8-2014

Integrating Visual Mnemonics and Input Feedback with Passphrases to Improve the Usability and Security of Digital Authentication

Kevin Juang
Clemson University, kjuang@clemson.edu

INTEGRATING VISUAL MNEMONICS AND INPUT FEEDBACK WITH PASSPHRASES TO IMPROVE THE USABILITY AND SECURITY OF DIGITAL AUTHENTICATION

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Industrial Engineering

---

by
Kevin A. Juang
August 2014

Accepted by:
Joel S. Greenstein, Committee Chair
Mary E. Kurz
Richard Pak
Patrick J. Rosopa

ABSTRACT

The need for both usable and secure authentication is more pronounced than ever before. Security researchers and professionals will need to have a deep understanding of human factors to address these issues. Due to their ubiquity, recoverability, and low barrier of entry, passwords remain the most common means of digital authentication. However, fundamental human nature dictates that it is exceedingly difficult for people to generate secure passwords on their own. System-generated random passwords can be secure but are often unusable, which is why most passwords are still created by humans. We developed a simple system for automatically generating mnemonic phrases and supporting mnemonic images for randomly generated passwords. We found that study participants remembered their passwords significantly better using our system than with existing systems. To combat shoulder surfing – looking at a user's screen or keyboard as he or she enters sensitive input such as passwords – we developed an input masking technique that was demonstrated to minimize the threat of shoulder surfing attacks while improving the usability of password entry over existing methods. We extended this previous work to support longer passphrases with increased security and evaluated the effectiveness of our new system against traditional passphrases. We found that our system exhibited greater memorability, increased usability and overall rankings, and maintained or improved upon the security of the traditional passphrase systems. Adopting our passphrase system will lead to more usable and secure digital authentication.

DEDICATION

I would like to dedicate this dissertation to my family, because sticks in a bundle can't be broken.

ACKNOWLEDGMENTS

Naturally, the first person I need to thank is my advisor, Joel Greenstein. I have never met anyone else who is even in the same league as him when it comes to academic mentoring, which is why he is on so many committees and why his opinions are in such high demand. Joel's greatest strength is his ability to adapt to each individual student, providing extra guidance or flexibility (in my case) as warranted. As great of a teacher as he is, he truly is an even greater person – I know that must have sounded trite, but I mean it! Joel, I honestly would not have pursued a PhD with anyone else.

To my committee members: you guys are the best, which is why I personally hand selected all of you to join my academic association of superfriends. Seriously, you have each contributed insightful comments every time we've met, and for that I'll always be grateful.

From what I've seen in other departments and at other schools, I am certain that we have the best staff in Freeman, bar none. You constantly went far beyond what you were required to do for me, just because you are such helpful and awesome people. I'm really going to miss all the great talks we've had these past several years!

I also need to thank my labmates and colleagues for grinding it out with me daily in the trenches. I know the rest of you will soon graduate too! I've made a tremendous number of friends here during my time at Clemson. I don't want to leave anyone out, so just know that I treasure the friendship I share with each and every one of you.

It’s rare in our department for a graduate student to teach classes, so I feel incredibly fortunate to have been thrust into teaching three years of IE 201. It caught me by surprise, but this has been my favorite and most rewarding experience at Clemson. Students, I never expected this, but reading all of your amazing comments in the teaching evaluations has been one of the highlights of my entire life so far. If it makes any sense, I get pumped up like the hero in a movie who’s about to be defeated by the villain but gets back up after receiving all the prayers from everyone back home.

Finally, of course I have to thank my wonderful family, who mean much more to me than they could even imagine. My brother Michael will always be my best friend for life, but he already knows that. Well, Mom and Dad, you know I love you, so how about I graduate, make a ton of cash, and start paying you back for all those years of everything?

I think happiness is just being able to loaf without stress, so I’m done here.

TABLE OF CONTENTS

Page

TITLE PAGE i

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

LIST OF FIGURES ix

CHAPTER

I. INTRODUCTION 1

II. LITERATURE REVIEW 5

Digital Authentication 5
Passwords 6
Technical Topics in Password Theory 8
Mnemonics for Passwords 12
Input Masking 15
Passphrases 16
Integration of Techniques 21

III. MNEMONICS TO AID SYSTEM-GENERATED PASSWORD RETENTION 23

Introduction 23
Implementation 24
Method 26
Threat Model and Limitations 26
Subjects 28
Experimental Design 28
Procedure 29
Dependent Variables 30
Results 31
Objective Measures 32
Subjective Measures 36

Table of Contents (Continued)

Page

Discussion 38

IV. METHODS OF MNEMONIC CREATION FOR AUTHENTICATION 41

Introduction 41
Method 42
Experimental Design 42
Participants 42
Procedure 43
Measures 45
Results 46
Discussion 52

V. INPUT MASKING TECHNIQUES FOR AUTHENTICATION 55

Introduction 55
Implementation 56
Method 58
Threat Model and Limitations 58
Subjects 59
Experimental Design 60
Procedure 61
Dependent Variables 62
Results 63
Objective Measures 63
Subjective Ratings 68
Preference Rankings 70
Discussion 72

VI. INTEGRATING MNEMONICS, INPUT FEEDBACK, AND PASSPHRASES 75

Introduction 75
Implementation 76
Implementation Considerations 78
Method 84
Independent Variable 84
Dependent Variables 93

Table of Contents (Continued)

Page

Hypotheses 95
Participants 97
Procedure 99
Results 101
Data Analysis 101
Machine Cracking 102
Objective Measures 104
Subjective Measures 119
Discussion 126
Hypotheses 126
Observations 130
Responses 134
Conclusion 137

VII. CONCLUSION 138

REFERENCES 141

LIST OF FIGURES

Figure — Page

3.1 The system-generated mnemonic is designed to be simple. 24

3.2 The mnemonic phrase and picture complement each other as simple, effective memory aids. 26

3.3 2nd-session Jaro-Winkler proximity (larger values indicate increased effectiveness) 33

3.4 2nd-session success rate (larger values indicate increased effectiveness) 34

3.5 Adversarial Damerau-Levenshtein distance (larger values indicate increased security) 35

3.6 Adversarial Jaro-Winkler proximity (smaller values indicate increased security) 36

3.7 2nd-session SUS score (larger values indicate increased usability) 37

4.1 Account creation time (smaller values indicate increased efficiency) 47

4.2 Success rate: login (larger values indicate increased effectiveness) 48

4.3 Damerau-Levenshtein distance: login and attack (smaller values indicate increased similarity) 49

4.4 Jaro-Winkler proximity: login and attack (larger values indicate increased similarity) 50

4.5 SUS: creation and login (larger values indicate increased usability) 52

5.1 The decoy text obfuscates the password while also acting as a trap for potential attackers. 57

List of Figures (Continued)

Figure Page

5.2 Damerau-Levenshtein distance (larger values indicate increased security) 65

5.3 Jaro-Winkler proximity (smaller values indicate increased security) 66

5.4 Task completion time (smaller values indicate increased efficiency) 67

5.5 Error rate (smaller values indicate increased effectiveness) 68

5.6 The first three subjective usability ratings (larger values indicate increased usability) 69

5.7 The last four subjective usability ratings (larger values indicate increased usability) 70

5.8 Preference rankings (smaller values indicate increased preference) 71

6.1 The user-generated passphrase condition (#1) appears similar during creation and login. 86

6.2 The Diceware10k system-generated passphrase condition (#2) results in shorter passphrases. 88

6.3 The Special English wordlist condition (#3) allows the user to select one of four passphrases. 90

6.4 The 6-word sentence condition (#4) shows the mnemonic picture when selected by the user. 92

6.5 1st-session recall time (smaller values indicate increased efficiency) 105

6.6 1st-session Damerau-Levenshtein distance (smaller values indicate increased effectiveness) 107

List of Figures (Continued)

Figure Page

6.7 2nd-session Damerau-Levenshtein distance (smaller values indicate increased effectiveness) 108

6.8 1st-session Jaro-Winkler proximity (larger values indicate increased effectiveness) 110

6.9 2nd-session Jaro-Winkler proximity (larger values indicate increased effectiveness) 111

6.10 1st-session recall success rate (larger values indicate increased effectiveness) 112

6.11 2nd-session recall success rate (larger values indicate increased effectiveness) 113

6.12 Shoulder surfing Damerau-Levenshtein distance (larger values indicate increased security) 114

6.13 Shoulder surfing Jaro-Winkler proximity (smaller values indicate increased security) 115

6.14 Cracking Damerau-Levenshtein distance (larger values indicate increased security) 116

6.15 Cracking Jaro-Winkler proximity (smaller values indicate increased security) 117

6.16 log2-transformed cracking resist attempts (larger values indicate increased security) 119

6.17 Creation SUS score (larger values indicate increased usability) 121

6.18 1st-session recall SUS score (larger values indicate increased usability) 122

List of Figures (Continued)

Figure Page

6.19 2nd-session recall SUS score (larger values indicate increased usability) 123

6.20 Perceived security on a 7-point Likert scale (larger values indicate increased security) 125

6.21 Overall rating on a 7-point Likert scale (larger values indicate increased usability and security) 126

CHAPTER 1: INTRODUCTION

One of the predominant themes in computer security lies in authentication: addressing how to grant access to authorized users while preventing access to others. Modern widespread adoption and reliance on computer systems, and online services in particular, only strengthens the importance of this goal. The traditional focus in computer security has been on hardening the computer systems themselves, but many successful attacks have targeted the human component (Tognazzini, 2005). This suggests a need for security researchers and professionals who possess a strong background in human factors and usability.

A fundamental issue with digital authentication is the balance between security and usability. As designers increase the level of difficulty of an authentication process in order to reduce the frequency of unwanted users gaining access (misses), the occurrence of legitimate users being denied access (false alarms) likewise increases. The way to reduce both types of errors is to design systems that better differentiate between authorized and unauthorized users; this is accomplished by exploiting any differences between these two groups of users (Tognazzini, 2005).

Although the significant cost of poor security has been extensively covered, considerably less attention has traditionally been paid to the cost of poor usability in digital authentication (Sasse & Flechais, 2005). When usability is neglected, a sharp

decrease in worker productivity, returning customers, or organizational reputation can often be expected. However, the fact that poor usability can also indirectly lead to decreased security is often overlooked (Tognazzini, 2005).

People can be quite adaptable when the situation demands it. Since our desire to maintain the highest possible level of security is trumped by our desire to get things done quickly and easily, we routinely come up with workarounds that sacrifice security for usability. This is why, for example, users often reuse passwords across multiple accounts, write passwords down, or share passwords with others (Sasse & Flechais, 2005).

The challenge in improving existing systems thus results from maximizing the security benefits while simultaneously limiting the usability regressions. This is what much of the previous research has focused on – when usability has even been considered at all. Alternately, we can maximize usability improvements while minimizing security degradations. Sometimes, it may even be possible to improve both.

The purpose of this research is to apply human factors and usability design to advance the progress of digital authentication. Chapter 2 provides a review of relevant topics, starting with an overview of digital authentication and passwords in particular, including a section covering important technical topics in password theory. Mnemonics, as they relate to the field of usable security, are covered, as well as input masking and the threat of shoulder surfing. Finally, the review ends with the topic of passphrases and

how they differ from passwords, before summarizing how the integration of these usable security techniques can provide advantages beyond the individual components.

Chapters 3-5 describe novel work done to improve on the usability and security of existing digital authentication practices. Chapter 3 details the addition of a new system of mnemonic phrases and user-created pictures to aid in the retention of system-generated passwords. Chapter 4 expands on the work from the previous chapter by examining the effects of different methods of creating those mnemonic pictures on usability and security. Chapter 5 examines different input masking techniques to combat shoulder surfing during digital authentication, including a novel method intended to match the usability benefits of cleartext and the security benefits of masked text.

Chapter 6 describes a study that integrates the work from Chapters 3-5 with passphrases to help facilitate a transition from passwords to passphrases. The implementation of the new system is detailed in the process. The impetus for a transition to passphrases is increased security, but usability must also be carefully considered for this change to be successful. The results indicate that the innovations from Chapters 3-5 combine particularly effectively with passphrases to address their major shortcomings.

Chapter 7 concludes by framing the research inside the bigger picture of the field of digital authentication. First, the main contributions of the research to the field are summarized. Recommendations and applications are also covered. Finally, Chapter 7 outlines possible directions for future research. Computer security is a dynamic and

rapidly evolving subject area, and even the best solutions to existing problems must continue to adapt in order to handle the new problems that inevitably arise.

CHAPTER 2: LITERATURE REVIEW

Digital Authentication

Users can be authenticated by what they know or recognize, what they hold, or what they are (Renaud, 2005). With the first category, the user and system agree upon a shared secret during account creation, and the system tests if the user knows the secret during authentication. This secret is typically in the form of a password. The second category relies on physical or electronic tokens, while the third category, commonly known as biometrics, identifies humans by their individual traits or characteristics.

Tokens and biometrics have seen increased usage in recent years, but they have severe issues that have limited their adoption (Renaud, 2005). Hardware tokens are more expensive, can be unwieldy to transport and use (especially in multiples), and can easily be lost or stolen. Software tokens, implemented using public encryption keys and private decryption keys, are difficult for the majority of users to understand (Whitten & Tygar, 1999), only work with one particular machine, and are less secure than hardware tokens. Biometrics depend on a strictly controlled environment, can often be forged (e.g., by using a photograph to defeat face recognition), and are susceptible to lockout caused by changes in traits of legitimate users.

A significant drawback with tokens and biometrics is the difficulty in recovering from user lockout; in comparison, a password can be reset immediately, usually by the system itself upon request. Because of these combined factors, passwords remain by far the most common means of digital authentication. Much of the continued lifespan of

passwords is due to their widespread availability and understandability, traits that should not change any time in the near future (Herley & van Oorschot, 2012).

Passwords

The process of authentication can always be improved, so researchers and practitioners have been trying to replace password authentication for decades, on the grounds of poor usability, security, or both. While passwords have the potential to be extremely secure, the cognitive demand of having to remember a unique password for each account often causes users to choose weak passwords or reuse them, drastically lowering security (Renaud, 2005).

For those who argue that passwords tend to be dangerously insecure, one of the primary reasons given is that users predominantly create their own passwords. Users typically prefer to create their own passwords, but user-generated passwords have been shown to be much less secure than randomly generated passwords created by computer systems (Proctor, Lien, Vu, Schultz, & Salvendy, 2002).

Many password strengthening techniques have been implemented or suggested, such as password restrictions, periodic forced rotation of passwords, substitution of special characters for letters, per-site password modifications, and guidelines like creating a phrase and using the first letter of each word to form a password. These degrade usability but commonly do not lead to any appreciable improvement in security (Kuo, Romanosky, & Cranor, 2006).

Moving beyond the ubiquitous textual passwords, many researchers have sought to develop graphical passwords also based on secrets. These rely either on recall – the same as almost all text-based passwords – or recognition, usually implemented as a series of prompts that must all be answered correctly to authenticate (Stobert & Biddle, 2013). The series of prompts is necessary to increase the total number of possible graphical passwords for security purposes. More research focus has been placed on recognition-based graphical passwords because the nature of displaying pictures lends itself to recognition tasks.

Graphical passwords based on recognition have been shown to be more usable than those based on recall (Stobert & Biddle, 2013) – incidentally, the same was not found to apply to textual passwords (Wright, Patrick, & Biddle, 2012). However, even using series of prompts, the number of possible passwords in typical implementations remains relatively low. Passfaces, the canonical example of a graphical password (Wright et al., 2012), asks users to select the appropriate face out of nine choices, thrice.

Both types of graphical passwords are especially susceptible to the human tendency of predictability, whether selecting more attractive faces, clicking on hotspots in an image, or drawing familiar shapes (Suo, Zhu, & Owen, 2005). As graphical passwords cannot be disguised without a tremendous hit to usability, they are also particularly vulnerable to attackers who observe the act of authentication – a threat known as shoulder surfing. Graphical passwords tend to increase login time significantly over textual passwords (Stobert & Biddle, 2013), which substantially hinder their

practical, daily use. Finally, there is a relative difficulty in deploying graphical passwords with existing password systems. Due to these issues, graphical passwords have not caught on in commercial systems, other than in contexts such as smartphones and screensavers where they often substituted for previously having no password at all.

Technical Topics in Password Theory

An increasingly popular suggestion made by security professionals is to utilize password managers (Gaw & Felten, 2006), which allow a user to have one master password that unlocks many other passwords for individual websites and applications. Password managers are usually integrated with automatic form fillers so that users need not even sign in to websites manually. Ideally, the use of password managers should result in an increase in both usability and security. In terms of usability, the user is freed from the heavy burden of having to remember dozens of passwords for different accounts. Because the individual passwords can now all be long, complex, randomly generated, and unique – thus bypassing the perils of password reuse across accounts, where the compromise of one account leads to the compromise of many – security is likewise improved (Gaw & Felten, 2006).

Despite the seemingly superior nature of password managers, few users actually employ them. One key reason why this is true is the important issue of trust (Karole, Saxena, & Christin, 2010). Online password managers have proven to be the most usable (Karole et al., 2010), but they are also the least likely to be trusted, due to security concerns. Users don't trust they will always have access to their passwords if

something goes wrong. They worry that their master password being stolen will lead to all of their accounts being compromised; this is true, but no different from the popular single sign-on (SSO) model or losing an e-mail account that can then reset the passwords of all of the other accounts.

For many users, the hassle and uncertainty make setting up a password manager not worth it. Even out of the people who do use password managers, most of them do not use them in conjunction with more complex passwords (Gaw & Felten, 2006), thus deriving only a usability benefit and no security benefit. Regardless of the future adoption of password managers or single sign-on, users would still require one truly secure master password – which they have largely proven unable to create – so security researchers and professionals would not find their jobs made any easier.

When discussing password security, a fundamental question is if online or offline attacks are being considered. Because websites can limit the number of repeated login attempts made by one IP address or for one account, the required level of security for online attacks tends to be much lower.

Almost all of the newsworthy security breaches in the headlines are instead cases where the web server is compromised and the attackers can download and attack the passwords offline. When the passwords are stored on a server simply as cleartext, which is rare but does still happen, all of those passwords are instantly stolen.

