

Over the next 10-14 days, we'll ask you to recall your PassHue password up to 10 times.

Just enter your 4 colors in the same order, and you're done!

Thank you for participating!

Figure 27: Tutorial Images Shown on the Store Page

PassHue is implemented on Android. Figure 27 shows the tutorial images that are presented on the Play Store listing for PassHue. Users are given no further guidance beyond these images.

PassHue is designed to simultaneously use three continuous sets of color: red, green, and blue, referred to as "RGB" values. The RGB system is the most common method for representing colors in digital applications– a color is made up of one value from each set: R, G, and B. In general, each set has a range of 0-255, so the sets are not actually continuous, but sufficiently large so that they appear continuous to a human. The size of RGB color space is quite large; 256^3 yields approximately 16.8 million possible colors, and this is often referred to as 24-bit color. Most modern mobile displays support 24 bit color. The iPhone for example, has supported 24-bit capable hardware displays since the iPhone 4. Sometimes, an additional 8 bits are assigned to transparency, often called 32-bit color. PassHue does not utilize transparency.

Because PassHue utilizes three continuums, it is possible to consider PassHue to be multidimensional. In fairness, many people would consider color to be a single dimension, and indeed in examples from previous chapters such as CMAPS, color was considered as one dimension, not the product of three dimensions of RGB color. The determination of PassHue as a multidimensional scheme is largely subjective, depending on a person's perception of color as one dimension, or a product of three R, G, and B dimensions. What is certain however, is that the R, G, and B dimensions are still counted multiplicatively towards password space, and it is the product of these three dimensions that gives us the size of RGB color space.

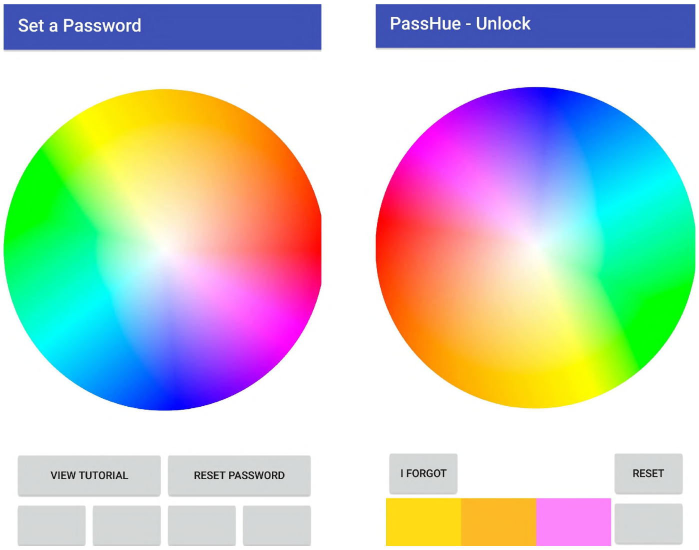


Figure 28: The Password Setup Screen (left), The Login Screen (right)

Users set a password by tapping 4 colors in order. The 4 RGB color values are stored as the user's password. The password setup screen is shown on the left in Figure 28. Tapping the "View Tutorial button" allows users to see the images in Figure 27 again. Before finishing password setup, the user must re-enter the same 4 colors an additional 3 times to verify that they remember the password. Until the 3 verification entries are complete, the password is not set. If the user decides that their password is too hard before verifying it 3 times, or wants to pick a different password for any other reason, they can reset it with no penalty using the "Reset Password" button.

To authenticate themselves later, users must pick the same 4 colors, within the tolerance, and in the same order. The login screen is shown on the right in Figure 28. The user has already picked three colors: orange, yellow, and pink – those choices are tracked at the bottom of the screen. The fourth color is still awaiting user input. The "Reset" button can be used to clear the current input if a mistake was made, and the "I Forgot" button clears the user's password and allows them to set a new one if they wish to continue the experiment. This button is included so users can easily communicate that they do not remember their password.

Colors are picked by tapping the standard color wheel shown in Figure 28. The wheel is identical to the color wheel found in many graphics applications such as Adobe Photoshop and Paint.net. There are three elements in RGB color, and it is difficult to express all three in a single 2D image while maintaining a continuously variable pattern. Ideally there should not be "jumps" in color when the user moves across the image, otherwise two very different colors may end up adjacent, and this can make picking a color accurately difficult. Additionally, the user should be able to locate colors quickly based on location, for example it is expected that orange falls between yellow and red. The color wheel accomplishes these requirements; movement in any direction around the wheel is associated with a gradual change in color, movement towards the center increases the "whiteness" of the color, and colors appear in classic order around the wheel: red, orange, yellow, green, blue (cyan), indigo (dark blue), violet, and purple.

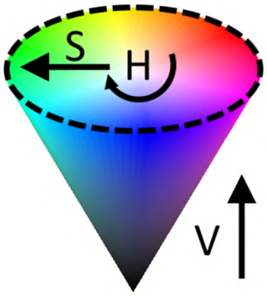


Figure 29: Cone Representation of HSV Color Space. (Hue is the primary color where red is 0 degrees, Saturation is the strength or intensity of the color, and Value describes how dark the color is.)

A tradeoff to using a color wheel is that relatively few RGB colors are represented. The color wheel used by PassHue is often called an HSV (Hue, Saturation, Value) wheel, which typically features a Value slider in addition to the wheel. HSV is a simple transformation of RGB. Figure 29 shows how an HSV system addresses colors; the flat area at the top enclosed by the dotted line represents the portion of colors used by PassHue and demonstrates that PassHue uses only Hue and Saturation in the HSV scheme.

Because PassHue uses only the top of the HSV cone, PassHue contains only the

RGB colors where at least 1 of the 3 RGB values equals 255. This allows PassHue to display a variety of colors with a consistent gradient on a 2D display but sacrifices the ability to display the colors that reside in the rest of the HSV cone. On the outside of the circle, the "pure" RGB values where 1 or 2 colors are equal to their maximum (255) and 1 or 2 colors are equal to their minimum appear. For example, pure red is (255, 0, 0) and pure yellow is (255, 255, 0). On the inside of the circle, pure white appears (255, 255, 255). Values where all three colors are less than 255 are not present, for example (5, 15, 135).

PassHue is designed around color because color is a continuum that humans are relatively good at processing. There are at least 2.8 million colors discernible to normal humans [129], and some researchers suggest as many as 10 million [130]. Halsey and Chapanis [131] presented participants with 342 CIE colors, all of equal luminance, and asked them to match given colors exactly to one of the 342 presented colors. Participants could pick out over 11 unique colors– that is, colors with no overlapping matches to other colors, at the 5% error level– and over 15 colors at the 10% level, significantly better than the expected "magic number seven" [21]. Hamwi and Landis [132] found no relationship between time delay and color memory for delays of 15 minutes, 24 hours, and 6 hours, indicating that color may be good for long-term memory.

6.5.1 Comparison of Color Values

PassHue illustrates a potential difficulty in analog authentication: humans are often better at discerning values on one part of the continuum than on another. This can make it difficult to establish an exact estimate for the tolerance t, since t varies depending on which part of the continuum the user picks. In color, for example, humans are worst at discerning shades of green [131, 133], so the tolerance should be greater for green colors. Euclidean distance between colors does not accommodate for different levels of performance with different colors.

PassHue compares colors using the CompuPhase algorithm [134], a commerciallyused, simple, and efficient method for calculating the distance between two colors in a way that tries to emulate how a human would perceive the distances. A key advantage of the CompuPhase algorithm over an algorithm like CIE2000 is that it has significantly fewer mathematical operations and does not require conversion into another color space, potentially saving valuable overhead. Processing overhead is especially important on mobile devices with limited computing resources and battery life. The algorithm describes the difference between two RGB colors using the following equation.

$$\sqrt{(2+\frac{\bar{r}}{256})\times\Delta R^2+4\times\Delta G^2+(2+\frac{255-\bar{r}}{256})\times\Delta B^2} \tag{6.1}$$

where $\bar{r}$ is the mean red level, i.e., $(R_1+R_2)/2$, and ΔR, ΔG, and ΔB are the differences between the respective R, G, and B values of the two colors. The result of Equation 6.1 will be referred to from here on as the similarity score– the lower the score, the more similar the colors.

A similarity score of 100 or lower is considered a match. All 4 colors in a user's password must match for authentication to succeed. That is, the similarity score for all 4 chosen colors vs the stored RGB values for that password must be 100 or less.

The greater the tolerance, the more colors are treated as identical, resulting in a smaller password space. A score of 100 was chosen after a brief pilot test with 5 participants. A goal of the user study in this paper is to determine if the score should be raised or lowered for the average person. For most users, a lower score may be sufficient, while a few may struggle without a higher score. In a full commercial application of PassHue, the similarity score may start high and gradually reduce if the user continuously meets a lower score, allowing users with better color-discerning ability to enjoy increased security.

6.6 Security Strength of PassHue

Current mobile authentication methods are limited in password space. A 4-digit PIN can generate only $10^4 = 10,000$ passwords, and pattern unlock offers $389,112$ possible passwords on a 3 by 3 grid [50]. This section addresses the password space of PassHue. Hotspots and other considerations that may lower effective password space will be discussed in Section 6.7.

The password space of a traditional discrete scheme is C^n, where C is the number of choices per item and n is the number of items chosen. The password space of an analog authentication scheme with one variable is $((C/t)^n)$, where C is the size of the continuum, t is the tolerance, and n is the number of choices picked from the continuum. We can consider t to be number of items in C which are treated as identical for purposes of satisfying the password. In a traditional discrete scheme, we can say that t is equal to 1.