The minimal line of defense is hashing the passwords on the server. A hash function is an algorithm that takes a large, variable amount of input and transforms that

into a small, fixed output. For example, if someone wants to check if the large file he or she downloaded is uncorrupted, it would be best to simply check the published hash for that file to see if it matches. Because they were designed for this type of use, the vast majority of hash functions are designed to operate as quickly as possible (while also satisfying other mathematical properties).

Hashing is important to password storage because the hashes can be stored and the passwords themselves discarded (Teat & Peltsverger, 2011). When a user tries to log in using a password, the server simply runs the hash function again on the password and checks if the hash matches. Although an attacker can't work backwards to derive a password from a hash, he or she can make guesses at the password, run the same hash function (which is easily determined), and see if the result matches the stolen hash. Offline, with no lockouts based on the number of failed attempts, the attacker can keep going as long as he or she wishes. As computing power increases, the number of attempts that can be made per second also rises.

To make things even more efficient, an attacker can pre-compute hashes from many inputs, which can be stored and loaded into memory. This is a classic time-memory tradeoff, and a common implementation as it relates to passwords is called a rainbow table (Avoine, Junod, & Oechslin, 2008).

These attacks are defeated by a process called salting, whereby a random (but not necessarily secret) piece of data is stored for each account. The salt is passed to the hash function every time along with the entered password, which results in a different

(but still consistent) hash for that account. This makes it infeasible for rainbow tables to pre-compute all of these new hashes (Teat & Peltsverger, 2011), as common password input now becomes uncommon input.

There is one specific idea in the field of password theory, known as key stretching (Percival, 2009) – hardly ever considered by any of the numerous alarmist articles about massive password leaks – which renders calculations of how long it would take to crack a certain password even more inaccurate. Hash functions are generally designed to work quickly, and in fact most passwords today are hashed on web servers, unsalted, using MD5, an extremely fast hash function from 1992. The reports of billions of passwords being tested per second are calculated under these ideal conditions.

There also exist a set of hash functions that are designed specifically to protect passwords; they take an exceptionally large amount of computation to complete and even allow parameters to be specified that roughly control how long they take through repeated hashes. The most prevalent security-focused hash functions are PBKDF2, bcrypt, and scrypt (Percival, 2009). If a legitimate user must wait an additional 50 milliseconds to be authenticated, this is essentially transparent, but it means an attacker could only make at most 20 attempts per second.

Key stretching is not a panacea for password security. Weak passwords that are easily guessed will still be easy to crack, and impractically strong passwords wouldn't have been cracked anyway, but for the many passwords that fall somewhere in the middle, key stretching is a valuable asset – and the idea will continue to be in the future.

From a practical standpoint, it means that passwords should try to be strong enough to reach that range where they need to be attacked by brute force instead of heuristics. If nothing else, key stretching serves as a reminder that like most aspects of computer security, the battle between users and attackers is like an escalating game of tug of war where both sides adapt – but we certainly aren't doomed to lose.

Mnemonics for Passwords

The fundamental issue with all types of user-generated passwords is that humans are not random creatures. We behave in patterns, and we remember information more readily when some sort of order is imposed. This very fact makes that information easier to guess. Of course, the disorder of randomly generated passwords makes them hard for people to use. But instead of starting with a nonrandom phrase and working backwards to form a nonrandom password, we can instead begin with a random password and then generate a mnemonic phrase to better remember it. For instance, rather than deriving *jajwuth* from the well-known *Jack and Jill went up the hill*, which results in a weaker password, the randomly generated *jpwjaop* can instead be converted to *Jill's pet wolf just ate our pizzas*.

This mnemonic phrase can be created by either the user (Scarfone & Souppaya, 2009) or the system. Taking a random password and converting it to a phrase can prove quite difficult and time-consuming for users, suggesting a potential advantage for system-generated mnemonic phrases (Jeyaraman & Topkara, 2005).

Previous attempts at generating mnemonic phrases automatically, such as a fairly complex system by Jeyaraman and Topkara (2005), have focused on linguistic techniques such as natural language processing and personalized corpuses. However, the produced mnemonics, such as *Basotho orthoper, shoots dais, toes Yakut hack* from *bosdtyh*, have not been found to be usable or memorable.

Since the translation from randomly generated password to mnemonic phrase is only to aid the user's memory, and the phrase itself is not used in authentication, there is no change in security. Therefore, a simple solution imposing maximum order (e.g., converting *bosdtyh* to *Barbara's other squirrel definitely took your hotcakes*) should work best. The effectiveness of the system could be further enhanced by utilizing a mnemonic picture, which Nelson and Vu (2010) showed to improve the memorability of user-generated passwords.

Goverover, Basso, Wood, Chiaravalloti, and DeLuca (2011) found that combining multiple recall strategies often improves memorability by more than any single strategy can alone. In the context of password recall, we can utilize multiple mnemonic strategies simultaneously, including cues, the generation effect, and the self-reference effect.

Cues have been shown to enhance the recall of items with which they are associated (Slamecka & Graf, 1978). In this context, we can employ user-created pictures to help cue the recall of phrases to bolster password recall. Cues that are more strongly associated with the target items to remember and hold large amounts of

information tend to be the most effective (Greenwald & Banaji, 1989). Hence, pictures that contain more information should more effectively cue password recall than pictures that hold less information do. However, if the pictures contain too much information about the authentication secret or are too strongly associated with the phrase (Mäntylä, 1986), attackers will have an easier time decoding the pictures.

But pictures need not contain large amounts of generalizable information in order to be strongly associated with a target item for a given person. The generation effect demonstrates that creating cues oneself rather than simply combining pieces of information together can additionally increase recall (Craik & Tulving, 1975). Because the generation effect works only for the person who created the artifact, users who design their own pictures should benefit from improved password recall (Bertsch, Pesta, Wiscott, & McDaniel, 2007), yet the chances of an attacker finding the password should not increase.

More specifically, creating pictures through the process of drawing may increase the benefits of the generation effect over simply using online images. With drawings, users generate the entire picture rather than merely linking images together in the process of compilation. Furthermore, drawing images may incite the self-reference effect: the enhancement of recall when referring to the self (Rogers, Kuiper, & Kirker, 1977). This is because the very nature of drawing an image may often be more personal than combining found images.

Input Masking

Even if the underlying authentication systems have been designed and implemented properly, the very act of authentication itself can also be a soft spot for unauthorized users to attack. One such method, known as shoulder surfing, entails looking at a user's screen or keyboard as he or she enters critical electronic input such as a password or credit card number. Different techniques are employed to hide sensitive input, such as the common practice of replacing entered text with bullets. Such techniques increase security but inevitably degrade the usability of the process by making it harder for the user to see what he or she is entering.

Even bullet-masked input may not be enough to stop determined attackers. Although looking at the screen is easier whenever possible, shoulder surfers can still look at a user's keyboard. In fact, they can also employ hidden cameras, keyloggers, or other devices to capture what is being typed without the need to be physically present (Wiedenbeck, Waters, Sobrado, & Birget, 2006).

The response by many researchers to this situation has been to design systems with increased resistance to shoulder surfing but a marked decrease in usability. Complicated schemes have been implemented based on ideas such as convex hull clicks (Wiedenbeck et al., 2006), probabilistic cognitive trapdoor games (Roth, Richter, & Freidinger, 2004), and cued gaze points (Forget, Chiasson, & Biddle, 2010).

It has been argued that bullet masking is not secure enough to warrant the cost to usability. Influential usability consultant Jakob Nielsen started a controversy when he

recommended on his website that most passwords should be displayed as cleartext instead of bullet-masked due to usability concerns (2009). While the usability of bullet-masked input could be improved, there are obvious security concerns with simply showing passwords as cleartext.

Because of the low level of planning, equipment, or technological sophistication required of the attacker (Hoanca & Mock, 2005), shoulder surfing is widely available as a form of attack. This greatly increases the pool of would-be assailants beyond merely the domain of dedicated experts.

Passphrases

The idea of a passphrase can be seen as an extension to the traditional passwords used in digital authentication. They fulfill the same role as a secret that the user demonstrates to the system that he or she knows in order to prove personal identity. Therefore, like passwords but unlike other proposed replacements for passwords such as tokens or biometrics, passphrases should also prove to be easy to implement, easy for users to understand, and easy for users to recover from lockout.

The fundamental feature of passphrases is that they are longer than passwords. The primary motivation for this change is to increase the level of security, particularly against brute-force attacks, which occur when an automated attack is conducted not by making informed guesses but by exhaustively trying all the possible permutations in a given password space (Herley & van Oorschot, 2012). As the price for large quantities of distributed computing power becomes increasingly affordable, making brute-force

attacks more viable, there is renewed interest in moving from shorter passwords to longer passphrases – not to use as mnemonics but as the actual secret texts themselves (Burr, Dodson, & Polk, 2006).

From a human factors standpoint, the key benefit of passphrases is that they allow the secret to be much less complex, and thus more memorable, while still maintaining or increasing security. Using a dataset of 12,000 actual passwords, Kelley et al. found that a password policy of "anything at least 16 characters long" was much stronger than any policy requiring at least 8 characters plus all the possible restrictions (e.g., character types, repeated characters, dictionary checks) commonly found today (2011). Likewise, Komanduri et al. found that the most effective combination of security and usability resulted from long passwords with no other restrictions (2011).

In mathematical terms, by increasing the number of characters, the total number of bits of entropy – which can be thought of as the randomness, disorder, or security of a password – can be maintained while greatly lowering the number of bits of entropy per character. Randomly selected characters would produce the greatest amount of entropy (as each character is independent), while English-language phrases would exhibit far less (as the letter *q* would almost always be followed by *u*).

According to guidelines from the National Institute of Standards and Technology (NIST) (Burr et al., 2006), the first character of an English-language phrase encapsulates approximately 4 bits of entropy, the next seven characters represent 2 additional bits of

entropy each, the next twelve characters represent 1.5 bits of entropy each, and every subsequent character adds 1 additional bit to the calculated entropy.

To illustrate with a previous example, "*Jill's pet wolf just ate our pizzas*" ($35\ total\ characters = 1\ character \times 4\ bits + 7\ characters \times 2\ bits + 12\ characters \times 1.5\ bits + 15\ characters \times 1\ bit = 51\ bits\ of\ entropy$) would be stronger than even a randomly generated 7-character password using uppercase and lowercase letters, numbers, and symbols ($7\ characters \times \log_2 95\ possible\ characters = 46\ bits\ of\ entropy$). When taking into account that user-created passwords exhibit far less security than system-generated ones, the difference drastically increases.

The first known mention of the idea of using a longer phrase in lieu of a traditional password comes from a paper by Sigmund Porter (1982). Just three years later, Kurzban created a system-generated passphrase system using words randomly chosen from a set of wordlists (1985). The resulting passphrases needed to be abbreviated since authentication systems of the day typically did not support passwords with double-digit length.

This technical issue was a major reason why the use of passphrases did not catch on in the following years. But with the necessity of higher levels of security in recent years due to the vast increase in widely available computing power, as well as the steady obsolescence of legacy authentication systems (Keith, Shao, & Steinbart, 2007), interest in passphrases is the highest it has ever been.

Keith, Shao, and Steinbart tested the memorability of user-created passphrases and found that while the memorability of the passphrases themselves was no better or worse than that of passwords, login success rate was significantly lower because of a significant increase in typographical errors (2007). This finding suggests that combining passphrases with techniques that reduce errors made while entering passwords, as discussed in Chapter 5, could mitigate this effect.

Around the same time, Kuo, Romanosky, and Cranor evaluated the security of user-created passphrases and found that the security benefit of this group of passphrases was lower than the security community expected (2006). This was because users frequently selected weak phrases based on topics such as lyrics, movies, literature, or television shows. This research recommended the possibility of using system-generated passphrases, such as an extension of the system-generated phrases discussed in Chapters 3 and 4, instead of user-created ones.

One of the most popular recommendations in the online security community for creating system-generated passphrases is Arnold Reinhold's Diceware (2012), first demonstrated in 1995. This system utilizes a wordlist containing 7,776 words – not all of which can be found in an English dictionary – and suggests that the user rolls five physical dice to determine each word in the passphrase; Reinhold recommends using five words in each passphrase (2012) for future-proof security. Observations among security professionals indicated that Diceware did not see much use in practice during

the years following its creation, and little research was initially conducted about the usability of the system.

As a result, Leonhard and Venkatakrishnan (2007) tested the usability of Diceware passphrases containing three words each. They found that Diceware performed poorly, with users rating their satisfaction with their password at 1.71 out of 5. Because the wordlist was so large, passphrases such as *lares ave ghent* were possible, suggesting that perhaps using a smaller wordlist would produce more memorable passphrases.

Despite any usability issues Diceware may have, as well as its failure to gain traction with typical users, Diceware has started to see some implementation in security-conscious circles. Diceware has been used since 2012 by the leading privacy-focused search engine DuckDuckGo to store user settings, and it is recommended by the password manager 1Password to create the master password (Shiner, 2013). The recent use of Diceware in the wild mirrors the general need to switch to passphrases from passwords for increased security.

To examine the effect of different wordlists on randomly generated passphrases, Shay et al. conducted a study using 1,476 online participants and a variety of system-generated passphrase conditions, including passphrases with wordlists of size 181, 401, and 1024 consisting of three and four words (2012). There was found to be little difference between wordlists of these sizes. Memorability was affected more by the number of characters in the passphrase rather than the number of words, reflecting the

typographical errors found by Keith, Shao, and Steinbart. Interestingly, the Diceware wordlist prioritizes short words (Reinhold, 2012).

Integration of Techniques

The system-generated mnemonic phrases of Chapters 3 and 4 work directly with existing password systems because the passwords themselves, not the mnemonics, are stored. Nevertheless, as the majority of systems these days support longer passphrases, the work in Chapters 3 and 4 could be extended to generate passphrases instead of passwords. In this scenario, the user would type *Jill's pet wolf just ate our pizzas* instead of using the phrase to remember *jpwjaop*. Because the individual words chosen would now impact security, this change would necessitate greatly expanding the size of the wordlist while attempting not to erode the usability benefit resulting from the specialized wordlist. In essence, the generated passphrases should still involve simple words and maintain a simple structure.

Shay et al. recommended that perhaps the most promising area for improving memorability of passphrases lies in asking users to construct a scene or story relating to their passphrase (2012). The work discussed in Chapters 3 and 4 found that asking users to create a mnemonic picture to assist them with system-generated passphrases did improve the level of usability of those passphrases.

The new input masking technique covered in Chapter 5, which also works easily with current password systems and policies already in place, could be extended to handle passphrases and other forms of readable text. When rotated by character,

readable input text tends to stand out amidst a sea of unreadable decoy text. For instance, *Jill's pet wolf just ate our pizzas* might be surrounded by *Utww!d ape hzwq ufde lep zfc atkkld* or *Hgjj/q ncr umjd hsqr yrc msp ngxxyq*, which would make it easy for attackers to find the genuine passphrase.

By incorporating a dictionary component into the system, readable text could be masked with other readable decoy text. Since typing errors are expected to increase as text string length grows and as typing speed rises due to the use of plain language passphrases, the proposed system might enjoy additional benefits when compared to other input masking techniques.

The hope is that by leveraging both the natural tendency of passphrases to imply a mental scene, as well as the reduced typographical errors resulting from usable input masking, passphrases can overcome their current limitations and see wider adoption.

CHAPTER 3: MNEMONICS TO AID SYSTEM-GENERATED PASSWORD RETENTION

Introduction

This chapter is based on previously published work (Juang, Ranganayakulu, & Greenstein, 2012).

The impetus for this study is the idea that user-generated passwords have been shown to be insecure, while system-generated passwords have been shown to be difficult to use. The majority of prior work has attempted to increase the security of user-generated passwords, while this study focuses on increasing the usability of system-generated passwords.

The first innovation of this study is the implementation of a specialized wordlist and template to generate consistently simple mnemonic sentences to help users remember their passwords, with no resulting change in security. Previous research in password mnemonics resulted in complex phrases that users did not like.

The second innovation is prompting the user to generate a mnemonic picture during account creation that is shown during the login process. This transforms the login task from one of uncued recall to more memorable cued recall.

We aim to investigate the effect that these contributions have on the usability and security of system-generated passwords. We hypothesize that they increase usability but that the mnemonic picture slightly decreases security of the system.

Implementation

The basis of our mnemonic phrase system relies on the realization that we can use a tiny specialized wordlist to construct our mnemonics from any given password. Since the mnemonic itself has no bearing on security, but functions only as an aid to the user, even a single sentence template with a direct mapping of letters to words is acceptable. We can thus design our exact wordlist and character positions so that all possible combinations make meaningful sense and only use simple grammar and vocabulary, as seen in Figure 1.

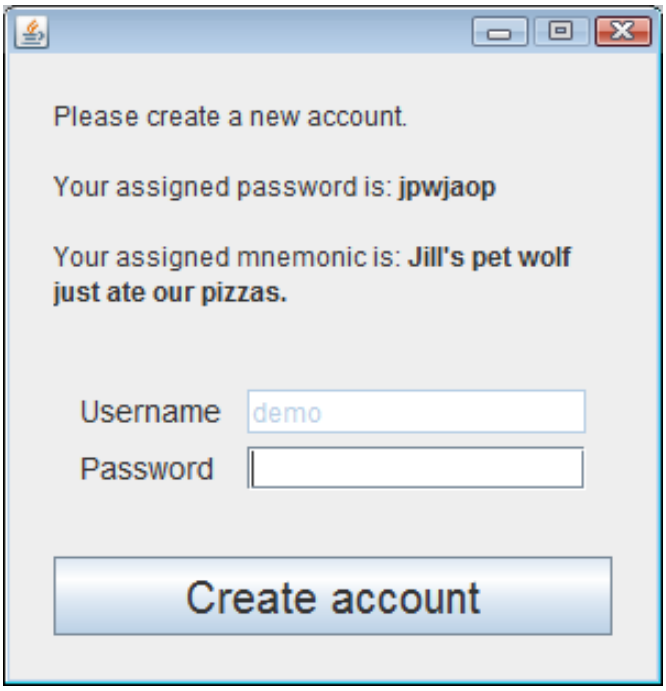

Figure 3.1. The system-generated mnemonic is designed to be simple.

We were able to come up with an appropriate word for each possible letter. However, some positions of particular letters (such as q, x, and z) resulted in words that could be considered difficult, especially to some of the international students in our pilot study. In addition, certain words that were easy for one person sometimes proved difficult for another, due to background or culture differences.