PassHue uses 3 continuums: red, green, and blue, but only a single tolerance based on a score generated using all 3 values. For simplicity we can combine the colors and consider C to be a single continuum from (0, 0, 0) to (255, 255, 255). The password space of PassHue is therefore $(C/t)^n$, where C is the size of the continuum and n is the number of choices picked from the continuum.

In practice, PassHue uses only the top circle of the HSV color cone, so at least one RGB value must always equal 255. We can choose any of the three colors R, G, or B to set equal to 255. While one color must be 255, there are 256 options for both other colors, leaving $(256 * 256)$ choices. The total size of C is therefore $3 * (256 * 256) = 196,608$ colors, which represents just over 1% of RGB color space. Users pick 4 colors, so n is equal to 4.

It is difficult to calculate a value for the tolerance. The similarity score, generated according to Equation 6.1, weighs the values in each continuum differently, so the tolerance varies depending on the values of the colors in question.

To find an accurate estimate for t, a short script is generated to process all RGB color pairs where at least one value in both colors is equal to 255, and the similarity score between the colors is between 99-100, yielding approximately 40 million pairs. The worst-case product of differences between two colors having a similarity score of 99-100 (i.e., $\Delta R \times \Delta G \times \Delta B$) is approximately 39,000, and the average product is approximately 3,400. That is, for any given RGB value, on average, there are 3,400 other RGB values that would be considered a match for purposes of authentication.

This can be considered a lower bound for purposes of determining how many colors will be treated as identical, since the value found by the algorithm is always the worst distance for that color, but we assume that this same distance will apply in every direction in RGB space.

Using the average estimate for t, PassHue has a password space of $(196,608/3,400)^4 = 1.1 * 10^7$ (23 bits), 1000 times larger than a traditional 4 digit PIN, and 28 times larger than pattern unlock on a 3 by 3 grid.

6.7 PassHue User Study

Participation in this experiment is anonymous. The user study is designed to determine the effectiveness of an analog authentication scheme in-the-wild. Users download the application on their own device, set a password, then recall that password several times over a period of 14 days to simulate a phone unlock scheme that is used daily. Participants are given little to no guidance about how to use the scheme, the entire tutorial is contained in Figure 27, and viewing it is optional. To keep the time burden on participants low, notifications to authenticate occur just once per day, though a typical authentication scheme will be used far more frequently by most users.

6.7.1 Data Collection

Information is transmitted via SQL to a dedicated private server. Participants download the application from the Google Play Store.

After downloading the application, participants must verify they are over 18 and consent to the terms and conditions before they can continue. After, participants are asked to provide optional demographic information, such as age, which is encrypted and transmitted to a remote server. Participants are then taken directly to the password setup screen in Figure 28 with no further guidance.

Passwords are stored on the device's local memory and also transmitted to the remote server upon creation. If the application fails to connect to the server for any reason, for example the participant's internet connection has failed, the participant can still use the application normally but any data that would be transmitted is instead lost forever. It was felt that including redundant transmission when the user regained internet access was too intrusive for a a voluntary experiment. Because modern cellular connections are relatively stable, little information should be lost in this manner.

Information about authentication attempts is transmitted to the remote server after each attempt, including total entry time, entry time for each individual color, raw RGB values of each attempted color, and the similarity score between the attempted colors and the actual password.

After initializing a password, participants are notified once per day, at approximately the same time of day the password was originally set, to recall the password. Notifications last a total of 14 days. Participants can chose to ignore a notification, so in practice most participants made between 5-10 recall attempts.

After 14 days, the application notifies the user to complete an exit survey. Users answer basic questions about how they liked the scheme and are given the option to leave written feedback. This information is encrypted and transmitted to the remote server.

6.7.2 Participants

Participants were recruited with fliers, social media posts, and word of mouth. A special thank you is extended to the /r/Android community on Reddit for providing a large number of participants. To be considered as completing the experiment, participants needed to attempt recall on at least 4 different days, or forget their password and have completed at least 2 authentication attempts. A total of 38 participants completed the experiment. The drop out rate, based on participants who set a password but did not meet the above criteria, is 30% (16 participants).

Participation is anonymous, but participants are asked to provide some optional demographic information at the start of the experiment. Some information, such as Android version and country of origin, is collected automatically by Android. Of the 38 participants, 35 chose to provide their age and 37 provided their gender. The average age of participants was 25.5 (median=21, std=11) and the population was 22% female. Most participants ($\geq$ 60%) were using Android 7.0 or higher. Approximately 70% of participants were from the United States, but participants were also found in Canada, Sweden, Russia, Australia, and the UK.

Because the experiment requires ownership of an Android device as well as some rudimentary abilities such as downloading the application from the Play Store, the experiment self-selects towards participants who are already skilled at using their device. Participants self-reported skill with using their device on a scale from 1 (worst) to 5 (best), with an average score of 4.7 (median=5, std=0.74).

Participants were asked what unlock method they currently use to lock their device, with the following options and scores respectively: PIN (1), Dot Pattern (7), Fingerprint (25), Alphanumeric Password (0), Other (2), Don't Lock Device (1), or Prefer not to Answer (2). The population has an unusually high rate of locking their device, but this is expected in a population of people who were interested in an experiment about device authentication. The rate of fingerprint is also quite high, especially compared to CMAPS and PassGame experiment populations, probably because this experiment appealed to mobile phone enthusiasts who tended to have higher end devices which supported fingerprint authentication. As mentioned previously, fingerprint authentication is not currently a replacement for knowledge based authentication, and users of fingerprint authentication are still required to set a separate PIN or password as well.

On the first application startup, participants were randomly assigned into one of the following two conditions for the remainder of the experiment.

(1) Stationary – In this condition, the color wheel appears in the same orientation for every login attempt. The default orientation is shown on the left in Figure 28.

(2) Rotating – In this condition, the color wheel has a different rotation for every authentication session. Figure 28 demonstrates this condition: the color wheel on the right is oriented differently from the color wheel on the left. The wheel's rotation is determined only once, at authentication start; the wheel is *not* rotated again if the user authenticates incorrectly. When initially setting the password, users must confirm the password 3 times before it is set. In the rotating condition, the color wheel is rotated after each successful attempt.

Nineteen participants were assigned to each condition. Participants were not made aware that different conditions existed and received no guidance about rotation or lack thereof. The rotating condition is designed to reduce shoulder-surfing at the cost of some usability. The following hypotheses are generated for the conditions.

H1: Entry times and failed authentication attempt per session will improve over time in both conditions. The rotating condition will perform slightly worse in terms of entry times and failed authentication attempts when used for the same amount of time. Hypothesis H1 is addressed in Section 6.7.4.

H2: The rotating condition will perform better in terms of shoulder-surfing resistance. Hypothesis H2 is addressed in Section 6.7.6.

The rotating condition is inherently resistant to smudge attacks or attacks based on observation of position, since the touch locations are not in the same place for different login sessions.

6.7.3 Memorability of PassHue

Three participants forgot their passwords, for an overall memorability of 92%. Two participants belonged to the rotating condition while the third belonged to the stationary condition. There was no significant difference in memorability between the conditions ($\chi^2 = .36, p = .548$).

All three participants forgot their passwords within the first three days of the experiment. After resetting a new password, all three participants went on to complete the experiment successfully. PassHue appears to be highly memorable, even after a period of two weeks, but a small subset of users can have issues with initially memorizing a password, possibly due to poor password choice.

6.7.4 Usability of PassHue

Entry times were recorded for a total of 1192 authentication attempts. To obtain more realistic timing data, attempts that were likely "pocket dials" or random tapping were filtered out. When the sum of difference scores for the four colors ($C1 + C2 + C3 + C4$) was greater than 500, that attempt was considered a pocket dial or random tapping and discarded. Most of these attempts had entry times lower than 1.5 seconds. There were 112 attempts removed in this manner. Attempts where the total login time was much longer than 60 seconds were also filtered, indicating the user accidentally left the application open, presumably after a partial pocket dial or interrupted session. Nine attempts were filtered in this manner, leaving a total of 1071 valid authentication attempts.

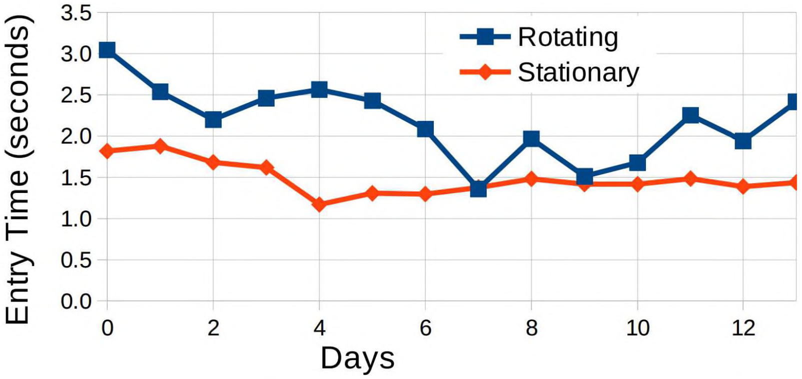


Figure 30: Median Entry Time of PassGame Users Over Time ("Days" represents the number of individual calender days the user has made an attempt at authentication.)