Because of these issues, we decided to screen the entire wordlist with people from different cultures. After being identified as difficult by more than one person, words and the letters in positions that would lead to those words were removed for the purposes of our study. Our system is robust enough that minor deletions or additions to the wordlist hardly affect its security, but to be fair, we screened out the same positions of letters from all conditions in our study.

A visual aid during login is the next major component of our system. Upon presenting the user with an automatically generated mnemonic during account creation, a simple paint program and a browser window are also opened by the system. By default, the browser window displays automatically generated search results of images related to the mnemonic.

The user's task is to draw a personal visual reminder, of any size, in the paint program. For convenience, images found in any image search or elsewhere on the Web can be copied and pasted directly into the paint program. The user can copy as few or as many images as he or she wishes. As illustrated in Figure 2, the picture is typically decipherable only by its creator.

Since our system opens the appropriate file in the paint program and later automatically saves it, the user does not need to specify or upload the finished file. This picture is then displayed back to the user on the login screen when he or she attempts to sign in later.

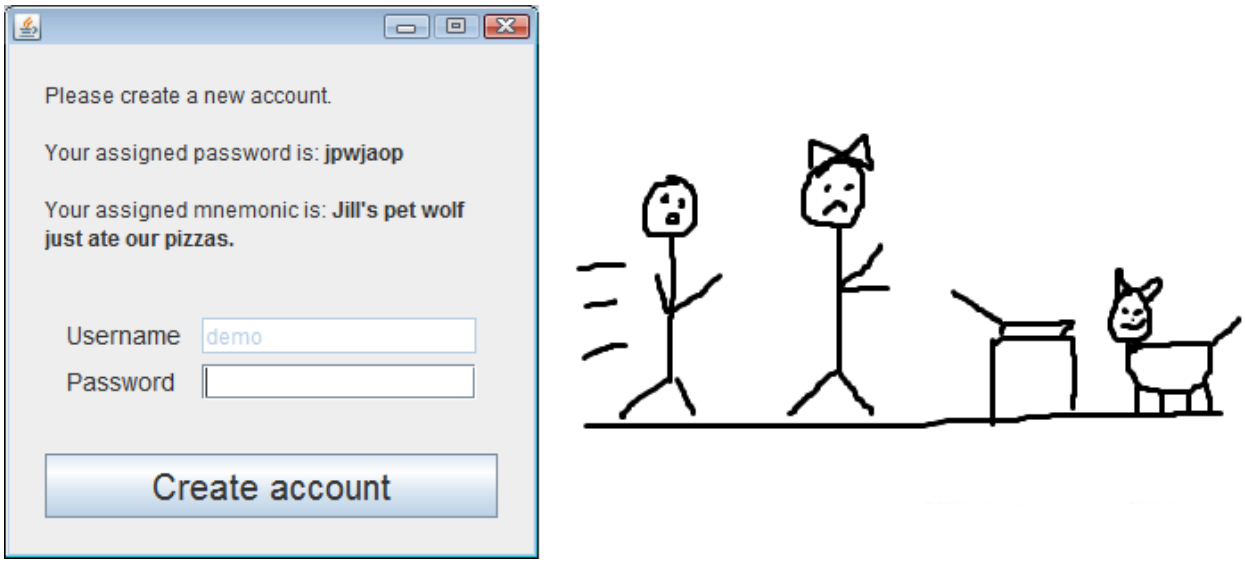

Figure 3.2. The mnemonic phrase and picture complement each other as simple, effective memory aids.

Method

Threat Models and Limitations

For our study, an adversary is assumed to have complete knowledge of our implementation, including the wordlist. We wanted to test the ability of users to remember passwords for multiple accounts simultaneously, as prior research has focused on single accounts. Naturally, this greatly increases task difficulty and consequently lowers memorability.

Although we have followed the NIST level 2 guidelines for authentication security (Burr et al., 2006) of at least 30 bits of entropy $(7\ characters \times \log_2 26\ possible\ characters = 33\ bits\ of\ entropy)$, distributed computing power has been greatly expanding. Should the current system-generated passwords become too weak to withstand brute-force attacks, it is trivial to modify our system to generate longer passphrases instead of mnemonics for passwords. On the other hand, to protect against adversaries who know our specific system, the wordlist would be expanded but could still retain simple words as best as possible. Additional sentence templates would also be added.

In our study, we did not force participants to take a specified amount of time to create their account. While the effect of condition on memorability could be mediated by creation time, if the system-generated mnemonic policy naturally encourages the user to think more deeply about the mnemonic, we feel this indirect effect is every bit as important as the direct effect of condition on memorability.

The use of pictures in our system leaks a small amount of information out to potential adversaries. We did not prohibit participants from creating laughably insecure "pictures" such as the original password in text form. Perhaps because they did not mind the task, or they understood the drawbacks, none of our participants created such weak pictures.

After our initial study was completed, we decided to quantitatively test the security of our system. We asked participants in a follow-up study to attempt to guess

previously created passwords. All of these participants created three passwords using our system and were provided with general and system-specific password guessing tips, as well as full knowledge of our implementation including the complete wordlist used.

Subjects

We recruited 54 college students (34 males, 20 females), ranging from age 19 to 35, via word of mouth and promotional fliers posted in several locations on campus. Each participant was compensated $8 for taking part in our study. All participants had over five years of computer experience and normal or corrected-to-normal eyesight.

Experimental Design

The study utilized a between-subject design. Participants were assigned to conditions based on a balanced Latin square. Participants created three passwords using one of the mnemonic policies investigated:

1. None: the user is not instructed to utilize any mnemonic aid
2. User-generated: the user creates his or her own mnemonic phrase based on guidelines from the National Institute of Standards and Technology (Scarfone & Souppaya, 2009)
3. System-generated: the user receives a randomly created mnemonic phrase (as specified in the earlier Implementation section) and then creates a mnemonic picture depicting the phrase

Procedure

Each session started with a brief overview of the study, after which participants signed an informed consent form and completed a short pre-test demographic questionnaire.

Regardless of condition, each user received three system-generated passwords in sequence, one at a time. For each password, the participants were asked to memorize the password and then enter it when ready to create an account. Participants were instructed to pay close attention to which passwords corresponded to which accounts.

Depending on the condition, part of the memorization process might have involved creating or reinforcing a mnemonic. Upon successful entry of the given password, the process was repeated until all three accounts were created.

Participants then completed the NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) and the System Usability Scale (SUS) (Brooke, 1996). These tests measured the workload and usability of the account creation task.

After a five-minute distraction task of solving two-digit mental arithmetic problems, participants moved to the authentication phase. One of the three accounts was randomly selected and users were prompted to sign in using the appropriate password. Participants were given a maximum of five attempts to sign in successfully before a failure was recorded instead. The process was then repeated until the login task was completed for all three accounts.

Finally, participants completed another NASA-TLX and SUS. This time, these tests measured the workload and usability of the password recall task. The total time required to complete the experiment was approximately thirty minutes.

Approximately a week after each participant's first session, he or she was invited back for a second session to repeat the recall tasks. Participants were specifically instructed not to write down or rehearse their passwords in the meantime.

Subsequently, twenty-four participants were tasked with attempting to guess the passwords of five accounts: three of the system-generated mnemonic condition and two of a no-picture control condition, representing the other conditions. The system treated these adversaries in exactly the same manner as ordinary users, including the five attempts allowed.

Dependent Variables

All measures, both objective and subjective, were recorded by the system:

- Creation time: seconds taken to create an account, measured from the presentation of the system-generated password to the successful account creation
- Recall time: seconds taken to either succeed or eventually fail to sign in to an account, measured starting from the presentation of the login screen
- Damerau-Levenshtein distance (Damerau, 1964) for both effectiveness and security: the length in edits (total sum of single character insertions, deletions, substitutions, and adjacent transpositions needed to transform

one string to another) between the participant's recollection or guess of a password and the actual stored password

- Jaro-Winkler proximity (Winkler, 1990) for both effectiveness and security: the similarity or correlation between the participant's recollection or guess of a password and the actual stored password, normalized such that 0 indicates no similarity and 1 indicates equality
- Success: a binary metric of either eventual success or failure to sign in
- Creation and recall NASA-TLX indices addressing mental demand, physical demand, temporal demand, performance, effort, and frustration, each on a 7-point Likert scale
- Creation and recall SUS scores addressing usability, each out of a total of 100 possible points

Results

We used SPSS 19 for the data analysis. For the majority of our dependent variables, we used a one-way ANOVA with a 95% confidence interval to determine significance. Welch's correction was used whenever variances were heterogeneous. Given significant results, Tukey's HSD test showed which mnemonic policies differed significantly from one another. The Games-Howell test was used in place of Tukey's HSD whenever variances were heterogeneous.

Since the creation time and recall time did not follow a normal distribution, a log transformation was first applied to each of these to achieve normality. The same type of

ANOVA as before was then conducted. The security measures used t-tests. Success was analyzed using a binary logistic regression.

Objective Measures

Efficiency. The results indicate a significant difference for creation time ($F=326$, $p<.001$) but not for first-session recall time ($F=2.41$, $p=.093$) or second-session recall time ($F=0.476$, $p=.622$). The creation time in seconds was shortest for the mnemonic policy of none ($M=32.0$), next shortest for user-generated mnemonic ($M=107$), and longest for system-generated mnemonic ($M=338$). The mean first-session recall times in seconds were 46.6 for none, 78.8 for user-generated mnemonic, and 60.0 for system-generated mnemonic. For the second session, the mean recall times were 70.8, 94.9, and 69.4 seconds, respectively.

Effectiveness. There were statistically significant differences for first-session ($F=22.3$, $p<.001$) and second-session Damerau-Levenshtein distance ($F=17.3$, $p<.001$), and, as seen in Figure 3, first-session ($F=17.7$, $p<.001$) and second-session Jaro-Winkler proximity ($F=14.3$, $p<.001$). The Damerau-Levenshtein distance (first-session; second-session) was shortest for system-generated ($M=0.150$; $M=0.930$), next shortest for user-generated ($M=1.35$; $M=3.15$), and longest for none ($M=2.54$, $M=3.43$). The Jaro-Winkler proximity (first-session; second-session) was highest for system-generated ($M=.987$; $M=.928$), next highest for user-generated ($M=.892$; $M=.750$), and lowest for none ($M=.768$; $M=.726$). For both measures, there were no significant differences between the second session of user-generated and none.

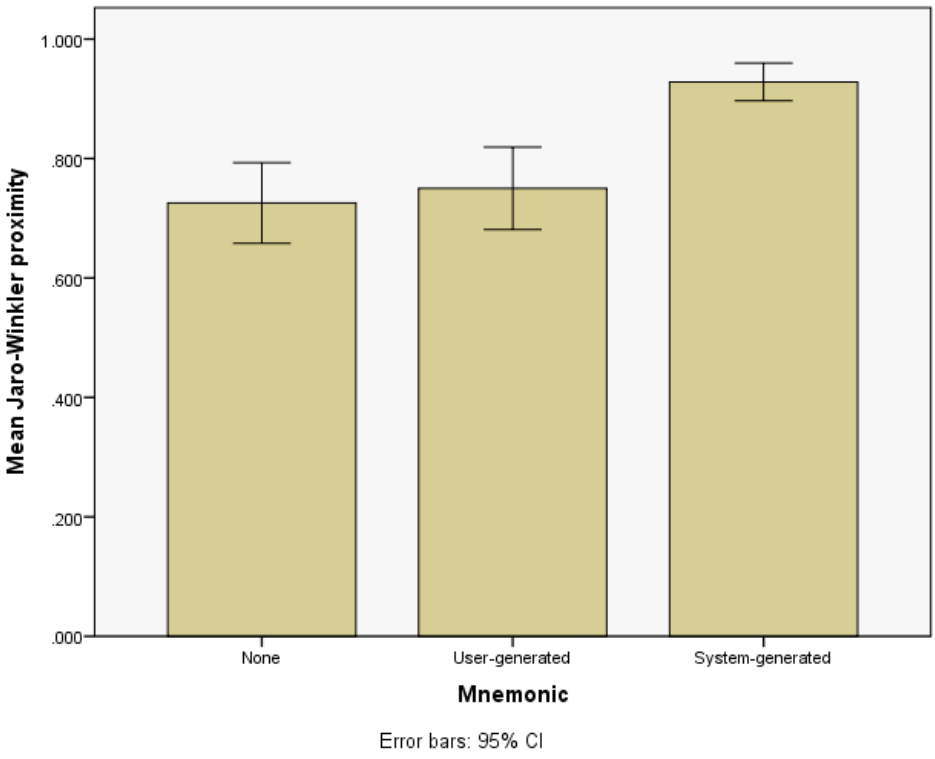

Figure 3.3. 2nd-session Jaro-Winkler proximity (larger values indicate increased effectiveness)

The effect of mnemonic policy on success was also determined to be statistically significant for both the first session (χ^2=29.8, p<.001) and second session (χ^2=13.8, p=.001). The first-session success rate was 87.0% for system-generated, 53.7% for user-generated, and 38.9% for none. As seen in Figure 4, the second-session success rate was 59.3% for system-generated, 33.3% for user-generated, and 25.9% for none. These represent drop-offs of 31.8%, 38.0%, and 33.4%, respectively, between first and second sessions.

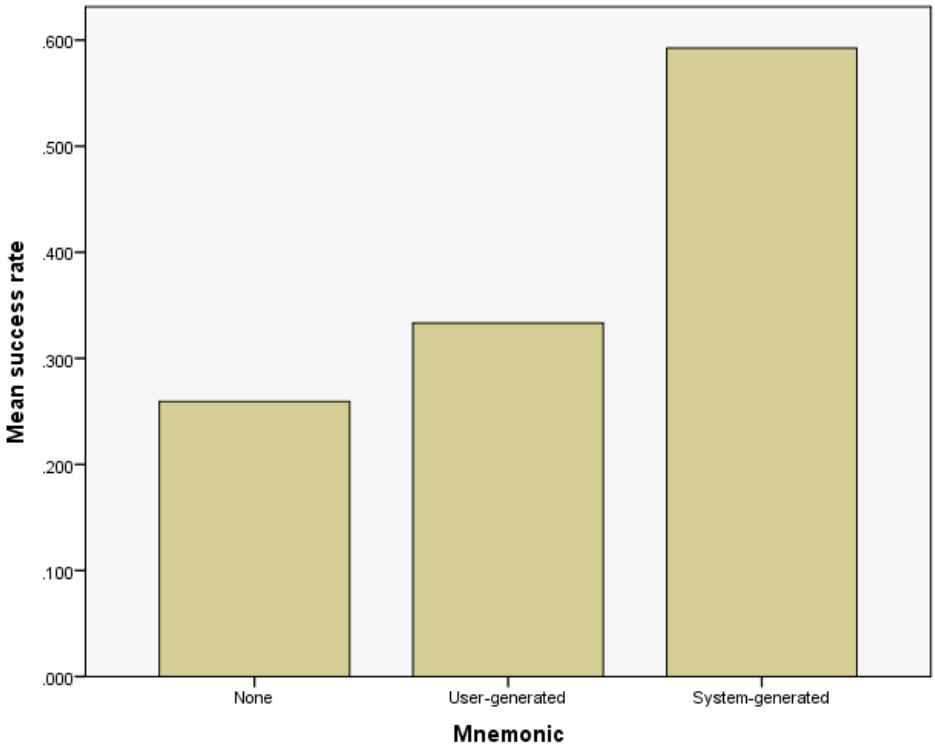

Figure 3.4. 2nd-session success rate (larger values indicate increased effectiveness)

Security. There was a statistically significant difference for the Damerau-Levenshtein distance between the adversaries' guesses and the actual passwords (F=8.31, p=.005). The Damerau-Levenshtein distance was longer for the no-picture condition (M=6.40) than the system-generated mnemonic condition (M=5.67), as shown in Figure 5. There was no statistically significant difference for the Jaro-Winkler proximity between the adversaries' guesses and the actual passwords (F=0.999, p=.320). The mean Jaro-Winkler proximity was .428 for the no-picture condition and .499 for the system-generated mnemonic condition. These values can be seen in Figure 6. No password was successfully guessed.