Figure 30 shows the median entry time in seconds for PassHue users over a 14 day period. Figure 30 is based on *days attempted*, though several days may have elapsed for the user. For example, if the user attempted authentication on days 0, 4, 7, and 12, those attempts would be plotted as day 0, 1, 2, and 3. Attempts are organized in this manner because the elapsed days between authentication attempts is not consistent for each user since many users opted to skip days. Participants attempted authentication on an average of 11 separate days. The inconsistent data seen beyond the 12 day mark may be due to the small sample size – only 10 users authenticated on 13 or more days.

The overall average time for a single authentication attempt is 2.63s (median=2.25, std=1.99) for rotating PassHue and 1.67s (median=1.46, std=.86) for stationary PassHue. As expected, two-tailed Mann-Whitney testing indicates a significant difference between the entry times for the two conditions ($p < .0001$).

The data supports H1: entry times improve over a short practice period, and the rotating condition is slower. The average entry time of both conditions is close to in-the-wild entry times reported by other research for traditional 4 digit PINs (1.5s) and Pattern Unlocks (3.1s) [135]. PassHue's in-the-wild entry times are superior to average lab entry times for similar PIN-based schemes: traditional PIN – 1.3s [47, 48], ColorPIN – 13.9s [48], SwiPIN – 3.7s [47], DOC – 25.7s [69], and The Phone Lock – 12.2s [49].

Error Rates

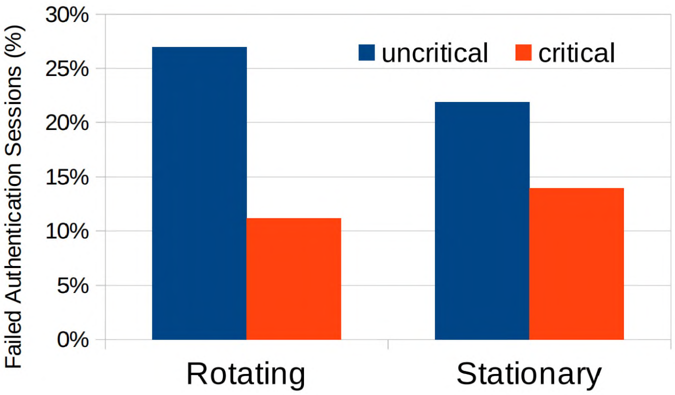


Figure 31: Authentication Sessions With Failures

Figure 31 shows the percentage of authentication sessions that result in critical or uncritical failures. A critical failure is defined as 3 or more incorrect attempts consecutively, as this could traditionally lead to a temporary device lockout. A session is defined as all the authentication attempts in a single instance of using the application. A uncritical failure is 1-2 incorrect attempts in the same session. Users required multiple attempts to authenticate in roughly 35% of authentication sessions in both conditions, but Stationary users face more critical failures.

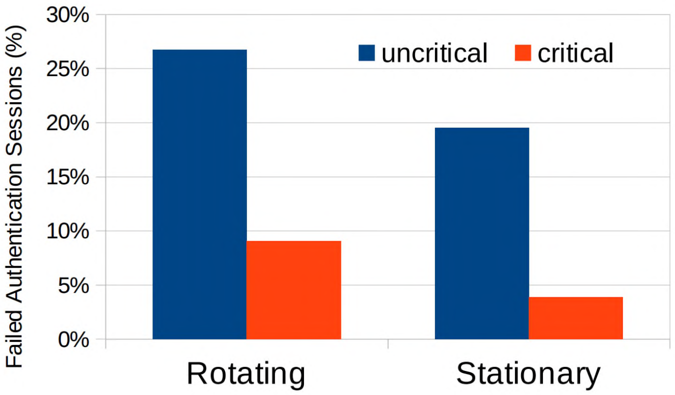


Figure 32: Authentication Sessions With Failures (Outliers Removed)

The average number of incorrect authentication attempts per authentication session is .90 and 1.34 for Rotating and Stationary respectively, violating the expectation of H1.

Most failed attempts originated from a small subset of users, particularly in the Stationary condition. Users who made errors more than two standard deviations above the mean error rate were removed as outliers to generate Figure 32. Even with outliers removed, for most users, the error rate is worse than Pattern Unlock error rates reported in other research (14.6% critical, 1.6% uncritical) [135]. PassHue would require a high error tolerance before lockout to be viable for many users.

With outliers removed, the average number of incorrect authentication attempts per authentication session is .76 and .46 for Rotating and Stationary respectively. The data supports now supports hypothesis H1, average Rotating users will make more errors than their Stationary counterparts.

Because PassHue is very fast, the time impact of incorrect authentication attempts is largely insignificant. Using timing results for one authentication attempt from the previous section, users can expect to spend an average of 1.99s and .77s making errors in Rotating and Stationary PassHue respectively. One explanation for the high error rate of PassHue is that some users simply preferred to go quickly rather than carefully since there was no punishment for multiple incorrect attempts. PassHue offers a greater chance to trade speed for precision than most discrete authentication methods, simply because there is a greater opportunity to miss. In Pattern Unlock for example, it is easily apparent when a mistake is made, and the swipe gestures used require far less precision.

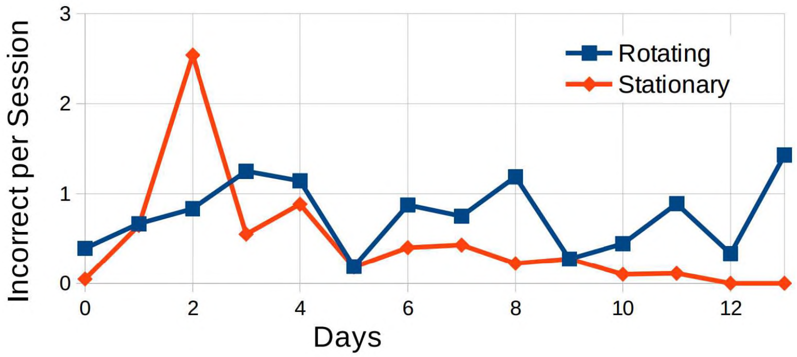


Figure 33: Failed Authentication Attempts per Session Over Time (Outliers Removed)

Figure 33 demonstrates the improvement over time in failed authentication attempts per session. A clear trend emerges in the Stationary condition, demonstrating that PassHue users become significantly less error prone after just 4 days of use, with diminishing gains in accuracy after one week. Rotating users do not share this training effect, violating the expectation from H1, possibly because they do not have a chance to build muscle memory due to the color wheel being in a different position each time.

User Survey

The exit survey asked participants a short series of questions about PassHue, followed by a section for write-in comments.

H3: Stationary PassHue will score higher on user-reported usability metrics than Rotating PassHue.

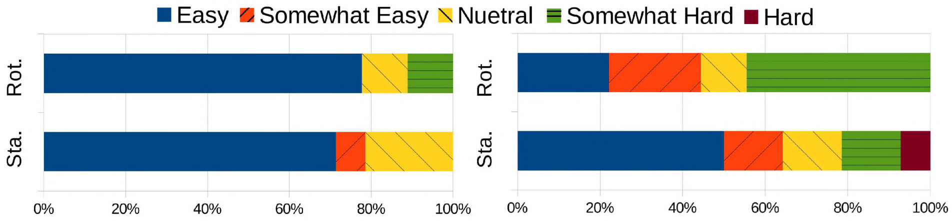


Figure 34: User Survey Responses by Condition (ease of creation (left), ease of login (right))

Figure 34 shows the user survey response rates for the questions: (1) “How easy was it to make a PassHue?” and (2) “How easy was it to login with PassHue?”. The average score on a scale from easy (5) to hard (1) is 4.5 and 4.44 ($stdev = .85, 1.13$) for creation in Stationary/Rotating respectively. The average score for login is 3.86 and 3.22 ($stdev = 1.41, 1.30$) in Stationary/Rotating respectively. For statistical analysis, responses were grouped into the categories satisfied (4 or greater) or unsatisfied (3 or lower). A Chi-Squared test reveals no significance for creation ($\chi^2 = .002$, $p = 0.964$) or login ($\chi^2 = .878$, $p = 0.349$). A majority of users found the PassHue creation and login experience to be easy in both conditions.

To the question “Would you prefer to use PassHue instead of the way you currently lock your phone?”, the response from the Stationary condition was 21% Prefer PassHue, 21% Somewhat Prefer PassHue, 29% Neutral, 7% Somewhat Prefer Current, and 21% Prefer Current; the response from the Rotating condition was 0% Prefer PassHue, 11% Somewhat Prefer PassHue, 11% Neutral, 22% Somewhat Prefer Current, and 56% Prefer Current. Not surprisingly, Stationary users perceive PassHue as a viable alternative for themselves, to use instead existing authentication methods. Rotating users were not quite as willing to adopt PassHue, a further indication that forfeiting even a small amount of usability is largely unacceptable to most users.

Most of this experiment’s population, which is majority fingerprint users, would likely not replace their biometric authentication with PassHue, particularly in the rotating condition. PassHue may still replace existing fallback schemes, like PIN, that accompany biometric methods, and a large percentage of Stationary participants indicated that they would prefer PassHue over what they currently have. Considering that PassHue was compared primarily against fingerprint, one of the fastest authentication methods, survey results can be considered very favorable for PassHue.

6.7.5 Color Selection and Hotspots

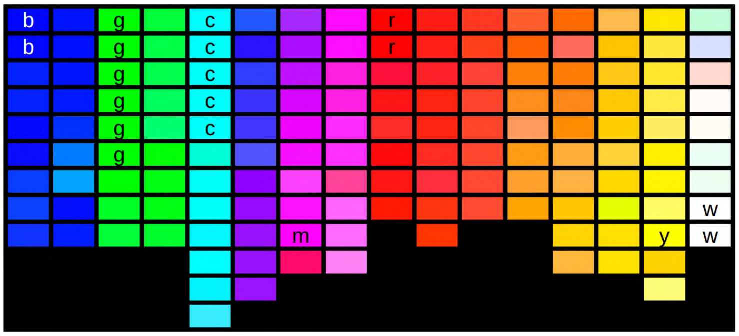


Figure 35: Colors Selected by Participants (Colors, within $\Delta R + \Delta B + \Delta G \leq 10$ of the true value of that color, e.g., 0, 255, 255 for true cyan, are marked for blue, green, cyan, magenta, red, yellow, and white.)

Figure 35 shows all colors selected by our participants, grouped roughly in ascending RGB order. There was no apparent impact on color selection by condition. Cyan, violet, and white hues are slightly under-represented, while hues between yellow and red, are slightly over-represented. The mean RGB values are ($\bar{R} = 161$, $stdev = 116$; $\bar{G} = 133$, $stdev = 99$; $\bar{B} = 131$, $stdev = 112$). The expected mean assuming an even distribution is 128, indicating red is slightly over-represented.

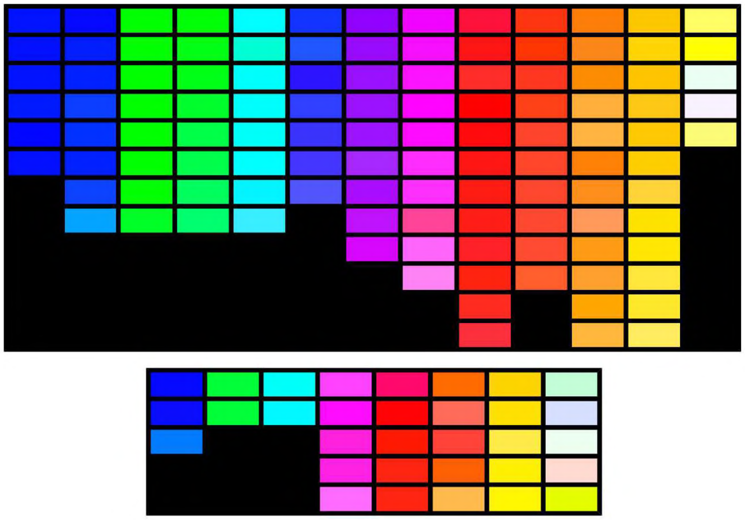


Figure 36: Colors Selected by Male (upper) and Female Participants (lower)

Figure 36 shows color choices separated by gender. Although both genders were inclined to choose colors in the yellow-red hue, females appear less likely to choose blue, green, cyan, and violet hues. Men may also be more likely to choose colors with less saturation, that is colors with lower total RGB values. In particular, the mean R value for males was 40 points lower than the mean R value for females (male: $\bar{R} = 154$, $stdev = 117$, female: $\bar{R} = 195$, $stdev = 145$), however Mann-Whitney testing on the R values found no significant difference($Z = -1.48$, $p = 1.39$).

A brute force attacker may gain some advantage guessing shades of red, orange, and yellow first. From Figure 35, we can note that roughly 40% of all color choices fall between true red (255, 0, 0) and true yellow (255, 255, 0), despite this section making up only one sixth (17%) of the color wheel. In other words, red-orange-yellow is selected roughly twice as frequently as expected. However, only one participant (3%) relied on these colors exclusively.

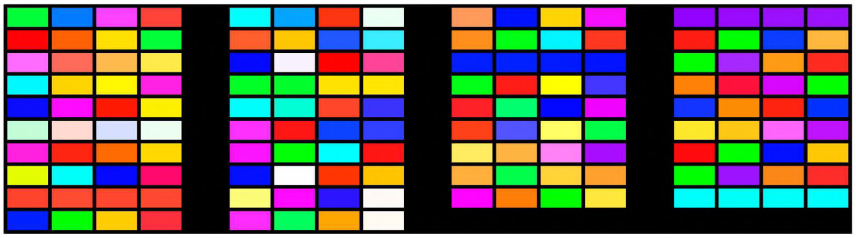


Figure 37: All PassHues Chosen by Participants

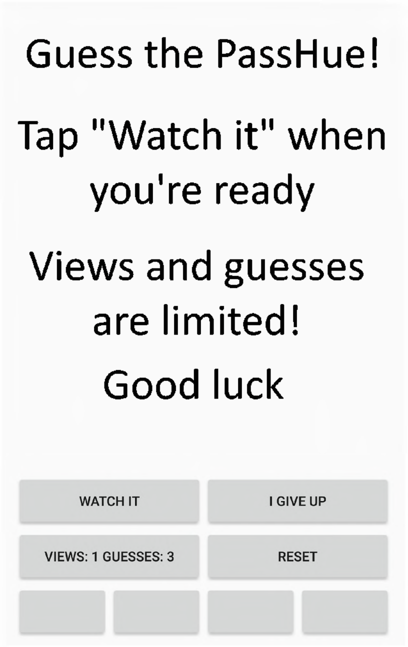
Figure 37 shows all the PassHues chosen by participants in the user study, each row represents one participant. Four participants (11%) generated a PassHue using the same color 4 times. We will define “same color” passwords as a password where all 4 colors had a difference score of less than 10 based on Equation 6.1. The results are very similar to 4-digit PINs, where roughly 8% of PINs are comprised of 4 duplicated digits [136]. Two more participants (5%) generated a PassHue using the same color twice.

Notably absent in the data are repeating patterns such as couplets in the form XYXY, which comprise approximately 18% of PINs. Number based patterns, such as the years 1951-2000 (accounting for roughly 6% of all PINs [136]) are impossible in PassHue. PassHue may have a substantial advantage in encouraging good password choices, simply because there are relatively few commonly occurring patterns based on color. Future work will study the PassHues of more participants in order to determine if any patterns emerge in password selection.

6.7.6 Shoulder-Surfing Resistance

After concluding the memorability and usability portion of the experiment, participants were invited to participate in a shoulder-surfing experiment within the same application. Most participants were notified approximately 1 month after completing the memorability study via an Android notification. The previous experiment had to be completed in its entirety to attempt the shoulder-surfing experiment.

The shoulder-surfing study is designed to determine the difficulty of guessing a PassHue after observing it. Four PassHues were generated using actual colors used by participants in the previous experiment.



Guess the PassHue!

Tap "Watch it" when you're ready

Views and guesses are limited!

Good luck

Figure 38: PassHue Shoulder-Surfing Experiment Start Screen

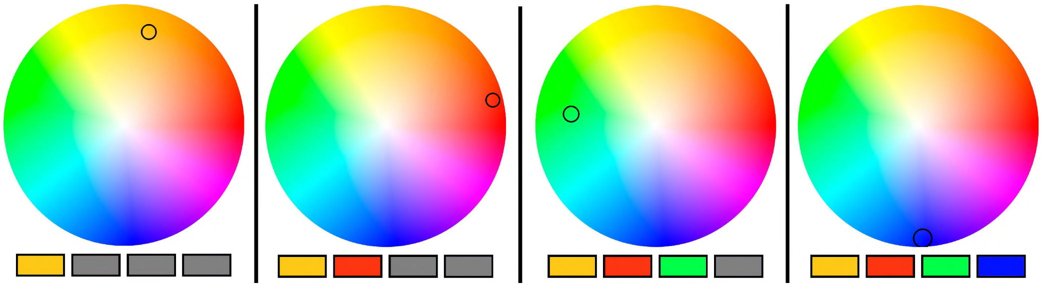


Figure 39: Shoulder-Surfing Images

Participants are directed to the screen shown in the Figure 38 with no further instruction. The “Watch It” button causes the top area to show a sequence of 4 images, an example is shown in Figure 39. Each image is displayed for .7s, so the PassHue is “entered” in a total of 2.8 seconds. The circled area indicates the correct touch location and color. Color selections are also shown on the four squares below the wheel. When the clip ends, the top area shows the color wheel, allowing user input.

The user can keep track of how many more times they can view the clip and how many guesses they have remaining. Users can reset their current input without consuming a guess at no penalty. Users can also give up at any time and move on to the next PassHue. Each PassHue has its own counter for views and guesses.

Participants remain in the same condition as for the previous experiment. Stationary participants see the color wheel in the same orientation for each PassHue, and enter the PassHue in the same orientation they saw it in. Rotating users see each PassHue in a different orientation, and the color wheel used for guessing is rotated after using up a view.

Users are issued 1 view and 3 guesses to begin. Upon giving up, or exhausting guesses, users are issued 2 additional views and 3 additional guesses, for a total of 3 views and 6 guesses. Once these are exhausted, or the PassHue is guessed correctly, users move on to the next PassHue. If a user guesses the PassHue without additional views and guesses, they automatically proceed to the next PassHue.

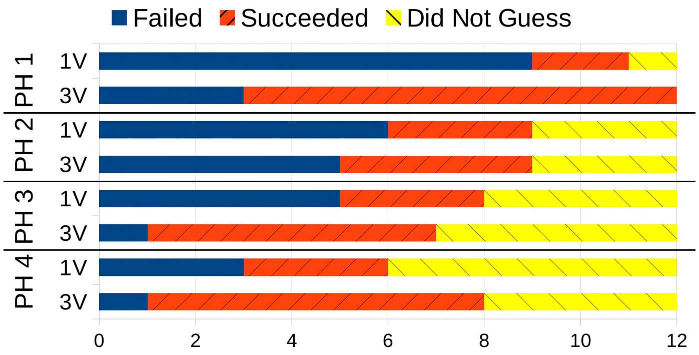


Figure 40: Shoulder-Surfing Results for PassHues 1-4 at 1 View and 3 Views (Stationary users only. PassHues were shown in chronological order from top to bottom. Successes at 1 View are also included in the 3 View total.)