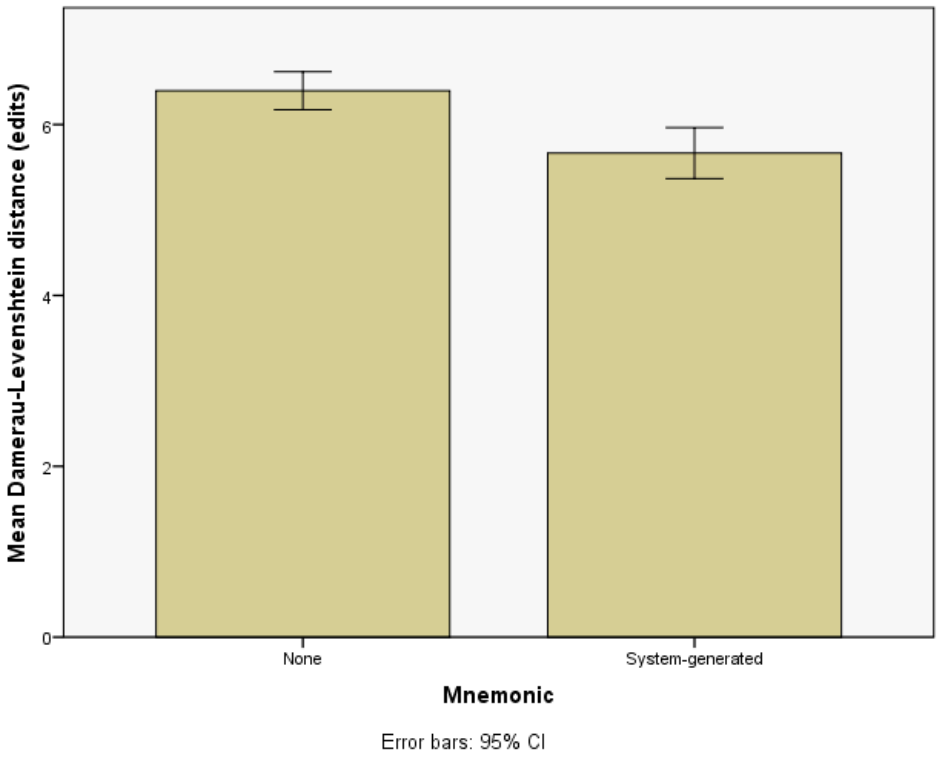

Figure 3.5. Adversarial Damerau-Levenshtein distance (larger values indicate increased security)

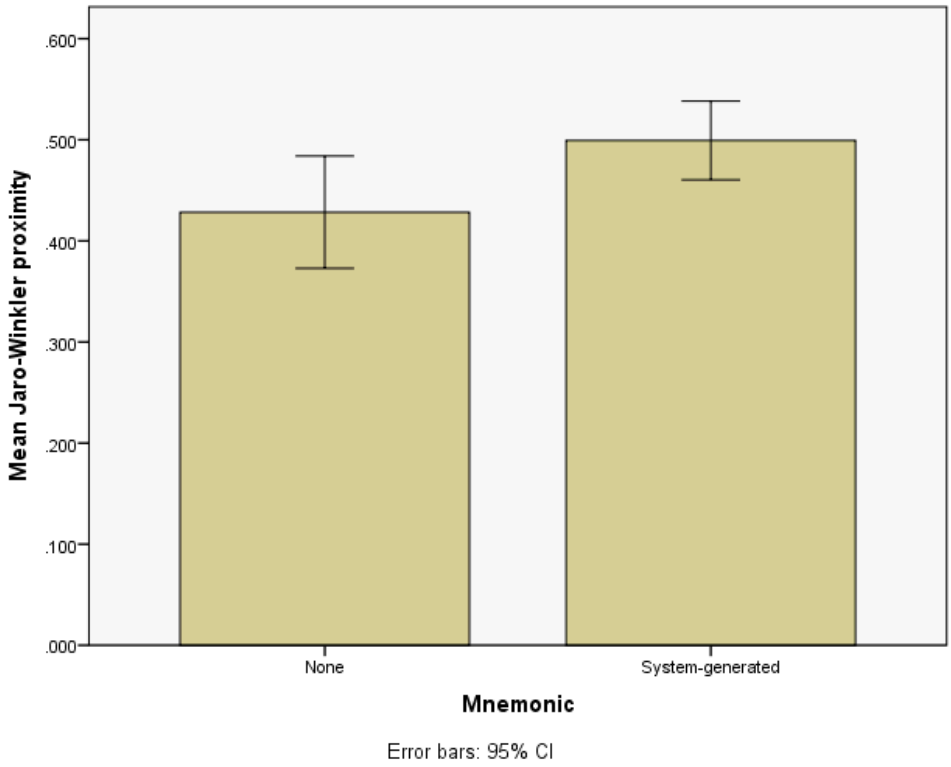

Figure 3.6. Adversarial Jaro-Winkler proximity (smaller values indicate increased security)

Subjective Measures

Workload. The only workload metric to achieve significance in the creation task was mental demand (F=3.81, p=.029). The mental demand was significantly lower for system-generated (M=4.94) than none (M=6.00). For first-session recall workload, mental demand (F=5.02, p=.010) and effort (F=4.14, p=.022) proved statistically significant. The mental demand of system-generated (M=4.00) was significantly lower than user-generated (M=5.56) and none (M=5.78). System-generated (M=3.44) participants rated their effort in the first-session recall task as significantly lower than

did participants who did not use a mnemonic (M=4.94). There were no significant differences in workload for the second session.

Usability. In the creation (F=9.53, p<.001), first-session recall (F=7.05, p=.002), and second-session recall (F=7.66, p=.001) tasks, differences in the SUS measure were statistically significant. The usability score (creation; first-session recall; second-session recall) for system-generated (M=66.0; M=68.3; M=66.7) was significantly higher than user-generated (except for second-session) (M=50.6; M=46.8; M=51.2) and none (M=41.1; M=43.2; M=37.8). The usability scores for second-session recall can be seen in Figure 7.

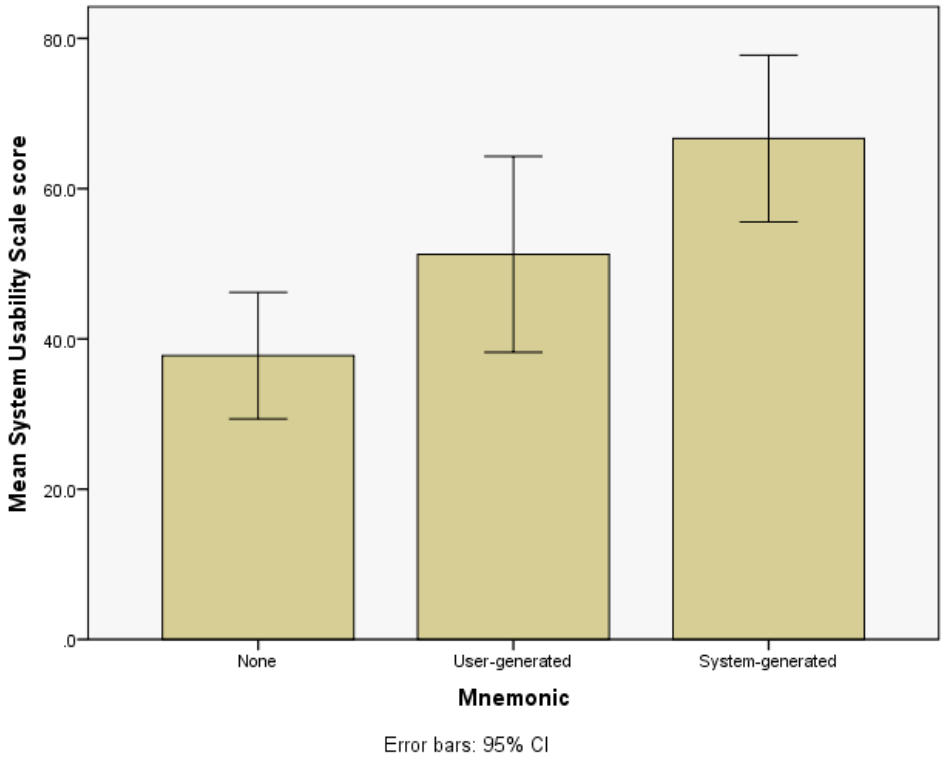

Figure 3.7. 2nd-session SUS score (larger values indicate increased usability)

Discussion

As a whole, the results indicate a promising outlook for the use of mnemonics to help users remember system-generated passwords. That outlook is particularly hopeful for our implementation of system-generated mnemonics.

Although the creation time for the system-generated mnemonic condition was much longer than the other conditions, account creation is a one-time event where successfully encoding the password into memory is much more important than the speed of the process. A forgotten password could potentially cause a far greater loss of time down the road than any initial time investment.

In contrast to creation time, any differences in recall time would lead to a cumulative effect over repeated logins. Therefore, it is encouraging that differences in recall time were not statistically significant between any of the conditions, although system-generated mnemonics and the absence of a mnemonic tended to be faster than user-generated mnemonics.

Regarding security, the Damerau-Levenshtein distance is useful primarily as a measure of strength against brute-force attacks. While the security of our system as measured this way was significantly lower than the control condition, the mean difference was less than three-fourths of a character. On the other hand, there was no significant difference in Jaro-Winkler proximity, which is useful primarily as an indication of strength against human-based attacks.

When participants created pictures, the weakest parts of the mnemonic phrases tended to be the nouns. This suggests that the possible addition of one non-noun to each phrase could cover the Damerau-Levenshtein difference, hopefully without degrading usability. It is worth emphasizing that when compared to the status quo of user-generated passwords, all of the conditions tested would be far more secure.

The results for recall Damerau-Levenshtein distance and Jaro-Winkler proximity show that whether we evaluate only exact matches or take near misses into account, there are clear divisions in memorability. System-generated mnemonics consistently performed better than the other two conditions.

By looking at the success rates for the three conditions, we see that a user utilizing a system-generated mnemonic is 1.78 times likelier to succeed in recalling his or her password than someone using a user-generated mnemonic and 2.29 times likelier to succeed than someone using no mnemonic.

The SUS scores indicate that users generally considered the system-generated mnemonic condition to be the most usable. The NASA-TLX results show that participants using system-generated mnemonics were not subjected to increased workload levels. If anything, their ratings tended to be more favorable compared to ratings for the other conditions, even during the account creation task.

By striving to outperform the security of user-created passwords and improve on the usability of randomly assigned passwords, we developed a system-generated mnemonic implementation using a tiny specialized wordlist along with picture creation

and display. We demonstrated that users with system-generated passwords can benefit from increased memorability and usability by adopting our system.

CHAPTER 4: METHODS OF MNEMONIC CREATION FOR AUTHENTICATION

Introduction

This chapter is based on previously published work (Fraune, Juang, Greenstein, Chalil Madathil, & Koikkara, 2013).

The study in Chapter 3 determined that mnemonic phrases and images increased the usability of system-generated passwords. However, security was slightly decreased. In that study, we allowed participants to choose between drawing a picture, arranging online images into a collage, or combining both drawings and online images. In addition, there was no condition that tested system-generated mnemonic phrases with no picture aid.

This study investigates the effect of different types of picture generation. All conditions utilize system-generated passwords as well as system-generated mnemonics. Our goal is to identify specifically which aspects of picture generation, if any, exhibit the largest effects on the previously demonstrated usability improvements. If a particular method proves most effective, we can use this method exclusively in future work. On the other hand, if there are hardly any differences between picture methods (excluding the no picture condition), we may still want to standardize the procedure for consistency. If we find no differences between *any* of the conditions (including the no picture condition), this would suggest that the previous usability improvements were entirely due to the system-generated mnemonic phrases.

Based on the generation and self-reference effects, we hypothesize that the drawing condition might be the most memorable, followed by the combined condition, then the online image condition, then a steep drop-off to the no picture condition. This mirrors the level of involvement required of the user in picture creation.

Likewise, because user-drawn pictures should provide less shared information than found images located by an online search engine, we hypothesize the drawing condition to be the most secure of the picture conditions, followed by the combined condition, and then the online image condition. The no picture condition would exhibit the greatest level of security since no picture cues are provided to potential attackers.

Method

Experimental Design

The study used a within-subjects design to test the effectiveness of different types of user-made pictures as mnemonic devices to cue the recall of phrases that in turn represent computer-generated passwords. Another within-subjects design was employed to evaluate the success of using pictures to attempt to crack users' passwords.

Participants

Twenty-nine participants were recruited from Clemson University by word of mouth and e-mail for a convenience sample. Recruits who failed to pass a test for color blindness were screened out, and 24 participants (14 males, 10 females), ranging from

age 21 to 38, were ultimately included in the study. No attrition occurred between sessions.

Procedure

In both sessions, participants worked at a desktop computer, and a researcher explained the purpose of the study. After signing the informed consent form, participants completed a demographic questionnaire and a test for color blindness (Ishihara, 1987). They then received training for the password creation system, covering the following points:

1. For each password, the system generates a random string of seven lower-case Latin letters (e.g., “blmiccs”)
2. For each password, the system generates an English phrase such that the first letter of each word corresponds to a letter in the password (e.g., the password “blmiccs” corresponds to the phrase, “Barbara’s little mouse intently chewed countless sandwiches”)
3. For each phrase, the user is asked to create a picture in accordance with further instructions
4. Tips for creating secure images are provided (e.g., do not include words from the phrase in the picture)

After training, the computer presented four successive account login screens along with four unique corresponding passwords and phrases that they were asked to memorize in relation to the accounts. For each phrase, participants were prompted to

create a digital picture using one of the following methods. Conditions were presented in a counterbalanced order across subjects to minimize order effects:

1. NP – participants made no picture
2. D – participants drew a picture using a paint application
3. OI – participants used online images to create a picture or collage
4. DOIC – participants combined drawings and online images in a paint application

After creating each account, participants completed the NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) and the System Usability Scale (SUS) (Brooke, 1996) to assess the workload and usability of the system. Participants created all four accounts, and then completed a five-minute mental arithmetic distractor task. For the second part of Session 1, participants were prompted to type their passwords into the matching accounts in a randomized order. For all conditions except the No Picture (NP) control, participants were shown the pictures they had previously created. Five unsuccessful login attempts were deemed an overall failure for the particular account, and participants proceeded to the next task. Again, participants completed the NASA-TLX and SUS after each condition. Session 1 lasted approximately forty-five minutes.

Session 2 occurred roughly one week after initial participation. Participants returned to the study location and repeated the second part of Session 1. Participants who did not remember their passwords during Session 1 were still invited back to try again in Session 2. They were not reminded of their passwords after Session 1 failure.

Participants then simulated attackers trying to decode others' passwords. Participants were given general and system-specific training on how to crack passwords from pictures, and participants were also provided the list of 182 words from which the computer generated the phrases to match the passwords in order to simulate an attacker that might have used the system until discovering all or most of the words used. Participants attacked two accounts of each of the four different conditions, all for different users. For each account, participants were asked to make at least one attempt and allowed up to five attempts to login. Session 2 lasted about thirty minutes.

Measures

The computer recorded the time users took to memorize passwords, including the time to create the picture. Then, for every login attempt, of "user" and "attacker," the system recorded the following:

- Time taken to correctly enter each password
- Success: a successful or failed login attempt
- Damerau-Levenshtein distance (Damerau, 1964): the length in edits (total sum of single character insertions, deletions, substitutions, and adjacent transpositions needed to transform one string to another) between the participant's typed password and the actual password
- Jaro-Winkler proximity (Winkler, 1990): the similarity between the participant's typed password and the actual password, normalized from 0 to 1

- NASA-TLX scores to measure workload of creating the picture and memorizing the password, and of the two login processes
- SUS scores to measure the usability of creating the picture and memorizing the password, and of the two login processes

Results

Data were analyzed using SPSS 19. For most data, repeated measures within-subjects 4 (condition) x 2 (session) ANOVAs were run. For login success, a binary logistic regression was performed. If Mauchly's test of sphericity was insignificant at the .05 alpha level, sphericity was assumed. If sphericity was violated, a Greenhouse-Geisser test was performed with appropriate corrections. Two-tailed Pearson's tests were used for correlations. For post-hoc analyses, LSD was used with a Bonferroni correction to adjust for multiple comparisons, and a simple effects test was used to evaluate interactions.

Creation. Analysis revealed a statistically significant difference between conditions in creation time ($F=27.2$, $p<.001$), as seen in Figure 8. The only significant difference was that NP ($M=61.3$) took the least amount of time to create (D: $M=390$; OI: $M=434$; DOIC: $M=451$).

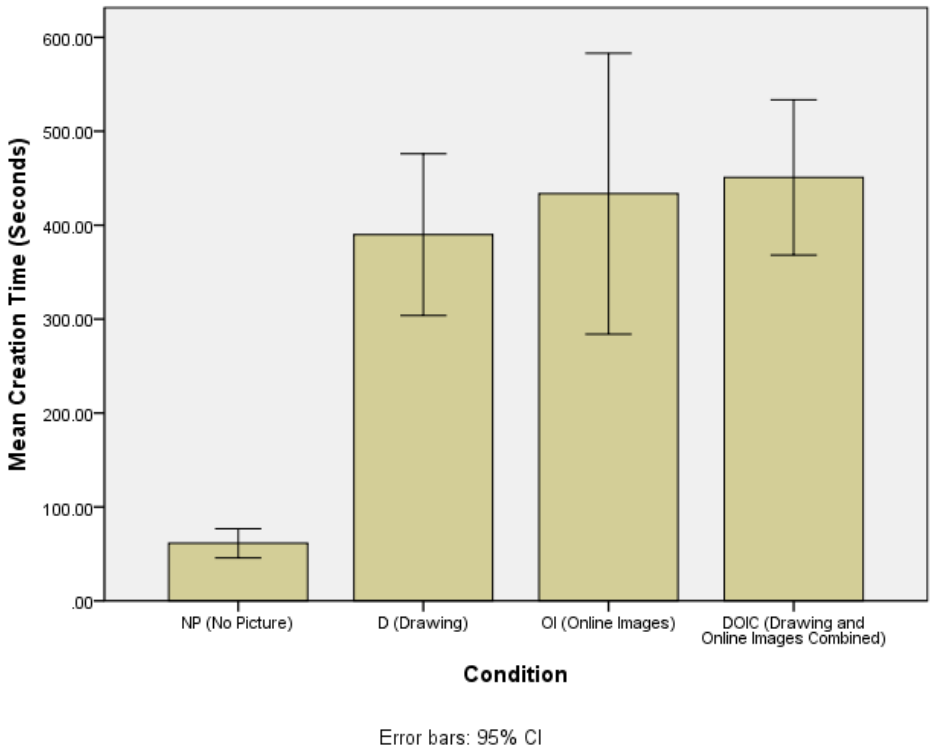

Figure 4.1. Account creation time (smaller values indicate increased efficiency)

Recall. No differences were found for recall time for successful logins among conditions (F=0.509, p=.677) or between sessions (F=0.241, p=.625), and no interaction was found (F=2.49, p=.067).

Login. Condition had no significant main effect on login success (χ^2=6.11, p=.106), as shown in Figure 9, although the mean success rate for NP was lower than the other conditions. Differences among sessions were found to be significant (χ^2=14.0, p<.001), with participants succeeding more during Login 1 (success rate of 52.1%) than in Login 2 (success rate of 26.0%). No interaction was found between Condition and Session (χ^2=0.450, p=.929).

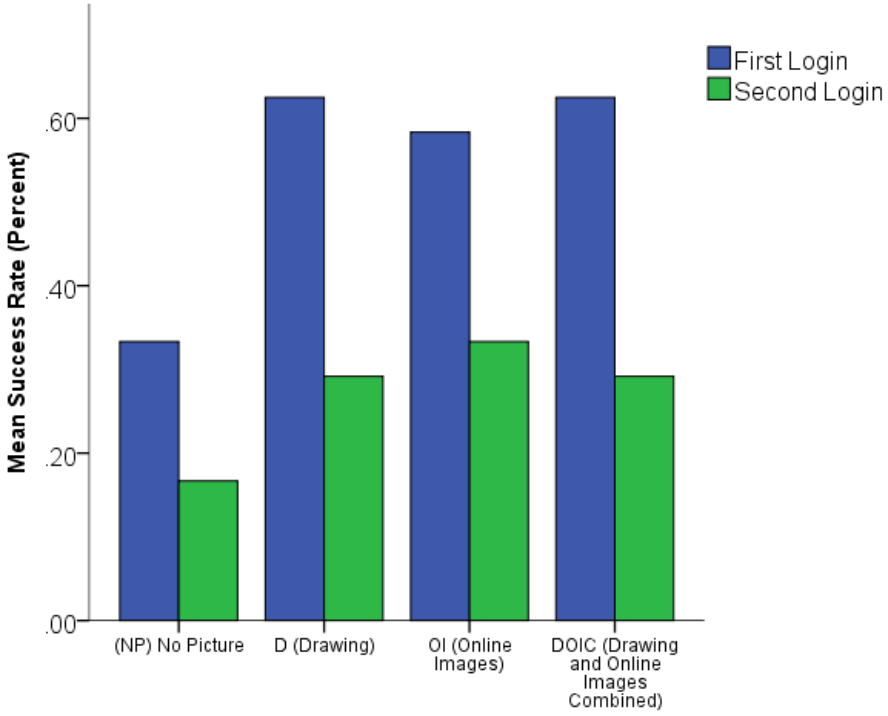

Figure 4.2. Success rate: login (larger values indicate increased effectiveness)

Figure 10 indicates that Condition affected Damerau-Levenshtein distance ($F=6.93$, $p=.002$), as did Session ($F=22.4$, $p<.001$). Participants in NP ($M=2.92$) had a larger Damerau-Levenshtein distance than those in other conditions (D: $M=1.52$; OI: $M=2.75$; DOIC: $M=2.63$), indicating that participants entered passwords less accurately during NP than in the Picture conditions across Logins 1 and 2. No differences were found among the Picture conditions. A larger Damerau-Levenshtein distance was also observed during Login 2 ($M=2.35$) than in Login 1 ($M=1.21$). No significant interactions

occurred between Condition and Session for Damerau-Levenshtein distance (F=0.590, p=.550).

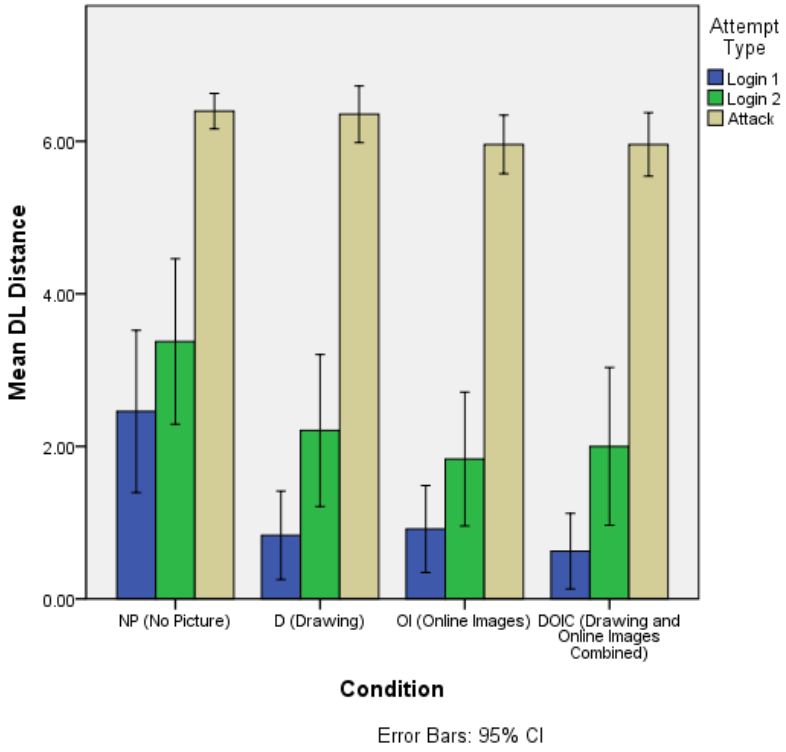

Figure 4.3. Damerau-Levenshtein distance: login and attack (smaller values indicate increased similarity)

Condition and Session affected Jaro-Winkler proximity (Condition: F=7.78, p=.002; Session: F=19.9, p<.001), as seen in Figure 11, but there was no significant interaction between Condition and Session for Jaro-Winkler proximity (F=0.220, p=.767). During the NP condition (M=.724), participants had significantly lower Jaro-Winkler proximities than in the other conditions (D: M=.885; OI: M=.879; DOIC: M=.910), meaning that they were further from the password in the NP condition than in the

Picture conditions. No significant differences were found among other conditions (p>.999). Jaro-Winkler proximity was higher during Login 1 (M=.899) than Login 2 (M=.800).

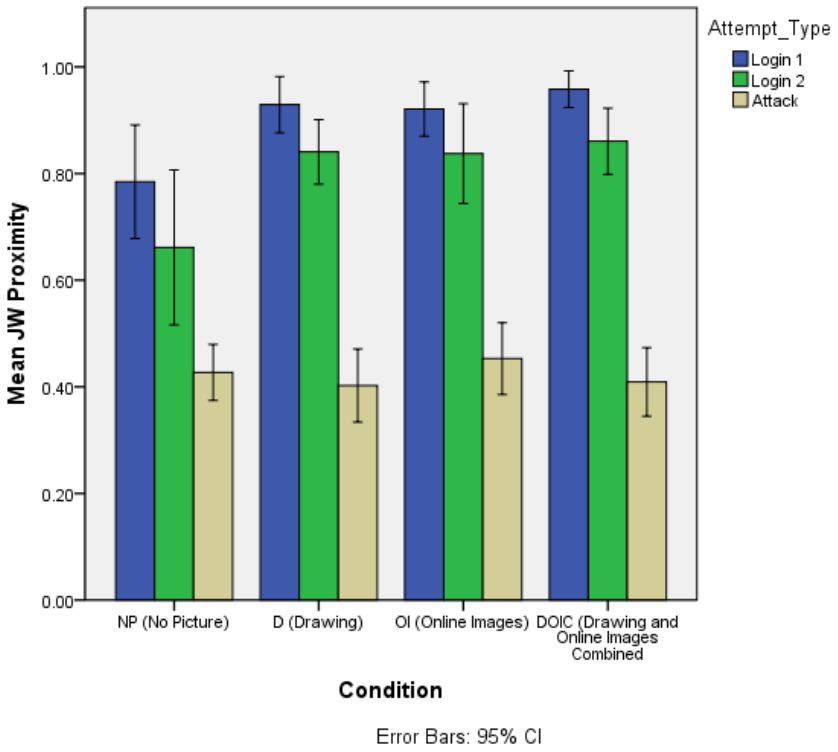

Figure 4.4. Jaro-Winkler proximity: login and attack (larger values indicate increased similarity)

Security. During attacks, no significant effect was found for Condition on Damerau-Levenshtein distance (F=2.37, p=.078), as seen in Figure 10, or on Jaro-Winkler proximity (F=0.520, p=.668), as seen in Figure 11. Time taken to attack correlated with success determining passwords, as measured by Damerau-Levenshtein distance (r=-.153, p=.009) and Jaro-Winkler proximity (r=.164, p=.005).

Subjective Ratings. NASA-TLX scores indicated higher ratings of physical demand ($F=7.13$, $p=.002$) during Creation than Login 1 ($p=.003$) or Login 2 ($p=.031$). NP was rated consistently lower in physical demand than the other conditions ($p<.001$). While DOIC was rated much less demanding during Logins 1 and 2 than Creation ($p<.001$), D was rated more demanding during Login 1 than Creation and Login 2 ($p<.001$), and OI was rated less demanding during Login 1 than during Creation and Login 2 ($p<.001$).

Participants rated their performance ($F=15.7$, $p<.001$) worse during Login 2 than during Creation ($p<.001$) or Login 1 ($p<.001$). Similarly, frustration ($F=7.73$, $p=.001$) was rated higher during Login 2 than during Creation ($p=.005$) or Login 1 ($p=.040$).

The usability of the systems ($F=7.70$, $p=.001$) was rated lower for Login 2 than during Creation ($p=.013$) or Login 1 ($p=.011$), as seen in Figure 12. Analysis revealed no main effects of Session on the remaining subjective ratings or of Condition on any subjective ratings.

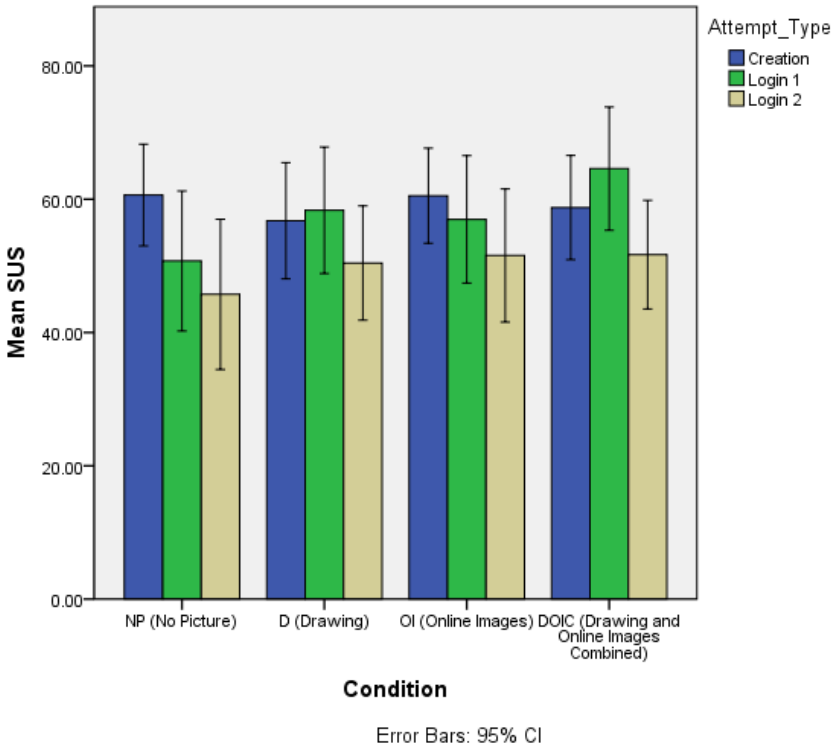

Figure 4.5. SUS: creation and login (larger values indicate increased usability)

Discussion

Account creation took longer in the Picture conditions than in the No Picture (NP) condition. However, no significant differences were found in recall time, success rates, or security among conditions. The increased memorability of passwords in the Picture conditions (according to Damerau-Levenshtein distance and Jaro-Winkler proximity) may justify the extra time required to create accounts, though it has not yet been shown if participants would willingly take the extra needed time to create the accounts.

As expected, login success, measured by success rate, Damerau-Levenshtein distance, and Jaro-Winkler proximity, dropped from the first to second session, confirming that participants tend to forget their passwords over time. Complementing these findings, participants rated Login 2 as characterized by more failure, less usability, and more frustration than Login 1. No differences were found for ratings of mental or temporal demand or effort. The low success rate (20-60%) was also expected because participants were asked to memorize four passwords at once and did not use them for a week; in practice, expected success rates for any one of these conditions would be higher.

Although no significant differences were found among conditions for login success, more sensitive measures (Damerau-Levenshtein distance and Jaro-Winkler proximity) revealed that creating pictures increases password memorability. This suggests that in the study in Chapter 3, the top performance of the system-generated mnemonic and picture condition was due, at least in part, to picture creation. However, it is unclear whether this benefit is due to creating the picture during account creation (employing the generation and self-reference effects), viewing the picture during login (to cue recall), taking more time to create the account, thinking more deeply about the phrase due to picture creation, or a particular combination of the above. Future studies might unpack these considerations in order to develop a more focused approach to the present authentication system.

Both Damerau-Levenshtein distance and Jaro-Winkler proximity (which are better suited to measure the threat of brute-force and guessing attacks, respectively) indicated that Picture conditions were no easier to crack than the No Picture (NP) condition. Time spent attempting to crack accounts correlated with coming closer to cracking them, suggesting that effort moderated cracking success.

Future studies might investigate the possible additional security gleaned from varying the structure of computer-generated phrases. This would not be expected to decrease memorability because participants would still see only one phrase structure per password. In fact, different phrase structures may help differentiate among accounts and decrease interference among phrases. Although the system's current small word bank relies on familiar words to make phrases more memorable, increasing the word bank might also enhance security by making pictures less susceptible to attack.

This study revealed that when computer-generated random passwords are combined with computer-generated mnemonic phrases, user-created pictures assist in password recall, enhancing usability without compromising security.

CHAPTER 5: INPUT MASKING TECHNIQUES FOR AUTHENTICATION

Introduction

This chapter is based on previously published work (Juang & Greenstein, 2011).

In contrast to Chapters 3 and 4, which dealt primarily with bolstering the memorability of system-generated passwords, the study in this chapter investigates how to increase the usability of the actual login process. When a user authenticates to a system, there are different methods that may be used to hide the login input from shoulder surfers.

At the extreme end of focusing entirely on usability, the text can be displayed as cleartext. At the opposite end of focusing solely on security, no feedback at all can be displayed to the user. In between these extremes are all the possible input masking techniques that aim to strike a balance between usability and security. The majority of previous work has attempted to strengthen security beyond the status quo of bullet masking, while this study focuses on improving on the usability of bullet masking.

The main innovation of this study is the implementation of a novel input masking technique that hides the login input in plain sight using decoys. The user can see the correct input in real time as it is being entered, while a shoulder surfer would not know which input on the screen is legitimate.

We aim to examine the effect that different input masking techniques, including our new system, have on the usability and security of the login process. We hypothesize

that our new system combines the usability benefits of cleartext with the security benefits of no feedback.

Implementation

We present Purloin: a novel input masking technique based on the concept of hiding something in plain view, from Edgar Allan Poe's "The Purloined Letter." We accomplish this without needing to alter the display of the sensitive input itself. Instead, by using superfluous input (decoys) to mask legitimate input, we strive to combine the usability advantage of unmasked text with the security advantage of masked text.

Purloin relies on the addition of any amount of decoy text surrounding typed text, as seen in Figure 13. The correct input text is automatically mapped to a certain color and position, which are tied together for redundancy. As a result, once the user first determines where to look, he or she will always know where to look for all future attempts. On the other hand, a shoulder surfer should not be able to tell where to look.

Figure 5.1. The decoy text obfuscates the password while also acting as a trap for potential attackers.

If the text to be entered is already known by the system, it can be mapped to a consistent position; otherwise, the position would have to be decided randomly. Even if the system does not know what the user will enter *a priori*, the user can still determine the appropriate position as he or she types. The decoy text is generated by applying a Caesar cipher (character-rotational encryption) to the plaintext input for each decoy. Letters, numbers, and symbols are all rotated within their own space so that each character is replaced with the same type of character. Each decoy is shifted by a different amount when possible, in order to avoid duplication of characters.

If submitted input text is incorrect but can be rotated into the correct text, Purloin knows a shoulder surfing attack has occurred and can lock out and record the

attacker and notify the legitimate user. This provides resistance against a camera-based attack that seeks to try each row one at a time.

In the case of repeated shoulder surfing, a dedicated attacker with long-term physical access could still try to figure out which text remains consistent across input sessions. This type of attack is prevented if the system stores both the text to be entered and the associated decoys. All rows can then be displayed consistently across sessions. Although this method requires additional information to be stored by the system, the result is an increase in security with no adverse effect on usability.

Based on a pilot study, we settled on ten rows of text as a tradeoff between security and usability. Once color was included to help signify the uniqueness of each row, the participants in our pilot study were able to quickly determine how to use Purloin without instructions. Providing basic support on first use would still make sense in a production environment, so the user is not startled.

We created a working Purloin implementation in Java 6 as well as a system for running the experiment, including faithful implementations for other common input masking techniques. We selected colors for Purloin such that no adjacent positions would be difficult to distinguish for all major types of color-blindness.

Method

Threat Model and Limitations

For our study, a shoulder surfer was defined as an adversary with the ability to listen to the surroundings and observe the user's screen (a standard 17-inch monitor)

and QWERTY keyboard. The shoulder surfer was allowed any technique to obtain the input – including taking notes – other than physical contact or looking at the user's provided sheet of assigned text. The user was allowed any technique to prevent the shoulder surfer from succeeding other than physical contact.

All input to be entered was in the form of randomly generated lowercase alphanumeric text strings (sequences of characters) of length eight. Although these did not represent "typical" weak user-generated passwords, they were in line with existing security guidelines for creating strong passwords (Burr et al., 2006). Participants were given practice with our assigned input strings, but their level of experience with the input would nevertheless be much higher with passwords that were already familiar to them.

The participants took turns as both the user and shoulder surfer, so they were well aware that the study involved testing computer security. In fact, because of these dual roles, participants tried hard to thwart their respective partner in a friendly bid to outperform the other. In the wild, many users do not realize when they are being shoulder surfed, and shoulder surfers are not free to act without fear of being caught. Ethical considerations would have to be addressed when conducting a real-world shoulder surfing study.

Subjects

We recruited 14 senior or graduate level engineering students (11 males, 3 females), ranging from age 22 to 28, via word of mouth. They were split into seven pairs

of participants. All participants had over five years of computer experience and normal or corrected-to-normal eyesight.

Experimental Design

The study utilized a within-subject repeated measures design. Participants were assigned to conditions based on a balanced Latin square to minimize order effects. Participants completed all of the input masking techniques investigated:

1. Cleartext: no input masking (displayed as normal text)
2. Invisible: no on-screen indication for typed text (usually seen in command line interfaces)
3. Bullet-masked: one bullet per typed character (ubiquitous on both the desktop and online)
4. Interval-masked: each newly typed character transforms into a bullet after the following character is typed or a 2-second interval has passed (usually seen in mobile devices)
5. Decoy-masked: typed text is hidden in plain view with the addition of decoy text surrounding it (see earlier Implementation section for Purloin)

"Bullet-masked with a checkbox to toggle into cleartext" was also considered, but our pilot study showed that users almost never selected the checkbox even when left alone and explicitly told they were in a secure location. Users typically claimed the checkbox was unnecessary since they could supposedly input text just fine without it,

and so it was not worth the effort to click it. Regardless of the validity of these claims, these findings made the inclusion of this technique redundant for this study.

Procedure

Each session started with a brief overview of the study, after which participants signed an informed consent form and completed a short pre-test demographic questionnaire. One participant was randomly selected to begin as the user and the other as the shoulder surfer.