There were 12 users in the Stationary condition and 4 in the Rotating condition. The results for Stationary users are shown in Figure 40. The results indicate that even Stationary PassHue has moderate resistance against shoulder-surfing. Most participants were able to guess a PassHue with 3 views and 6 guesses. With 1 view and 3 guesses, success rates never improved beyond 50%. While success rates seem to improve with practice, one confounding factor is that the study self-selects for success, participants who failed to guess a PassHue sometimes gave up and chose not move on to the next one.

Unfortunately, due to the low response rate of Rotating users, it was not possible to test hypothesis H2. Of the 4 participants in the Rotating condition, 2 were able to guess PH3 after 1 view, and 1 was able to guess PH1 and PH3 after 3 views. PH1 was not guessed with 1 view, while PH2 and PH4 were not guessed with any number of views. These preliminary results support the hypothesis that Rotating PassHue is more shoulder-surfing resistant. Future work will include a larger scale user study and an apples to apples comparison against other schemes using the same approach.

Although camera attacks were not considered in this chapter, it should be noted that PassHue, especially in the Rotating condition, is resistant against black-and-white only cameras, and against cameras with poor color resolution.

6.8 Discussion

6.8.1 Color Blindness and Tetrachromacy

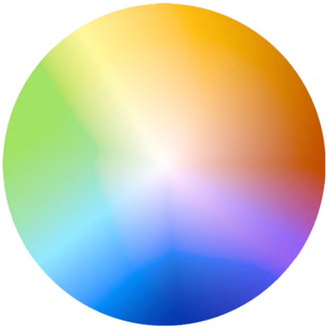


Figure 41: The Passhue Wheel Seen With Minor Deuteranomaly

Some people with minor color blindness may still be able to use PassHue if they avoid colors that cause them difficulty, by using some sort of relative position on the wheel, or by using only the most intense colors they can still see. In general, these techniques should not reduce the security strength of PassHue, since the attacker is unlikely to know if the user is color blind. Even if the attacker knows about the victim's condition, it is nearly impossible for the attacker to know *how* color blind the user is. Figure 41 demonstrates what the PassHue wheel would look like to someone with minor green color blindness (deuteranomaly), the most common form of color blindness. Subjectively, it appears to be still be usable, and distinct line patterns are now clearly visible.

As part of the demographic information, participants were asked to report if they were color blind, and if so, what type of blindness they had. One participant reported that they were red-green color blind. This participant was assigned to the Stationary condition. Password creation time was 50 seconds, and average entry time was 2.42s, slightly below average. The average number of incorrect attempts per authentication session was 1.83, again slightly worse than average.

The entry time for this participant in the first 3 days was 3.62s, declining to 1.9s over the latter 11 days. Likewise, the error rate for this participant in the first 3 days was 4.38 incorrect attempts per session, declining to .88 incorrect attempts per session for the latter 11 days of the experiment. Clearly the participant was able to find a way to use PassHue despite not having perfect color vision.

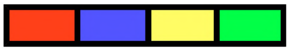


Figure 42: Color-Blind Participant's PassHue

Figure 42 shows the PassHue for this participant. Notably, the colors are less saturated and relatively far from true color values for red, blue, yellow, and green. In other words, the participant did not pick "simple" colors such as simply using the outermost points of the wheel. The participant gave the following response on their exit survey.

"Since I know I could never remember the colour I set my password according to a colour sequence which consists of the most obvious colours from each 4 groups namely red, blue, yellow, and green. Since in blue and yellow groups I can see very distinct lines of that colour I use them in the password sequence. The rest is muscle memory."

It appears that color blind people will tend to pick colors that are very different from each other, since they have a harder time discerning similar colors. This may actually lead to improved password choice. Investigating this with more color blind participants is a plan for future work.

Despite being based on the color continuum, PassHue can actually be used by someone with limited ability to discern parts of the continuum. The user can effectively substitute memorizing color with memorizing x-y location relative to some color or pattern hint.

Although roughly 8% of males of .5% of females suffer from color blindness and may not be able to use PassHue effectively, it is also estimated that 1% of the population has tetrachromatic vision, allowing them to see additional colors. By including more colors from the types that tetrachromes can better discern, a version of PassHue can be developed that is highly secure against anyone that doesn't have tetrachromatic vision. This system would require a tetrachromatic display, and finding participants with tetrachromatic vision is difficult.

6.8.2 Gender Bias

Because most kinds of color blindness occur more frequently men, and because research suggests that on average women remember color more easily and accurately than men [133], PassHue may be slightly biased towards women. Hypothesis H4 is formed based on this supposition.

H4: Females, on average, will be faster and more accurate when entering their PassHue.

Surprisingly, women were slightly slower at entering their PassHue, with average entry times of 2.69 and 2.44 seconds vs male average entry times of 2.53 and 1.68 seconds for Rotating and Stationary respectively. Mann-Whitney testing found a significant difference in entry times between genders for both conditions ($p = .006$ and $p = .016$ for Rotating and Stationary respectively). The difference in timings may be due to motor proficiency, where studies have found men to have some advantage [137].

However, the data supported the hypothesis that women would be more accurate on average. For Rotating and Stationary respectively, women made an average of 0.5 and 0.24 incorrect authentication attempts per authentication session vs male error rates of 0.87 and 1.55 incorrect attempts per session ($\chi^2 = 5.27$, $p = .022$; $\chi^2 = 35.21$, $p \leq .00001$). Additionally, the lowest performers, discussed in Section 6.7.4, were all male.

The data shows that PassHue is generally suitable for most of population, but the error rate may be quite high for a small subset of males. This may suggest that tolerances for male users should be slightly higher by default.

6.8.3 Inclusion of Additional Colors

Adding a “Value” slider to PassHue would greatly increase the password space as it would allow use of the entire RGB color space. Alternatively, there are also 2d images containing the entire RGB color space. The size of C would become all of RGB color space, $(256^3) = 16.8$ million colors. Even assuming the worst-case tolerance of 39,000, the password space is $(256^3/39000)^4 = 3.4 * 10^{10}$, about the same as a 6-character case-sensitive alphanumeric password with no symbols ($62^6 = 5.8 * 10^{10}$). Sampling 40 million random color pairs, the average product of distances is 15,000. Using this value for the tolerance, the password space is $(256^3/15000)^4 = 1.5 * 10^{12}$, roughly on-par with a 7-character alphanumeric password ($62^7 = 3.5 * 10^{12}$). Extending PassHue to the entire RGB color space without impacting usability is a plan for future work.

CHAPTER VII

CONCLUSION

7.1 Summary

Five novel proof-of-concept authentication methods were presented in this work.

1. CMAPS – Chess-based MAPS, a proof-of-concept *Multi-dimensionAl Authentication Scheme* (MAPS), demonstrated that fusing multiple dimensions into one action can greatly increase password space without excessively impacting usability or memorability. CMAPS is able to exceed the security strength of an 8 character alphanumeric password with just 6 gestures.

2. PassGame – An extension of CMAPS and the concept of MAPS, PassGame was a *challenge-response* authentication scheme, designed specifically to counter shoulder-surfing while retaining high overall security strength. PassGame proved itself extremely resilient against shoulder-surfing, able to resist against dozens of observations even with unlimited authentication attempts allowed on the device. Against all but the most dedicated attackers, PassGame offers nearly total resistance from observation-based attacks.

3. 3DPass – Extending MAPS to virtual reality (VR), 3DPass demonstrates the concept of 3D authentication. A foundation for 3Dpasswords is established in various physical and psychological advantages. 3DPass, utilizing these advantages, demonstrates the ability of 3D authentication to generate a massive password space while maintaining high usability and memorability. Most importantly, 3DPass demonstrates the superiority of 3D-based authentication versus traditional alphanumeric authentication when the user is already inside a 3D context. In a future filled with VR enabled devices, 3Dpasswords may become an appealing choice for authentication.

4. Typing Authentication – Applying the idea of MAPS to biometric behavioral passive authentication, this chapter demonstrates that high accuracy can be achieved when many dimensions, including device acceleration, are used for touch identification. While keystroke dynamics on mobile has already been tried and proven, this chapter shows how accuracy can be improved by using information from multiple dimensions at once.

5. PassHue – Analog authentication uses continuous information for authentication rather than traditional discrete information. PassHue, based on the color continuum, demonstrates excellent security strength, memorability, usability, and shoulder-surfing resistance, all while maintaining a very PIN-like environment. PassHue offers a way to users to transition to more secure authentication without drastically changing their user experience. As a proof-of-concept for analog authentication, PassHue demonstrates that continuous information can be viable in terms of security, memorability, and usability.

Each method addresses issues in current mobile authentication schemes, from low security to shoulder-surfing resistance, and any one of them could one day find itself as the basis for the mobile authentication scheme of the future.

7.2 Future Work

7.2.1 Planned Improvements

Some decisions made during the planning of user studies ended up leading to substandard results. The most significant error was using a single authentication attempt to obtain entry time data for CMAPS and PassGame. This decision was made based on similar experiments conducted prior to 2010, published primarily in security conferences and journals, where usability was not a very significant concern. Additionally, both CMAPS and PassGame experiments ran for a rather long time, and it was assumed that additional timing attempts would take too long, especially when multiple participants were queued to use the same device. Lastly, security was the primary focus of both of these works, and measuring usability was considered something of an afterthought. While it is true that both schemes are security-focused, it is equally true that regardless of security strength, a scheme with poor entry times has no chance of adoption on the mobile platform.