For each masking technique, each user completed a set of three text strings. For each string, the user entered the text five times correctly. Incorrect inputs prompted the user to try again. On the fifth time for each string, the shoulder surfer was allowed that one opportunity to observe. Next, the shoulder surfer was asked to enter the observed text as accurately as possible using a second keyboard. Upon completion of each masking technique, the user completed the NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) and an abridged set of seven questions taken from the Computer System Usability Questionnaire (CSUQ) (Lewis, 1995).

After completing the entire list of fifteen inputs, the user assigned a unique ranked preference for each input masking technique during a short post-test questionnaire. Participants were given the chance to take a break before switching roles and repeating the procedure. The total time required to complete the full experiment was approximately one hour.

Dependent Variables

Objective measures were recorded by the system:

- Damerau-Levenshtein distance (Damerau, 1964): the length in edits (total sum of single character insertions, deletions, substitutions, and adjacent transpositions needed to transform one string to another) between the shoulder surfer's observation and the actual input
- Jaro-Winkler proximity (Winkler, 1990): the similarity or correlation between the shoulder surfer's observation and the actual input, normalized such that 0 indicates no similarity and 1 indicates equality
- Task completion time: mean time in seconds of all (five) correct inputs for each string, measured for each input from the first typed character to eventual successful entry
- Error rate: the number of incorrect inputs plus the number of times the user restarted (by hitting backspace at least three consecutive times to clear the entire input), divided by that sum plus the number of correct inputs (five)

Subjective measures were completed by the participants on paper forms and tallied during data analysis:

- CSUQ questions addressing ease of use, effectiveness, efficiency, perceived security, learnability, recoverability, and satisfaction, on a 5-point Likert scale
- NASA-TLX indices addressing mental demand, physical demand, temporal demand, performance, effort, and frustration, on a 7-point Likert scale

- Preference rankings for usability, security, and overall, determined by assigning each input masking technique a unique number from 1 to 5

Distance was chosen instead of simple match rate because an adversary would still be able to launch a successful attack by programmatically examining close text strings. The NASA-TLX was used to evaluate whether the added complexity or distraction of input masking contributed to user workload.

Results

We used SPSS 18 for the data analysis. For the majority of our dependent variables, we used a repeated measures one-way ANOVA with a 95% confidence interval to determine significance. Welch's correction was used whenever variances were heterogeneous. Given significant results, Tukey's HSD test showed which input masking techniques differed significantly from one another. Since the error rate and user preference rankings did not follow a normal distribution, we used a Friedman test for each of those measures. We then used pairwise Wilcoxon signed-rank tests with Dunn-Šidák correction to see which techniques differed significantly.

Objective Measures

Security. The results indicate significant differences for Damerau-Levenshtein distance ($F=46.8$, $p<.001$) and Jaro-Winkler proximity ($F=33.3$, $p<.001$). The Damerau-Levenshtein distance was shortest for cleartext ($M=0.640$), next shortest for interval-masked (2.38), and longest for the subset of invisible (5.57), bullet-masked (5.79), and decoy-masked (5.64). This is shown in Figure 14. The Jaro-Winkler proximity was higher

for cleartext (.958) and interval-masked (.856) than it was for invisible (.595), bullet-masked (.580), and decoy-masked (.558), as seen in Figure 15.

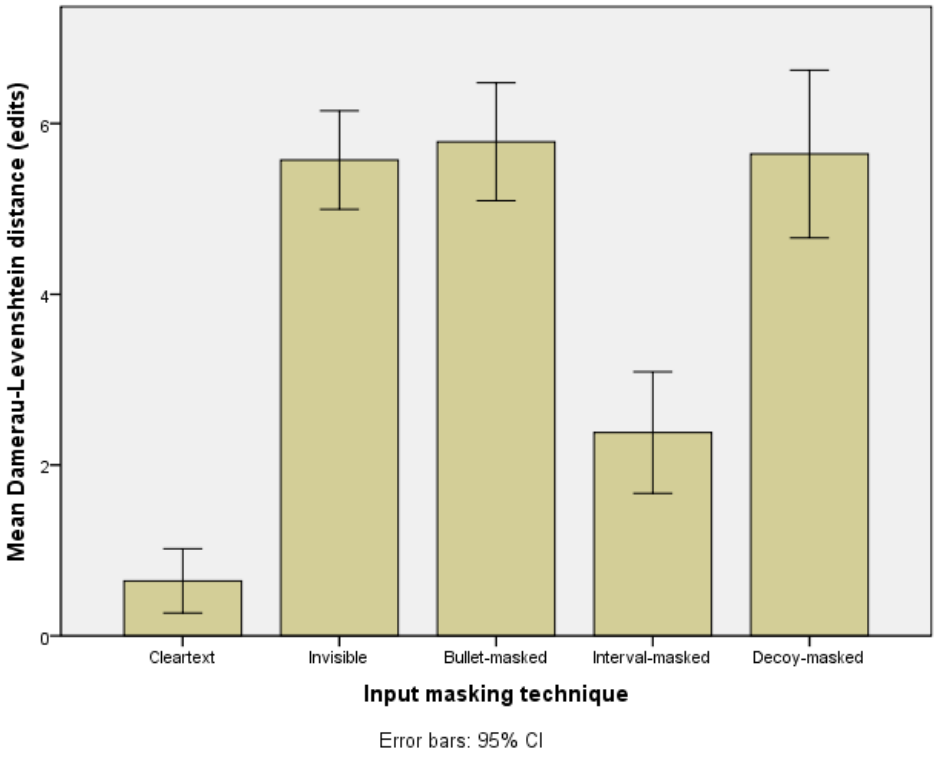

Figure 5.2. Damerau-Levenshtein distance (larger values indicate increased security)

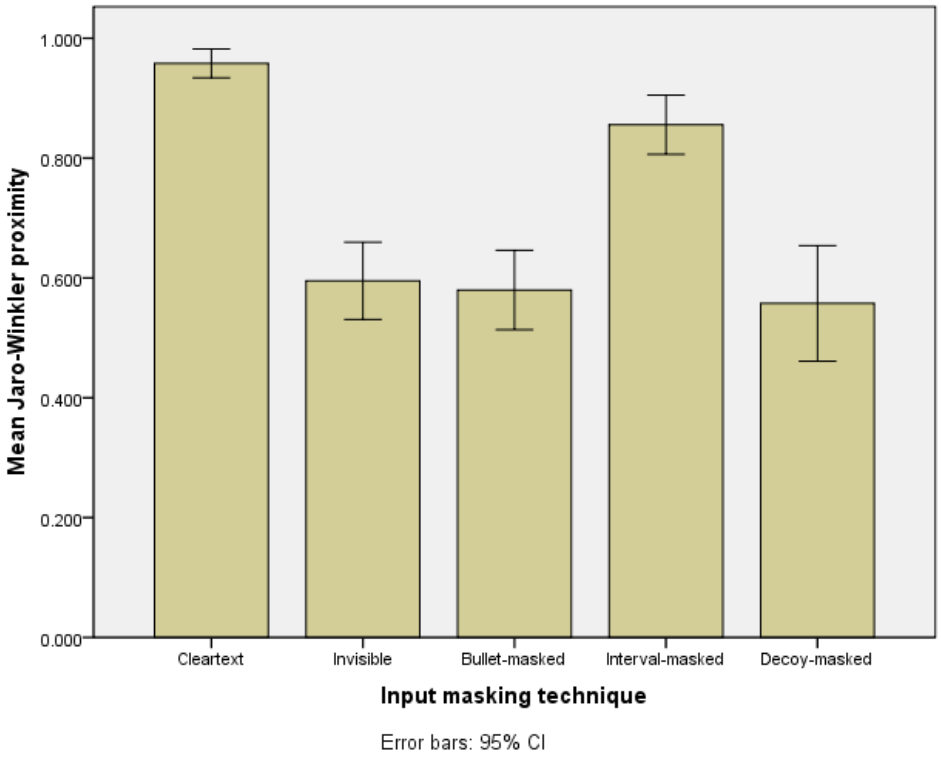

Figure 5.3. Jaro-Winkler proximity (smaller values indicate increased security)

Usability. As seen in Figure 16, there was no statistical significance for task completion time (F=0.533, p=.712). The error rate (χ^2=22.9, p<.001) was higher for invisible (M=.094), bullet-masked (.091), and interval-masked (.087) than it was for cleartext (.015) and decoy-masked (.020), as shown in Figure 17.

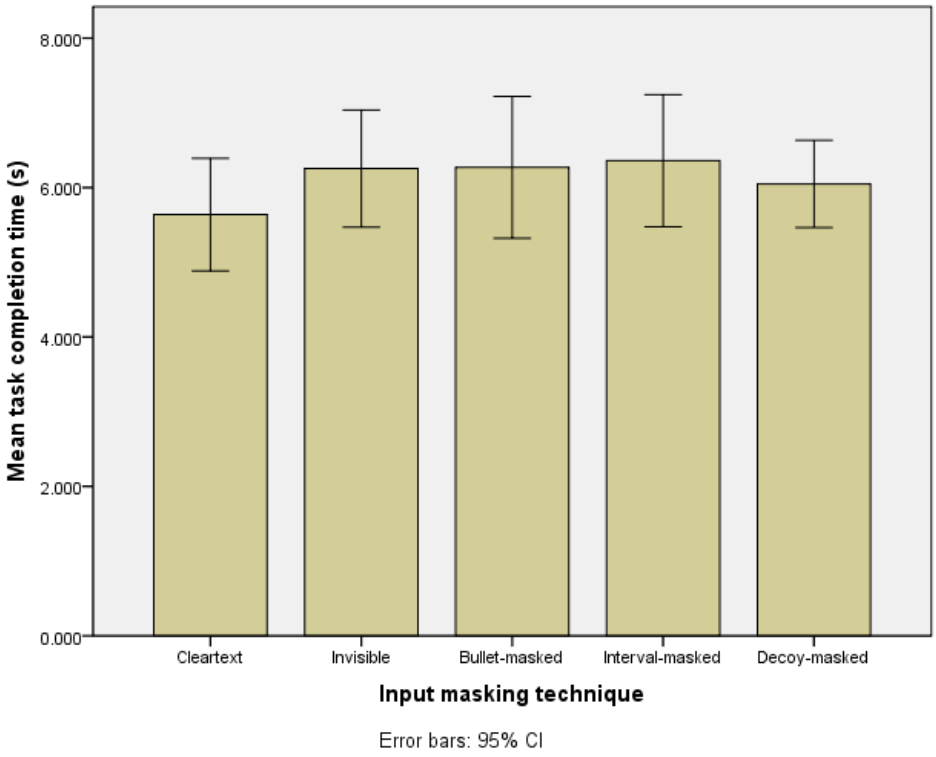

Figure 5.4. Task completion time (smaller values indicate increased efficiency)

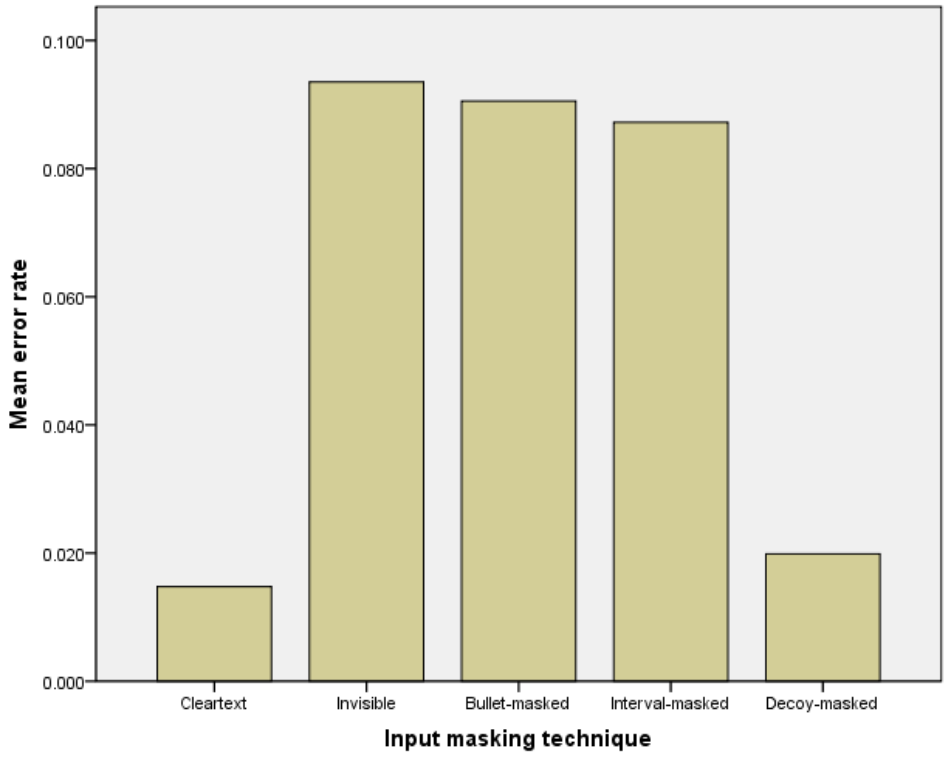

Figure 5.5. Error rate (smaller values indicate increased effectiveness)

Subjective Ratings

Usability. The invisible masking technique was rated lower for ease of use, effectiveness, efficiency, and learnability than the other input masking techniques. For recoverability ($F=25.3$, $p<.001$), invisible ($M=1.71$) was also rated lowest, with bullet-masked (3.14) rating lower than cleartext (4.50) and interval-masked (3.71). For perceived security ($F=7.91$, $p<.001$), cleartext (1.79) rated significantly lower than invisible (3.86), bullet-masked (4.07), and decoy-masked (3.79). Cleartext (2.50) was also rated lower than bullet-masked (3.71) and decoy-masked (3.79) in terms of overall satisfaction ($F=3.61$, $p=.010$). These results are shown in Figures 18 and 19.

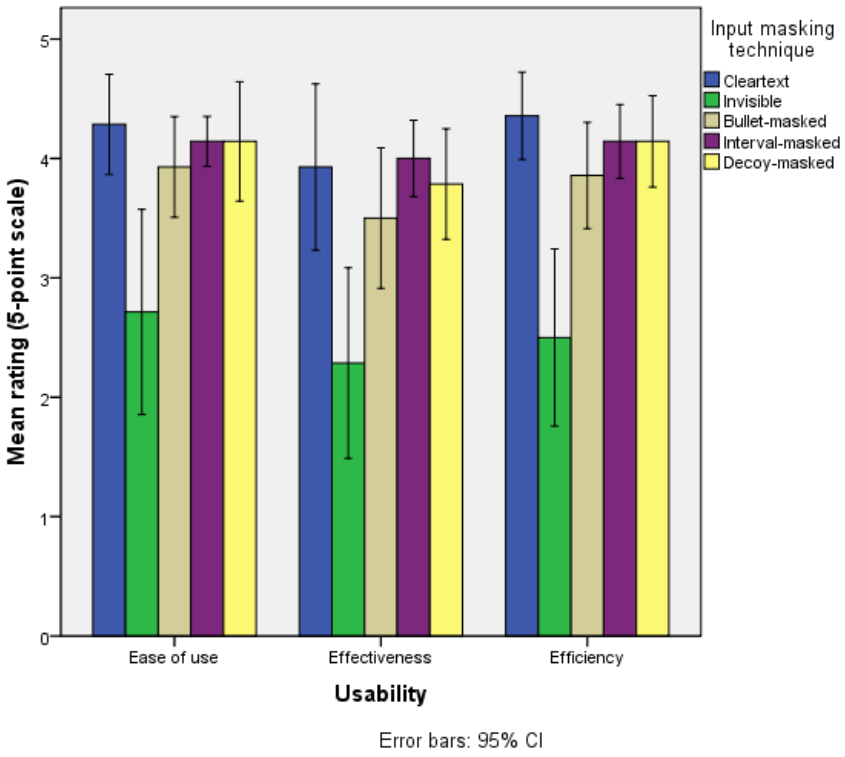

Figure 5.6. The first three subjective usability ratings (larger values indicate increased usability)

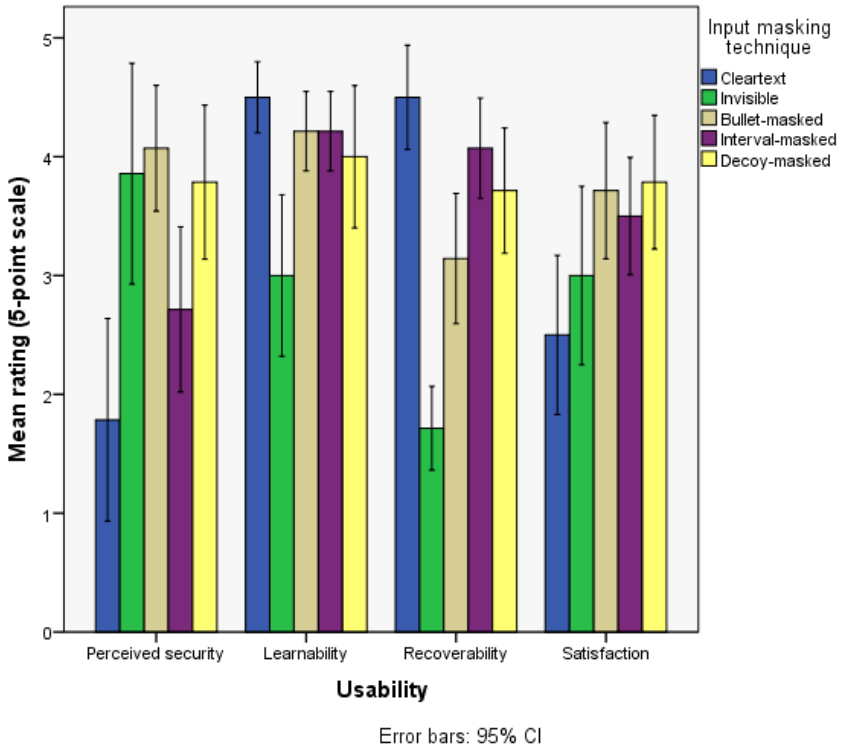

Figure 5.7. The last four subjective usability ratings (larger values indicate increased usability)

Workload. There was no statistical significance (p>.05) for physical demand, temporal demand, performance, or frustration. For mental demand (F=3.91, p=.007), invisible (M=4.36) was rated significantly higher than cleartext (1.79). Invisible (4.36) was rated higher than cleartext (2.29), bullet-masked (2.29), and interval-masked (2.43) in terms of effort (F=4.37, p=.003).