It was assumed that allowing just one attempt would paint a realistic picture of entry times. In practice, rather than generating an unbiased report of entry time, the single-attempt approach was subject to distracted participants, and more commonly to participants who spent a considerable amount of time on the act of remembering the password. Since participants were not instructed to prioritize speed, they didn't, and entry times for both CMAPS and PassGame are considerably poorer than expected as a result.

For these reasons, most modern works seeking publication outside of security-based venues take an approach similar to 3DPass, where entry times are taken from several attempts conducted after the password has already been remembered. This generates a far more realistic picture of entry time, which is ultimately the goal in lab experimentation.

A planned improvement to CMAPS and PassGame is to adopt the methodology from 3DPass in a new usability experiment. Participants should be instructed to prioritize speed, and entry time data should be averaged from a few attempts taken after the participant has already demonstrated recall. Of course, an in-the-wild experiment would be even more preferable, providing an even more realistic picture of real-world entry time.

Another possible oversight was the inclusion of reminder emails in CMAPS and PassGame studies. While these emails may have had an impact on memorability, there were simply not enough users forgetting their passwords to collect significant information as to their impact. A study where significance could be found would likely need hundreds of users, which is infeasible. In reality, it would be more impressive if these schemes were memorable without reminders, and it would be more interesting to compare 2 week memorability as in the 3DPass study against traditional authentication. Once again, of course, an in-the-wild study would be preferable to study memorability in real-world terms.

The exclusion of a control group using an existing scheme is another possible omission in the CMAPS, PassGame, and PassHue studies. It was assumed that the time burden of a within-subjects study would be too great for participants, and there was simply not enough participation interest for a between-subjects study as originally planned. Since participants were already familiar with PIN/Pattern Unlock, participants would be biased towards those schemes, which could confound a between-subjects study and divert attention away from the actual scheme being tested. Furthermore, the usability and memorability of passwords, PINs, and Pattern Unlocks has already been well-studied by other authors. Especially for PassHue, which is already deployed in-the-wild, a fair comparison is easy to make.

In summary, a within-subjects study was too time-consuming for participants and potentially would damage results for the scheme being tested, while a between-subjects was infeasible for recruitment reasons and largely irrelevant because plenty of data is already available for PIN, alphanumeric, and Pattern Unlock schemes. Of course, if a between-subjects study can be made feasible in the future, CMAPS and PassGame will likely tested against PIN and alphanumeric schemes as well, but it is not a priority.

The reader may ask why 3DPass was given a within-subjects study using an alphanumeric control group. 3DPass was the most “different” authentication method, and the effect of remembering something in a 3D vs 2D context at the same time is not something that is well-studied. Comparing 3DPass against one-week memorability results for alphanumeric passwords wouldn’t be quite fair, because the act of switching between 3D and 2D contexts could confound memorability results. Indeed, alphanumeric password memorability results were even poorer than expected, indicating that when faced with two types of passwords to memorize, the brain prioritizes the 3D one.

7.2.2 Upcoming Works

Several authentication schemes are already planned for production.

In the domain of analog authentication, Android has recently updated the vibration functionality on their newest devices to support true analog vibration, with control over both strength and pulse duration (previously, only pulse duration was under developer control). A vibration-based analog authentication scheme is in the works.

In the domain of biometric authentication, many mobile manufacturers have started to offer very high resolution front facing cameras that feature facial recognition technology. While these technologies, such as FaceID, follow a tradition of relative insecurity in biometrics, the addition of facial recognition combined with tracking technology (like the Kinect’s) can be used to develop a new authentication method based on movements of the user’s head. For example, the user can authenticate with a series of winks and nods. The feasibility of such a scheme from a development standpoint is currently not certain, but it will likely be possible in the near future as APIs are added to deal with facial recognition.

BIBLIOGRAPHY

[1] D. Yadron, "Man behind the first computer password: Its become a nightmare," 2014. [Online]. Available: https://blogs.wsj.com/digits/2014/05/21/the-man-behind-the-first-computer-password-its-become-a-nightmare/

[2] C. Herley, P. Van Oorschot, and A. Patrick, "Passwords: If we're so smart, why are we still using them?" *Financial Cryptography and Data Security*, pp. 230–237, 2009.

[3] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th International Conference on World Wide Web*. ACM, 2007, pp. 657–666.

[4] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis of chinese web passwords," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 559–574.

[5] A. S. Patrick, A. C. Long, and S. Flinn, "Hci and security systems," in *CHI'03 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2003, pp. 1056–1057.

[6] B. Chaparro, B. Nguyen, M. Phan, A. Smith, and J. Teves, "Keyboard performance: ipad versus netbook," *Usability News*, vol. 12, no. 2, pp. 1–9, 2010.

[7] I. S. MacKenzie, S. X. Zhang, and R. W. Soukoreff, "Text entry using soft keyboards," *Behaviour & Information Technology*, vol. 18, no. 4, pp. 235–244, 1999.

[8] F. Schaub, R. Deyhle, and M. Weber, "Password entry usability and shoulder surfing susceptibility on different smartphone platforms," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 2012, pp. 13–23.

[9] M. Harbach, E. von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith, "It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception," in *Symposium on Usable Privacy and Security*, 2014, pp. 213–230.

[10] E. Von Zezschwitz, A. De Luca, and H. Hussmann, "Honey, i shrunk the keys: influences of mobile devices on password composition and authentication performance," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. ACM, 2014, pp. 461–470.

[11] S. Harvey and D. Evans, "Defending against cyber espionage: The us office of personnel management hack as a case study in information assurance," *NCUR*, 2016.

[12] M. Wachtel, "Give me your password because congress can say so: An analysis of fifth amendment protection afforded individuals regarding compelled production of

encrypted data and possible solutions to the problem of getting data from someone's mind," *Pitt. J. Tech. L. & Pol'y*, vol. 14, pp. 44–351, 2013.

[13] A. M. Clemens, R. Salgado, and C. Genetski, "No computer exception to the constitution: The fifth amendment protects against compelled production of an encrypted document or private key," *UCLA JL & TECH.*, pp. 1–27, 2004.

[14] J. R. Atwood, "The encryption problem: Why the courts and technology are creating a mess for law enforcement," *St. Louis U. Pub. L. Rev.*, vol. 34, pp. 407–465, 2015.

[15] A. M. Geshowitz, "Password protected-can a password save your cell phone from a search incident to arrest," *Iowa L. Rev.*, vol. 96, pp. 1125–1791, 2010.

[16] D. Ovalle, "He didnt give police his iphone pass code, so he got 180 days in jail," 2017. [Online]. Available: http://www.miamiherald.com/news/local/community/broward/article153373524.html

[17] F. Rieger, "Chaos computer club breaks apple touchid," 2013. [Online]. Available: https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid

[18] T. Fox-Brewster, "Apple face id fooled again – this time by $200 evil twin mask," 2017. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2017/11/27/apple-face-id-artificial-intelligence-twin-mask-attacks-iphone-x/#2a374e027754

[19] J. S. Krisller, "Chaos computer clubs breaks iris recognition system of the samsung galaxy s8," 2017. [Online]. Available: https://www.ccc.de/en/updates/2017/iriden

[20] K. Olmstead and A. Smith, "Americans and cybersecurity," Jan 2017. [Online]. Available: http://www.pewinternet.org/2017/01/26/2-password-management-and-mobile-security/

[21] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological review*, vol. 63, no. 2, pp. 81–97, 1956.

[22] N. Cowan, "The magical mystery four: How is working memory capacity limited, and why?" *Current Directions in Psychological Science*, vol. 19, no. 1, pp. 51–57, 2010.

[23] J. Gurary, Y. Zhu, G. Corser, J. Oluoch, N. Alnahash, and H. Fu, "Maps: A multi-dimensional password scheme for mobile authentication," in *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. ACM, 2015, pp. 409–412.

[24] C. Herley and P. Van Oorschot, "A research agenda acknowledging the persistence of passwords," *IEEE Security Privacy*, vol. 10, no. 1, pp. 28–36, 2012.

[25] A. L. Titcomb, V. F. Reyna, F. Dempster, and C. Brainerd, "Memory interference and misinformation effects," *Interference and Inhibition in Cognition*, pp. 263–294, 1995.

[26] G. Blonder, "Graphical password," Sep. 1996, patent 5,559,961.

[27] A. Eljetlawi and N. Ithnin, "Graphical password: Comprehensive study of the usability features of the recognition base graphical password methods," in *Third International Conference on Convergence and Hybrid Information Technology*, vol. 2, Nov. 2008, pp. 1137–1143.

[28] M. Defetyer, R. Russo, and P. McPartlin, "The picture superiority effect in recognition memory: a developmental study using the response signal procedure," *Cognitive Development*, vol. 24, no. 1, pp. 265–273, 2009.

[29] M. S. Weldon, H. L. Roediger, and B. H. Challis, "The properties of retrieval cues constrain the picture superiority effect," *Memory & Cognition*, vol. 17, no. 1, pp. 95–105, 1989.

[30] T. L. Childers and M. J. Houston, "Conditions for a picture-superiority effect on consumer memory," *Journal of Consumer Research*, vol. 11, no. 2, pp. 643–654, 1984.