Preference Rankings

Usability. The rank of invisible (M=5.00) was significantly worse (χ^2=39.9, p<.001) than all others: cleartext (1.29), bullet-masked (3.21), interval-masked (2.86), and

decoy-masked (2.64). In addition, bullet-masked and interval-masked ranked worse than cleartext, the category winner. These results are displayed in Figure 20.

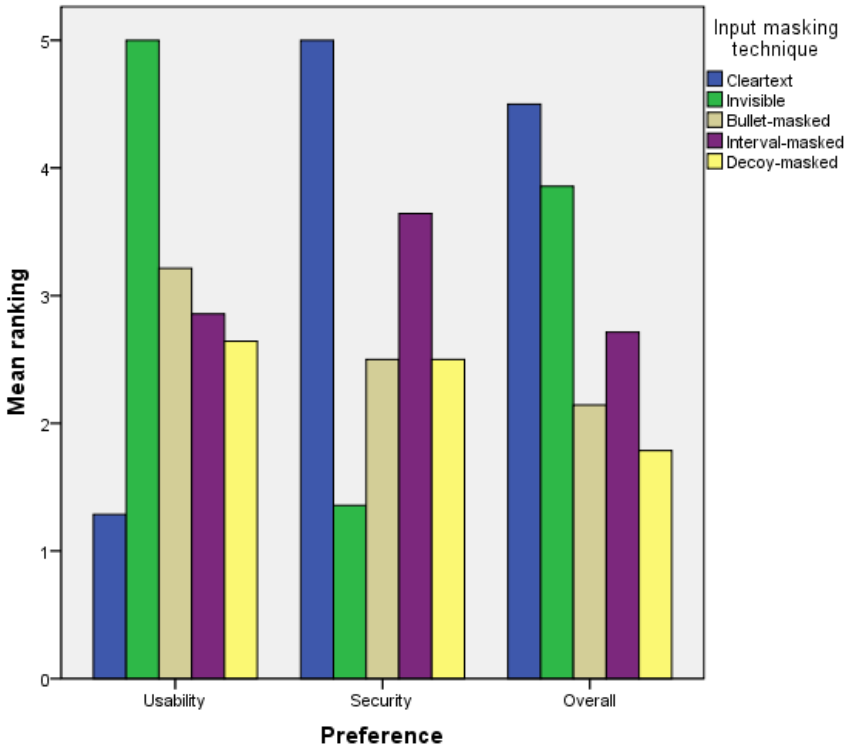

Figure 5.8. Preference rankings (smaller values indicate increased preference)

Security. Based on experience gained from both the user and shoulder surfer perspectives, participants ranked cleartext (M=5.00) significantly worse (χ^2=42.6, p<.001) than all others: invisible (1.36), bullet-masked (2.50), interval-masked (3.64), and decoy-masked (2.50). Interval-masked ranked worse than invisible and bullet-masked, the category winners. Although the bullet-masked and decoy-masked means were identical, the variability for decoy-masked was higher.

Overall. Based on the entire range of factors in selecting an input masking technique, cleartext (M=4.50) ranked significantly worse (χ^2=29.5, p<.001) than bullet-masked (2.14), interval-masked (2.71), and decoy-masked (1.79). Furthermore, invisible (3.86) ranked significantly worse than decoy-masked, the overall winner.

Discussion

The results support several conclusions with practical ramifications for input masking policy. Cleartext performed worst in both objective and subjective measures of security. Invisible input performed worst in both objective and especially subjective measures of usability and workload. These were the least preferred input masking methods overall.

Purloin (decoy-masked) appeared to indeed capture the best of both worlds, as it was the only input masking technique to place in the highest significant subset for objective measures of both security and usability. Notably, low error rates resulted from being able to see the entire input text as it was being entered. Subjective ratings for Purloin were also highly favorable. In addition, it exhibited the best overall preference ranking, although this ranking did not differ significantly from bullet-masked or interval-masked input.

Conversely, interval-masked input appeared to combine the worst of both worlds, as objective measures placed it at the bottom or near-bottom for both security and usability. Its subjective ratings and preference rankings were quite positive nonetheless. Tari, Ozok, and Holden had previously shown the rift between perceived

and actual shoulder surfing threat levels for graphical and non-dictionary passwords (2006). Interval-masked implementations are currently most prevalent on mobile devices. We strongly recommend against their use in desktop settings, but we would have to study mobile usage before making suggestions for that domain.

While conducting the experiment, we noticed that two participants devised a clever attack on Purloin. Because the user often hovered over the first key before beginning to type, these shoulder surfers were able to determine the first character and then locate the row on the screen that started with that character. After the study concluded, we modified the system to display the same initial characters on every row. The remaining characters follow the original algorithm.

The effects of reduced screen space on Purloin have yet to be studied. While a desktop system can display an overlay to deal with an overcrowded screen, physical limitations of mobile devices would necessitate smaller text size, tighter spacing, multiple columns of text in each row, or simply fewer rows.

One possible next step in our research with Purloin is to examine the interaction effects between type of input text and input masking technique. Since errors are expected to increase as the text string length or complexity grows, Purloin might see increased benefits when used with stronger passwords containing upper case and symbols. There might also be advantages for plain language passphrases, as the number of errors would be expected to increase along with increased typing speed and number of characters typed.

Purloin does not currently handle readable text gracefully, as readable input text tends to stand out amidst a sea of unreadable decoy text. We plan to incorporate a dictionary component into our system to support the masking of readable text by using words in the decoy text.

The widespread use of bullet-masked implementations appears to be supported by our results. While cleartext enjoys at the very least a concession of security for usability, we do not see any situations on the desktop for which it would be preferable to switch from bullet-masked to either invisible or interval-masked input. On the other hand, the results of this study support the idea that a switch to Purloin should increase usability with no loss in security.

One of the greatest benefits of Purloin is the ease of transition from existing input masking techniques. Unlike more complicated shoulder surfing resistant schemes, Purloin is simple enough to require neither instructions nor special equipment such as a touchscreen or eye tracker. Perhaps most importantly, it works cleanly with the password systems and policies already in deployment today.

CHAPTER 6: INTEGRATING MNEMONICS, INPUT FEEDBACK, AND PASSPHRASES

Introduction

The state of the art in the area of passwords lies in their longer, more readable cousins: passphrases. The theory is that the added length of passphrases should result in an increased level of security, while their word-based nature should simultaneously result in an increased level of usability. The drive behind a move to passphrases comes from a need to combat the progressing insecurity of passwords, as well as the ability of modern authentication systems to support passphrases.

Running parallel to the existing work in passwords, which focused mainly on user-generated passwords, the majority of prior research and in-the-wild implementations of passphrases has centered on user-generated passphrases. However, researchers have found that the memorability of user-generated passphrases was no different from that of user-generated passwords (Keith et al., 2007). The security benefit of user-generated passphrases has also been considered disappointing, as users often select relatively weak passphrases (Kuo et al., 2006).

Previous attempts at system-generated passphrases have produced passphrases that have been demonstrated to be difficult to use (Leonhard & Venkatakrishnan, 2007). The likely cause is utilizing a wordlist that contains far too many words, resulting in individual words that can be extremely uncommon and hard to remember.

Both user-generated and system-generated passphrases suffer from a great usability concern during authentication: typographical errors. Often when the user has

recalled the passphrase correctly, login will still fail due to erroneous input caused by a lack of feedback provided by standard input masking techniques.

The research in this chapter incorporates the three major innovations from Chapters 3-5: a specialized wordlist and sentence template, a user-created mnemonic picture that provides cued recall, and a decoy-based input masking technique. We apply these innovations to system-generated passphrases, with the expectation that they will prove particularly well suited to passphrases. In the process, we hope to advance the state of digital authentication.

Implementation

We implemented a new system that integrates our previous contributions (a specialized wordlist and sentence template, user-created mnemonic pictures shown during login, and a decoy-based input masking technique) with system-generated passphrase authentication. The basis for our system is the automated mnemonic phrase generator used in Chapters 3 and 4. However, instead of merely using the generated phrase to remember a system-generated password, the user now inputs the phrase directly to sign in.

The user-created mnemonic pictures were incorporated, with no significant changes necessary, into the existing authentication system from the work in Chapters 3 and 4. Furthermore, the decoy-based input masking technique from Chapter 5 was integrated with the existing authentication system in the formation of our new system. The decoy-based input masking technique was modified such that decoys are drawn

from the other words in the wordlist. Without this change, it would be easy for a shoulder surfer to identify the legitimate line amidst the character-rotated gibberish.

Unlike in Chapters 3 and 4, since the generated phrase itself is now the authentication secret, the design of the wordlist must be changed to effect the security enhancements of a passphrase. It proved to be a delicate balance to expand the size of the wordlist to increase security while ensuring that the usability of the system was not degraded. Our goal was to maintain the usage of simple words.

We developed two different wordlists that can be used with our new system, differing primarily in the sentence structure they embody. The first passphrase structure is not actually based on any designed sentence structure. Instead, one wordlist is used for all of the words in the passphrase. Although different types of words are in the wordlist, the permutation of words is not guaranteed or expected to make semantic or even grammatical sense; this lack of formal structure matches the majority of existing system-generated passphrases. Each of the four words in the passphrase is drawn randomly without replacement from the wordlist.

Users are shown four randomly selected passphrases (e.g., *succeed complete murder aid*, *weak parade of chemistry*, *point local instrument pain*, and *weigh in jewel girl*) and are given the opportunity to choose the passphrase out of the four that they are most comfortable with. The wordlist used is based on the 1510-word Special English list (Kelly, 2010). The exact wordlist was determined through pilot testing to filter out problematic words. Sixty words were removed to maintain the desired 40 bits of

entropy; the rationale and calculations behind this decision are covered at the beginning of the Method section.

The second passphrase structure is based on the existing 7-word sentence structure from Chapters 3 and 4. The associated wordlist was manually expanded beyond 26 animals, 26 adjectives, 26 actions, 26 foods, and so forth. Since English speakers know comfortably many more than 26 each of these types of words, we found that we were able to maintain the usage of simple words while increasing the level of security. In fact, we were even able to eliminate the need for one of the words, resulting in a new 6-word sentence structure with varying numbers of possibilities for the different word positions. These values were selected to maintain the same 40 bits of entropy as the first passphrase structure; these calculations are likewise covered at the beginning of the Method section. Again, the exact wordlist was determined through pilot testing to screen out problematic words.

Implementation Considerations

Shay et al. (2012) determined that roughly 1000 to 2000 words is a reasonable size for wordlists for system-generated passphrases. In fact, to avoid reinventing the wheel, our original idea for our new system was to reuse the same 1024-word list from their paper, derived from the most common words in the Corpus of Contemporary American English (COCA). A survey of available wordlists found that none of them fit our requirements perfectly for system-generated passphrases, although some came closer than others. This is due to the differences between the purposes behind the design of

most wordlists and the requirements for a usable passphrase-generating wordlist. Some of these concerns, in roughly descending order of importance, include:

- The wordlist is too large (e.g., Diceware list, Crypt::Diceware).
- The wordlist is outdated (e.g., Basic English, General Service List).
- The wordlist is domain-specific (e.g., Business English, Simplified Technical English for aerospace, Dolch Word List for children).
- The wordlist is commercialized, or the methodology behind its construction is unclear (e.g., Globish).
- The wordlist contains American-centric proper nouns (e.g., XKPasswd) or British spellings (e.g., Oxford English Corpus list).
- A wordlist based on the most common words may contain proper nouns, abbreviations, and slang (e.g., COCA). It may be especially volatile over time. It may also include semantic duplicates that cause interference with one another.
- A wordlist based on teaching English as a second language (e.g., Special English, Basic English, General Service List) may contain words that are prioritized as being useful to learn – perhaps appearing frequently in news articles or travel conversations – rather than simple.

Taking into account these issues, we decided the best fit was to start with the 1510-word Special English list (Kelly, 2010). This wordlist was created in 1959 by Voice of America (VOA), the official external broadcaster of the United States government,

and updated regularly since. It is aimed at international listeners and English learners. As stated earlier, pilot testing was conducted to eliminate the more troublesome words.

An additional benefit of basing our wordlist on the Special English list is that each word in the list comes with a simple definition. Our system incorporates these definitions as tooltips, so that a user who does not understand a particular word can mouse over the word to receive a definition. This proves to be especially helpful since our system relies on user-created mnemonic pictures.

For the decoy-based input masking to function appropriately, the previous method of a rotational cipher was changed. In the new system, the decoys come from other words in the wordlist. We considered prioritizing the decoy words for each legitimate word by selecting decoys that are most similar to the legitimate word. While this would possibly increase the security against shoulder surfing, it would also decrease the resistance against human guessing, as attackers would be able to begin guessing near the similar decoys.

The fundamental issue here is that the more the decoys are based on the passphrase, the more information is leaked during guessing attacks. Even using decoys from the same wordlist or sentence structure leaks information about the wordlist or sentence structure, but this is considered an acceptable tradeoff.

If security against camera-based shoulder surfing attacks is instead the top priority, our system can be adapted to generate consistent decoys in the same cohort of similar words and always in the same order. In a simplified example with three rows,

buck, *bump*, and *bust* would always result in those three words in the same rows, no matter which of the three was typed. This design decision is so an attacker cannot record the screen with a camera and later determine which input results in the recorded screen. The system can also detect what has been typed so far and identify a shoulder surfing attack by noticing an in-progress typed decoy, even if that decoy is never actually submitted as a login attempt. The system would then lock out and record the attacker and alert the legitimate user of the shoulder surfing attack.

Maximizing the similarity between decoys would provide better defense against shoulder surfers who attempt to look back and forth between the user's fingers and the screen. This system could still defend against intrusion attempts by detecting attacks on decoy passphrases and words, but testing would show if the false alarm rate (i.e., locking out legitimate users) can remain absolutely at zero. Whereas with the rotational cipher, it would be extraordinarily unlikely for a decoy to be accidentally typed, it is relatively more feasible that a legitimate user could type *buck* instead of *bump*.

As previously stated, the user-created picture aids were readily integrated into the new system without further changes. The user's interaction with the system has changed somewhat, since the generated passphrases are different from the previous mnemonic phrases, but the implementation of the system itself remains the same. In contrast to complex character-based passwords containing letters, numbers, and symbols, the word-based structure of passphrases allows the user to envision a scene or story about the authentication secret (Jeyaraman & Topkara, 2005).

There is one last consideration concerning the simultaneous display of mnemonic pictures with decoy masking. Since a shoulder surfer could see which of the possible decoys best fits the picture, the combination of the two techniques could result in a degraded level of security against shoulder surfing attacks. This attack would be particularly effective when carefully reviewing a camera recording. To defend against this potential vulnerability, we hide the picture by default and allow the user to show the picture. The user can still receive memory assistance when necessary (e.g., when learning a new passphrase or for less commonly used passphrases), but frequently or publicly used passphrases need not always have their picture displayed.

A final addition to the basic system-generated passphrase system is the idea of error correction to reduce typographical errors, also known as automatic spell checking. Because the wordlists we use are tremendously smaller than a wordlist found in a word processor or web browser, our system is able to correct to the intended word more aggressively and accurately. This is accomplished by converting the user's input to the word in the wordlist with the shortest Damerau-Levenshtein distance (the identical word in case of a match); the word with the highest Jaro-Winkler similarity is used in case of a tie. Damerau-Levenshtein distance (Equation 1) and Jaro-Winkler proximity (Equation 2) are formally defined:

$$distance(x, y) = i + d + s + t \tag{1}$$

$$to\ minimally\ transform\ x\ into\ y, where$$

$$i = insertions$$

$$d = deletions$$

$$s = substitutions$$

$$t = adjacent\ transpositions$$

$$proximity(x, y) = \begin{cases} d & if\ d \leq b \\ d + p \times \min(s, l) \times (1 - d) & otherwise \end{cases} \tag{2}$$

$$where$$

$$b = boost\ threshold = .7\ (canonically)$$

$$p = scaling\ factor = .1\ (canonically)$$

$$s = prefix\ size = 4\ (canonically)$$

$$l = length\ of\ common\ prefix$$

$$d = \begin{cases} 0 & if\ \min(m, x, y) = 0 \\ \frac{1}{3}\left(\frac{m}{|x|} + \frac{m}{|y|} + \frac{m - \left\lfloor \frac{t}{2} \right\rfloor}{m}\right) & otherwise \end{cases}$$

$$where$$

$$m = matching\ characters$$

$$\left(identical\ and\ not\ farther\ than\ \left\lfloor \frac{\max(|x|, |y|)}{2} \right\rfloor - 1\right)$$

$$t = transpositions$$

$$(matching\ but\ in\ a\ different\ order)$$

With our system, if a word in the correct passphrase is *bump*, then *bmup* would also be accepted. If an attacker is attempting to input a decoy such as *bust*, close typographical errors such as *bush* should still resolve to the closest word, which is a decoy in this case. Because we correct to both legitimate words and decoys with equal weight, the security of our new system remains unaffected.

Method

Independent Variable

The goal of this research is to examine the effect of authentication scheme on the usability and security of digital authentication. Accordingly, the main independent variable in the study is authentication scheme; session is another independent variable, but we are only interested in its interaction with authentication scheme, not any main effects.