[31] M. Hafiz, A. Abdullah, N. Ithnin, and H. Mammi, "Towards identifying usability and security features of graphical password in knowledge based authentication technique," in *Modeling Simulation, 2008. Second Asia International Conference on*, May 2008, pp. 396–403.

[32] T. S. Tullis, D. P. Tedesco, and K. E. McCaffrey, "Can users remember their pictorial passwords six years later," in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 2011, pp. 1789–1794.

[33] R. Biddle, S. Chiasson, and P. Van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Computing Surveys (CSUR)*, vol. 44, no. 4, pp. 1–41, Sep. 2012.

[34] J. G. Raaijmakers and R. M. Shiffrin, "Models for recall and recognition." *Annual Review of Psychology*, vol. 43, pp. 205–234, 1992.

[35] R. Dhamija and A. Perrig, "Deja vu: A user study using images for authentication," in *Proceedings of the 9th Conference on USENIX Security Symposium SSYM'00*. USENIX Association, 2000.

[36] RealUser, "Passfaces: Two factor authentication for the enterprise." [Online]. Available: http://www.realuser.com

[37] D. Davis, F. Monrose, and M. K. Reiter, "On user choice in graphical password schemes," in *Proceedings of the 13th Conference on USENIX Security Symposium SSYM'04*. USENIX Association, 2004.

[38] J. Nicholson, L. Coventry, and P. Briggs, "Age-related performance issues for pin and face-based authentication systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 323–332.

[39] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *Proceedings of the 8th Conference on USENIX Security Symposium SSYM'99*. USENIX Association, 1999.

[40] A. De Luca, M. Harbach, E. von Zezschwitz, M.-E. Maurer, B. E. Slawik, H. Hussmann, and M. Smith, "Now you see me, now you don't: protecting smartphone authentication from shoulder surfers," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 2937–2946.

[41] Y. Yang, G. D. Clark, J. Lindqvist, and A. Oulasvirta, "Free-form gesture authentication in the wild," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 3722–3735.

[42] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system," *International journal of human-computer studies*, vol. 63, no. 1, pp. 102–127, 2005.

[43] J. Thorpe and P. C. van Oorschot, "Human-seeded attacks and exploiting hot-spots in graphical passwords," in *16th USENIX Security Symposium*, 2007, pp. 103–118.

[44] A. Sadovnik and T. Chen, "A visual dictionary attack on picture passwords," *IEEE International Conference on Image Processing*, 2013.

[45] Z. Zhao, G.-J. Ahn, J.-J. Seo, and H. Hu, "On the security of picture gesture authentication," in *Proceedings of the 22Nd USENIX Conference on Security SEC'13*. USENIX Association, 2013, pp. 383–398.

[46] M. Hosenball, "Fbi paid under 1 million to unlock san bernardino iphone: Sources," May 2016. [Online]. Available: https://www.reuters.com/article/us-apple-encryption/fbi-paid-under-1-million-to-unlock-san-bernardino-iphone-sources-idUSKCN0XQ032

[47] E. von Zezschwitz, A. De Luca, B. Brunkow, and H. Hussmann, "Swipin: Fast and secure pin-entry on smartphones," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1403–1406.

[48] A. De Luca, K. Hertzschuch, and H. Hussmann, "Colorpin: securing pin entry through indirect input," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 1103–1106.

[49] A. Bianchi, I. Oakley, V. Kostakos, and D. S. Kwon, "The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices," in *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2011, pp. 197–200.

[50] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: the case of android unlock patterns," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. ACM, 2013, pp. 161–172.

[51] P. Andriotis, T. Tryfonas, and G. Oikonomou, "Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method," *Human Aspects of Information Security, Privacy, and Trust*, pp. 115–126, 2014.

[52] T. Kwon and S. Na, "Tinylock: Affordable defense against smudge attacks on smartphone pattern lock systems," *Computers & Security*, vol. 42, pp. 137–150, 2014.

[53] W. A. van Eekelen, J. van den Elst, and V.-J. Khan, "Picassopass: a password scheme using a dynamically layered combination of graphical elements," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 1857–1862.

[54] A. Bianchi, I. Oakley, and D. S. Kwon, "Counting clicks and beeps: Exploring numerosity based haptic and audio pin entry," *Interacting with Computers*, vol. 24, no. 5, pp. 409–422, 2012.

[55] H. Sasamoto, N. Christin, and E. Hayashi, "Undercover: authentication usable in front of prying eyes," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 183–192.

[56] A. De Luca, E. Von Zezschwitz, and H. Hußmann, "Vibrapass: secure authentication based on shared lies," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2009, pp. 913–916.

[57] N. Wright, A. S. Patrick, and R. Biddle, "Do you see your password?: applying recognition to textual passwords," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*. ACM, 2012.

[58] N. Hikmet and S. K. Chen, "An investigation into low mail survey response rates of information technology users in health care organizations," *International Journal of Medical Informatics*, vol. 72, no. 1, pp. 29–34, 2003.

[59] H. Gao, Z. Ren, X. Chang, X. Liu, and U. Aickelin, "A new graphical password scheme resistant to shoulder-surfing," in *International Conference on Cyberworlds (CW)*. IEEE, 2010, pp. 194–199.

[60] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proceedings of the Seventh Symposium on Usable Privacy and Security*. ACM, 2011.

[61] F. Alt, S. Schneegass, A. S. Shirazi, M. Hassib, and A. Bulling, "Graphical passwords in the wild: Understanding how users choose pictures and passwords in

image-based authentication schemes," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2015, pp. 316–322.

[62] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

[63] Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: An algorithmic framework and empirical analysis," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010, pp. 176–186.

[64] J. Gurary, Y. Zhu, N. Alnahash, and H. Fu, "Passgame: A shoulder-surfing resistant mobile authentication scheme," in *Advances in Computer-Human Interactions*, Mar. 2017.

[65] ——, "Passgame: Robust shoulder-surfing resistance through challenge-response authentication," *International Journal On Advances in Security*, vol. 10, Dec. 2017.

[66] F. Tari, A. A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proceedings of the Second Symposium on Usable Privacy and Security*, 2006, pp. 56–66.

[67] X. Suo, Y. Zhu, and G. Owen, "Graphical passwords: a survey," in *21st Annual Computer Security Applications Conference*, Dec. 2005, pp. 462–472.

[68] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens." *Woot*, vol. 10, pp. 1–7, 2010.

[69] V. Roth, K. Richter, and R. Freidinger, "A pin-entry method resilient against shoulder surfing," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 2004, pp. 236–245.

[70] D. Lin, P. Dunphy, P. Olivier, and J. Yan, "Graphical passwords & qualitative spatial relations," in *Proceedings of the 3rd Symposium on Usable Privacy and Security*, 2007, pp. 161–162.

[71] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, 2006, pp. 177–184.

[72] A. De Luca, E. Von Zezschwitz, N. D. H. Nguyen, M.-E. Maurer, E. Rubegni, M. P. Scipioni, and M. Langheinrich, "Back-of-device authentication on smartphones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2389–2398.

[73] C. Winkler, J. Gugenheimer, A. De Luca, G. Haas, P. Speidel, D. Dobbelstein, and E. Rukzio, "Glass unlock: Enhancing security of smartphone unlocking through leveraging a private near-eye display," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, vol. 15, 2015, pp. 1407–1410.

[74] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proceedings of the 3rd Symposium on Usable Privacy and Security*, 2007, pp. 13–19.

[75] M. Swift and B. Shah, "Challenge-response authentication and key exchange for a connectionless security protocol," Apr. 2002, uS Patent 6,377,691.

[76] C. Perkins, "Mobile ipv4 challenge/response extensions," Nov 2000. [Online]. Available: http://tools.ietf.org/html/rfc3012

[77] S. Kumar and V. Kumar, "Keyless encryption of messages using challenge response," Mar 2003, uS Patent 6,535,980.

[78] T. Takada and M. Ishizuka, "Chameleon dial: repeated camera-recording attack resilient pin input scheme," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2015, pp. 365–368.

[79] P. Shi, B. Zhu, and A. Youssef, "A pin entry scheme resistant to recording-based shoulder-surfing," in *Third International Conference on Emerging Security Information, Systems and Technologies (Secureware)*, 2009, pp. 237–241.

[80] W.-C. Ku, D.-M. Liao, C.-J. Chang, and P.-J. Qiu, "An enhanced capture attacks resistant text-based graphical password scheme," in *International Conference on Communications in China (ICCC)*, 2014, pp. 204–208.

[81] J. Gurary, Y. Zhu, and H. Fu, "Leveraging 3d benefits for authentication," *International Journal of Communications, Network and System Sciences*, vol. 10, no. 08, p. 324, 2017.

[82] Gartner, "Gartner says worldwide wearable devices sales to grow 18.4 percent in 2016," 2016. [Online]. Available: http://www.gartner.com/newsroom/id/3198018

[83] F. A. Alsulaiman and A. El Saddik, "Three-dimensional password for more secure authentication," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 1929–1938, 2008.

[84] N. Salian, S. Godbole, and S. Wagh, "Advanced authentication using 3d passwords in virtual world," *International Journal of Engineering and Technical Research*, vol. 3, no. 2, 2015.

[85] V. Kolhe, V. Gunjal, S. Kalasakar, and P. Rathod, "Secure authentication with 3d password," *International Journal Of Engineering Science And Innovative Technology (IJESIT)*, 2013.

[86] G. Aman and N. Winnie, "4-d password: Strengthening the authentication scene," *International Journal of Scientific & Engineering Research*, vol. 3, no. 10, 2012.

[87] R. M. Baños, C. Botella, I. Rubió, S. Quero, A. García-Palacios, and M. Alcañiz, "Presence and emotions in virtual environments: The influence of stereoscopy," *CyberPsychology & Behavior*, vol. 11, no. 1, pp. 1–8, 2008.

[88] R. M. Baños, C. Botella, M. Alcañiz, V. Liaño, B. Guerrero, and B. Rey, "Immersion and emotion: their impact on the sense of presence," *CyberPsychology & Behavior*, vol. 7, no. 6, pp. 734–741, 2004.

[89] M. Slater, V. Linakis, M. Usoh, R. Kooper, and G. Street, "Immersion, presence, and performance in virtual environments: An experiment with tri-dimensional chess," in *ACM Virtual Reality Software and Technology (VRST)*. ACM, 1996, pp. 163–172.

[90] E. Teng and L. R. Squire, "Memory for places learned long ago is intact after hippocampal damage," *Nature*, vol. 400, no. 6745, pp. 675–677, 1999.

[91] E. E. Smith, J. Jonides, R. A. Koeppe, E. Awh, E. H. Schumacher, and S. Minoshima, "Spatial versus object working memory: Pet investigations," *Journal of Cognitive Neuroscience*, vol. 7, no. 3, pp. 337–356, 1995.

[92] N. J. Broadbent, L. R. Squire, and R. E. Clark, "Spatial memory, recognition memory, and the hippocampus," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 40, pp. 14 515–14 520, 2004.

[93] E. Attree, B. Brooks, F. Rose, T. Andrews, A. Leadbetter, and B. Clifford, "Memory processes and virtual environments: I cant remember what was there, but i can remember how i got there. implications for people with disabilities," in *ECDVRAT: 1st European Conference on Disability, Virtual Reality and Associated Technologies*, vol. 118, 1996.

[94] E. C. Tolman, "Cognitive maps in rats and men," *Psychological Review*, vol. 55, no. 4, pp. 189–208, 1948.

[95] E. Tulving, "Episodic and semantic memory 1," *Organization of Memory. London: Academic*, vol. 381, no. 4, pp. 382–404, 1972.

[96] S. M. Smith, "Remembering in and out of context," *Journal of Experimental Psychology: Human Learning and Memory*, vol. 5, no. 5, pp. 460–471, 1979.

[97] I. Begg and J. M. Clark, "Contextual imagery in meaning and memory," *Memory & Cognition*, vol. 3, no. 2, pp. 117–122, 1975.

[98] A. Neil, "Autostereoscopic 3d displays," *Computer*, vol. 8, pp. 32–36, 2005.

[99] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: the design and implementation of the cave," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1993, pp. 135–142.

[100] W. IJsselsteijn, H. de Ridder, J. Freeman, S. E. Avons, and D. Bouwhuis, "Effects of stereoscopic presentation, image motion, and screen size on subjective and objective corroborative measures of presence," *Presence*, vol. 10, no. 3, pp. 298–311, 2001.

[101] S. H. Ferris, "Motion parallax and absolute distance," *Journal of Experimental Psychology*, vol. 95, no. 2, pp. 258–263, 1972.

[102] C. Hendrix and W. Barfield, "Presence within virtual environments as a function of visual display parameters," *Presence: Teleoperators & Virtual Environments*, vol. 5, no. 3, pp. 274–289, 1996.

[103] W. Barfield, C. Hendrix, and K.-E. Bystrom, "Effects of stereopsis and head tracking on performance using desktop virtual environment displays," *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 2, pp. 237–240, 1999.

[104] H. G. Hoffman, W. J. Meyer III, M. Ramirez, L. Roberts, E. J. Seibel, B. Atzori, S. R. Sharar, and D. R. Patterson, "Feasibility of articulated arm mounted oculus rift virtual reality goggles for adjunctive pain control during occupational therapy in pediatric burn patients," *Cyberpsychology, Behavior, and Social Networking*, vol. 17, no. 6, pp. 397–401, 2014.

[105] M. Tavanti and M. Lind, "2d vs 3d, implications on spatial memory," in *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on.* IEEE, 2001, pp. 139–145.

[106] P. Isokoski and B. Martin, "Eye tracker input in first person shooter games," in *Proceedings of the 2nd Conference on Communication by Gaze Interaction: Communication by Gaze Interaction-COGAIN 2006: Gazing into the Future*, 2006, pp. 78–81.

[107] A. Davidson, "An evaluation of visual gesture based controls for exploring three dimensional environments," Ph.D. dissertation, School of Computer Science & Statistics, 2012.

[108] C. Ardito, P. Buono, M. F. Costabile, R. Lanzilotti, and A. L. Simeone, "Comparing low cost input devices for interacting with 3d virtual environments," in *2nd Conference on Human System Interactions.* IEEE, 2009, pp. 292–297.

[109] J. C. Coelho and F. J. Verbeek, "Pointing task evaluation of leap motion controller in 3d virtual environment," *Creating the Difference*, vol. 78, 2014.

[110] Builderhouseplans, "Tavern-like features," 2016. [Online]. Available: http://www.builderhouseplans.com/house-plans/bhp/hwbdo69293.html

[111] E. L. Newman, J. B. Caplan, M. P. Kirschen, I. O. Korolev, R. Sekuler, and M. J. Kahana, "Learning your way around town: How virtual taxicab drivers learn to use both layout and landmark information," *Cognition*, vol. 104, no. 2, pp. 231–253, 2007.

[112] T. Schubert, F. Friedmann, and H. Regenbrecht, "The experience of presence: Factor analytic insights," *Presence*, vol. 10, no. 3, pp. 266–281, 2001.

[113] D. Bachmann, F. Weichert, and G. Rinkenauer, "Evaluation of the leap motion controller as a new contact-free pointing device," *Sensors*, vol. 15, no. 1, pp. 214–233, 2014.

[114] J. Gurary, Y. Zhu, N. Alnahash, and H. Fu, "Implicit authentication for mobile devices using typing behavior," in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 2016, pp. 25–36.

[115] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor," *Journal of Computers*, vol. 1, no. 7, pp. 51–59, 2006.

[116] J. P. Campbell Jr, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[117] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 955–966, 1995.

[118] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri, "Keystroke dynamics authentication for mobile phones," in *Proceedings of the 2011 ACM Symposium on Applied Computing*. ACM, 2011, pp. 21–26.

[119] H. Saevanee and P. Bhattarakosol, "Authenticating user using keystroke dynamics and finger pressure," in *Consumer Communications and Networking Conference (CCNC 2009)*. IEEE, 2009, pp. 1–2.

[120] A. Buchoux and N. L. Clarke, "Deployment of keystroke analysis on a smartphone," in *Australian Information Security Management Conference*, 2008.

[121] M. Trojahn and F. Ortmeier, "Biometric authentication through a virtual keyboard for smartphones," *International Journal of Computer Science & Information Technology*, vol. 4, no. 5, 2012.

[122] M. Frank, R. Biedert, E.-D. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[123] B. Draffin, J. Zhu, and J. Zhang, "Keysens: passive user authentication through micro-behavior modeling of soft keyboard interaction," in *Mobile Computing, Applications, and Services*. Springer, 2014, pp. 184–201.

[124] M. Antal, L. Z. Szabó, and I. László, "Keystroke dynamics on android platform," *Procedia Technology*, vol. 19, pp. 820–826, 2015.

[125] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. K. Nguyen, "Continuous mobile authentication using touchscreen gestures," in *2012 IEEE Conference on Technologies for Homeland Security (HST)*. IEEE, 2012, pp. 451–456.

[126] J. Gurary, “Analog authentication (working title),” in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, Jul. 2018.

[127] M. Sherman, G. Clark, Y. Yang, S. Sugrim, A. Modig, J. Lindqvist, A. Oulasvirta, and T. Roos, “User-generated free-form gestures for authentication: Security and memorability,” in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2014, pp. 176–189.

[128] G. D. Clark and J. Lindqvist, “Engineering gesture-based authentication systems,” *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 18–25, 2015.

[129] M. Pointer and G. Attridge, “The number of discernible colours,” *Color Research & Application*, vol. 23, no. 1, pp. 52–54, 1998.

[130] D. B. Judd and G. Wyszecki, *Color in business, Science and Industry*. John Wiley and Sons. Ltd., 1963.

[131] R. M. Halsey and A. Chapanis, “Chromaticity-confusion contours in a complex viewing situation,” *JOSA*, vol. 44, no. 6, pp. 442–454, 1954.

[132] V. Hamwi and C. Landis, “Memory for color,” *The Journal of Psychology*, vol. 39, no. 1, pp. 183–194, 1955.

[133] J. Pérez-Carpinell, R. Baldoví, M. D. de Fez, and J. Castro, “Color memory matching: Time effect and other factors,” *Color Research & Application*, vol. 23, no. 4, pp. 234–247, 1998.

[134] CompuPhase, “Color metric,” Apr. 2012. [Online]. Available: https://www.compuphase.com/cmetric.htm

[135] E. Von Zezschwitz, P. Dunphy, and A. De Luca, “Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices,” in *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 2013, pp. 261–270.

[136] H. Kim and J. H. Huh, “Pin selection policies: Are they really effective?” *Computers & Security*, vol. 31, no. 4, pp. 484–496, 2012.

[137] C. E. Plimpton and C. Regimbal, “Differences in motor proficiency according to gender and race,” *Perceptual and Motor Skills*, vol. 74, no. 2, pp. 399–402, 1992.