The following conditions were selected to fulfill 40 bits of entropy, which is a commonly used level of security for passwords protecting valuable resources (Biancuzzi, 2006; Hruska, 2011; Shiner, 2013). The security of the system can then also be enhanced through key stretching (i.e., repeated password hashing designed to increase computation time) (Percival, 2009). The first two conditions represent the most common and recommended currently available passphrase systems, and the last two represent our new system using two different wordlists:

1. A user-generated passphrase of at least 24 characters (including spaces) with no other restrictions (see Figure 21). No mnemonics are used, and the input masking is standard bullet masking. $(24\ total\ characters = 1\ character \times 4\ bits + 7\ characters \times 2\ bits + 12\ characters \times 1.5\ bits + 4\ characters \times 1\ bit = 40\ bits\ of\ entropy)$ (Burr et al., 2006)

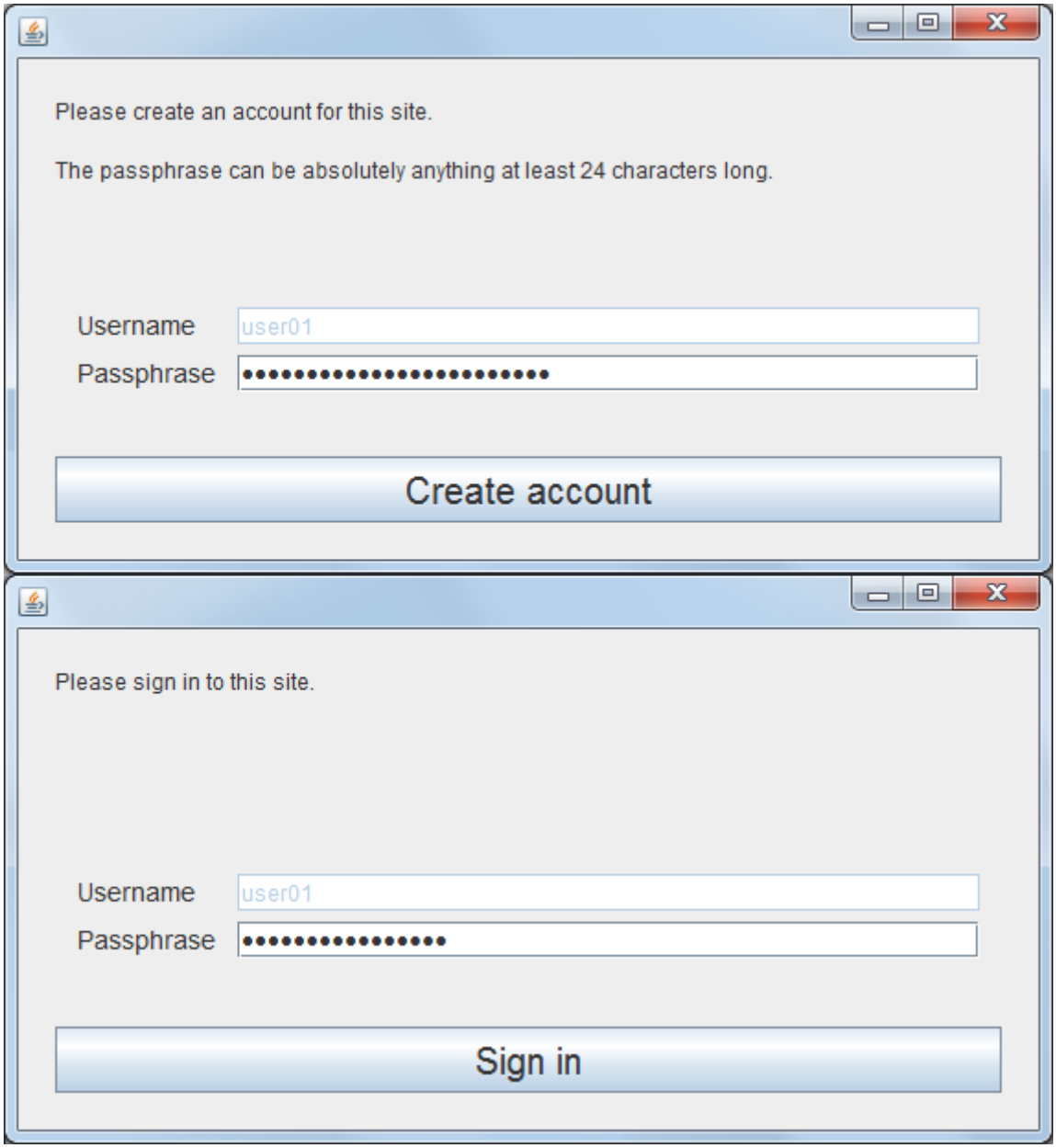

Figure 6.1. The user-generated passphrase condition (#1) appears similar during creation and login.

2. A system-generated passphrase utilizing three words drawn randomly without replacement from the Diceware8k list (Reinhold, 2012) supplemented with Beale's alternate list (Reinhold, 2012) and the shortest words from the 1Password expanded English Diceware list (Shiner, 2013), totaling 10326 unique words (see Figure 22), i.e. "Diceware10k." No mnemonics are used, and the input masking is standard bullet masking. $(3\ words \times \log_2 10326\ possible\ words = 40.0\ bits\ of\ entropy)$

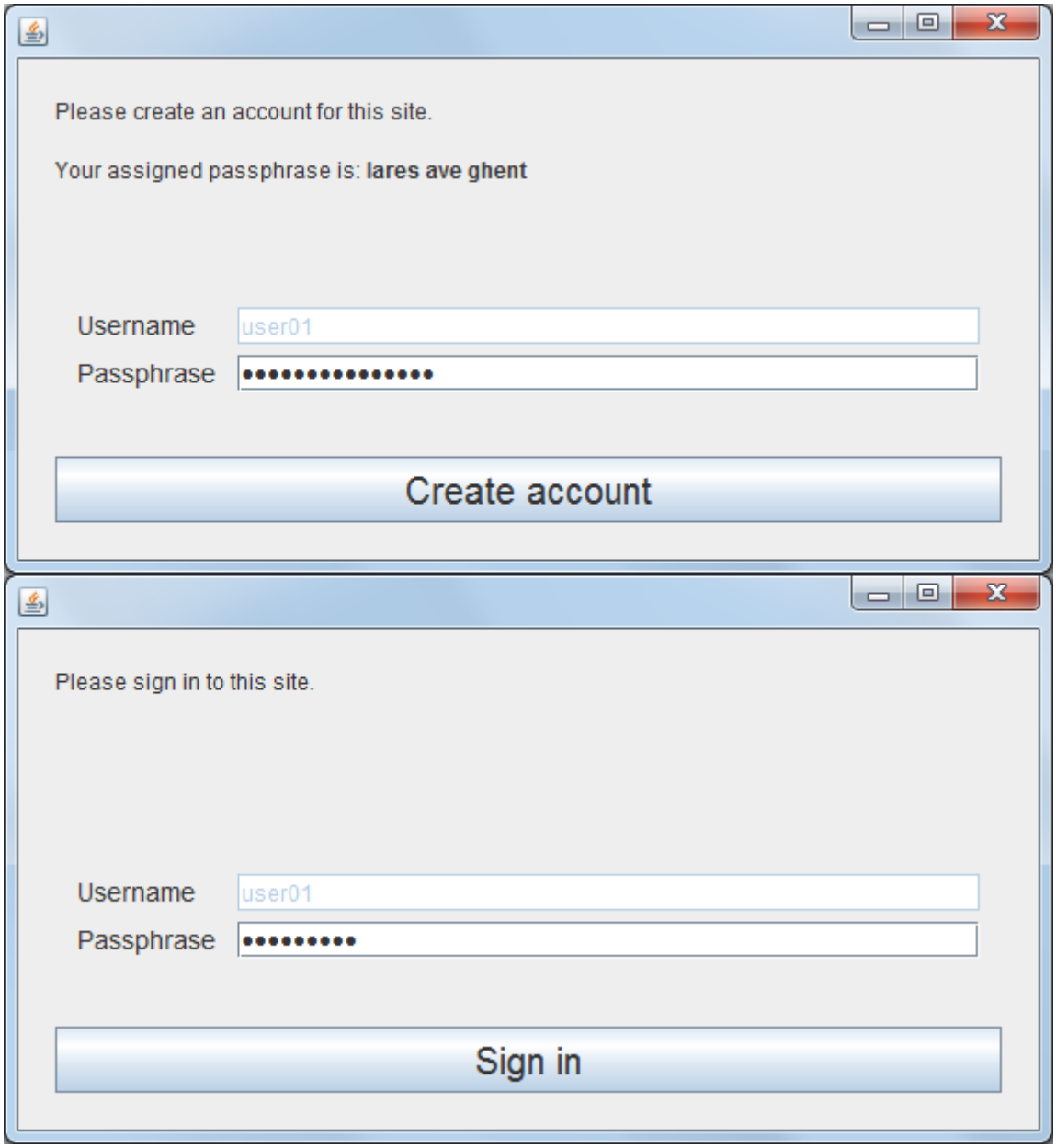

Figure 6.2. The Diceware10k system-generated passphrase condition (#2) results in shorter passphrases.

3. A passphrase chosen from a set of four system-generated passphrases, each utilizing four words drawn randomly without replacement from an abridged version of the Special English list (Kelly, 2010) containing 1450 words with definition tooltips (see Figure 23). A user-created mnemonic picture is shown during login, error correction is enabled, and the input masking is decoy masking. $(4\ words \times \log_2 1450\ possible\ words - \log_2 4\ choices = 40.0\ bits\ of\ entropy)$

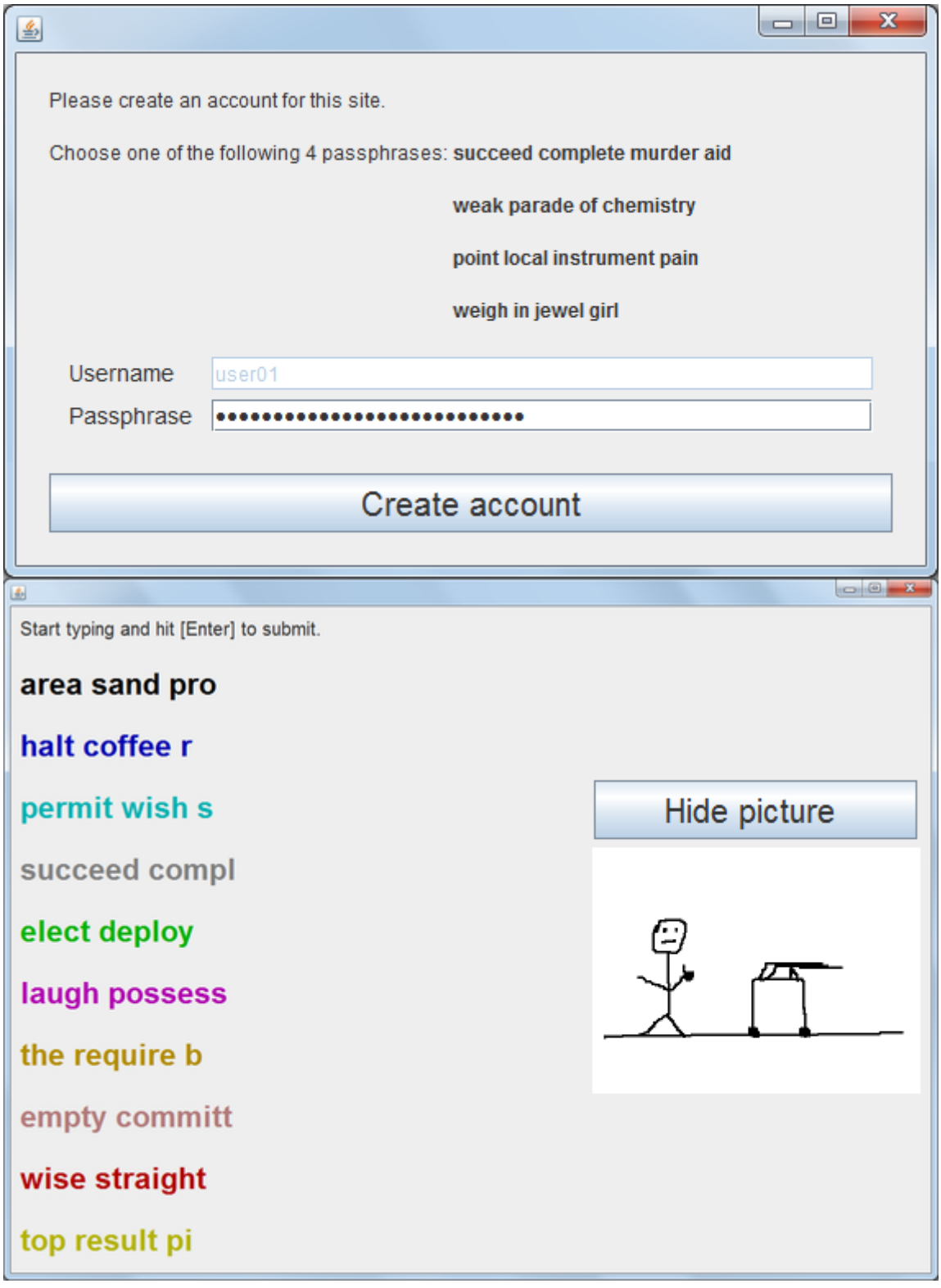

Figure 6.3. The Special English wordlist condition (#3) allows the user to select one of four passphrases.

4. A system-generated passphrase utilizing a 6-word sentence structure based on the 7-word structure from Chapters 3 and 4, with an expanded number of possible randomly drawn words for each location (see Figure 24). A user-created mnemonic picture is shown during login, error correction is enabled, and the input masking is decoy masking. $(\log_2(151 \times 151 \times 155 \times 61 \times 78 \times 66)\ possible\ combinations = 40.0\ bits\ of\ entropy)$

Figure 6.4. The 6-word sentence condition (#4) shows the mnemonic picture when selected by the user.

Dependent Variables

Our study employed both quantitative and qualitative measures. The quantitative dependent variables measured the security and usability of the authentication schemes. The data for these objective measures were recorded by the system:

- Recall time: seconds taken to either succeed or eventually fail to sign in to an account, measured starting from the presentation of the login screen
- Damerau-Levenshtein distance (Damerau, 1964) for recall effectiveness, shoulder surfing security, and cracking security: the length in edits (total sum of single character insertions, deletions, substitutions, and adjacent transpositions needed to transform one string to another) between the guess of a passphrase and the actual stored passphrase
- Jaro-Winkler proximity (Winkler, 1990) for recall effectiveness, shoulder surfing security, and cracking security: the similarity or correlation between the guess of a passphrase and the actual stored passphrase, normalized such that 0 indicates no similarity and 1 indicates equality
- Recall success: a binary metric of either success or failure to sign in
- Shoulder surfing resist success: a binary metric of either resisting shoulder surfing or failing to do so
- Cracking resist success: a binary metric of either resisting cracking or failing to do so, based on guesses by human attackers

- Cracking resist attempts: the number of attempts taken by a machine to crack the passphrase

It is worth noting that creation time was not included as a dependent variable, although we had previously measured creation time in our studies from Chapters 3 and 4. We stated that the effect of condition on performance could be mediated by creation time, but that we also valued the indirect effect of encouraging the user to think more deeply during account creation. For this study, we decided to examine the direct effect without mediation by standardizing each instance of account creation to five minutes each, regardless of the condition.

These quantitative subjective measures were recorded by the system:

- Creation and recall NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) indices addressing mental demand, physical demand, temporal demand, performance, effort, and frustration, each on a 7-point Likert scale
- Creation and recall System Usability Scale (SUS) (Brooke, 1996) scores addressing usability, each out of a total of 100 possible points
- Perceived level of security on a 7-point Likert scale
- Overall rating on a 7-point Likert scale

Qualitative research was also used to assess the security and usability of the authentication schemes. These qualitative subjective measures were collected by the researcher:

- Direct observation of the participant during the experimental tasks
- Post-test open-ended interview questions about likes, dislikes, suggestions, and perceived usability, security, and overall qualities

Due to the largely internal nature of the tasks, especially during recall, any video recordings of participants would be low on visible actions. Therefore, it was decided that a retrospective think-aloud protocol would not contribute much additional information over direct observation and post-test interview questions.

Hypotheses

Prior to conducting the study, we proposed four primary hypotheses:

1. Recall effectiveness measures will be most favorable for the new system conditions (#3 and #4).
2. Shoulder surfing security measures will not differ between authentication schemes.
3. Cracking security measures will be least favorable overall for the user-generated passphrase condition (#1).
4. Subjective usability measures will be least favorable for the Diceware10k system-generated passphrase condition (#2).

For the first primary hypothesis, the work from Chapters 3 and 4 indicated that mnemonic aids improved recall effectiveness, although the conditions examined were all based on system-generated passwords. The decoy masking technique from Chapter 5 should also reduce the number of errors made during the recall task. Since the new system conditions (#3 and #4) integrate techniques designed to improve memorability and usability, they should in theory exhibit the highest levels of recall effectiveness: lower Damerau-Levenshtein distance, higher Jaro-Winkler proximity, and higher recall success.

For the second primary hypothesis, the work from Chapter 5 indicated that the use of decoy masking over conventional bullet masking had no adverse effect on the level of security against shoulder surfing, although the study examined randomly generated passwords instead of passphrases. The modifications to the decoy masking system to support passphrases should preserve this strong level of security in the new system conditions (#3 and #4) versus the bullet masking conditions (#1 and #2). We expected to see no significant differences in Damerau-Levenshtein distance, Jaro-Winkler proximity, and resist success for shoulder surfing security.

For the third primary hypothesis, human attackers would not be expected to come close to correctly guessing any legitimate passphrases. Against machine cracking, the three system-generated passphrase conditions (#2, #3, and #4) should share similar levels of security; each of these conditions is randomly generated and designed to capture 40 bits of entropy. According to NIST guidelines (Burr et al., 2006), the user-

generated passphrase condition (#1) is estimated to have 40 bits of entropy. However, since users tend to choose patterned, less secure passwords and passphrases, it is still probable that the user-generated passphrase condition would exhibit the lowest levels of machine cracking security, as measured by lower cracking resist attempts.

For the fourth primary hypothesis, since users typically prefer to choose their own passwords (Proctor et al., 2002), perceived usability would not likely be the lowest for the user-generated passphrase condition (#1). When comparing the three system-generated passphrase conditions, the new system conditions (#3 and #4) integrate techniques to improve usability over the Diceware10k system-generated passphrase condition (#2). Research has also indicated that the large wordlist size of Diceware systems leads to a reduction in usability (Leonhard & Venkatakrishnan, 2007), as measured subjectively in our study by recall SUS and overall rating.

Participants

We recruited a fairly diverse set of 52 participants (28 males, 24 females), ranging from age 20 to 61 (M=29), via word of mouth and e-mail. Fifty of our participants returned between 7 and 11 days later for a second session; the other two participants were lost to attrition. The 50 participants who returned were each compensated with a $20 gift card for completing our study. The study utilized a within-subject design where each participant received each condition. All participants were assigned to conditions based on a balanced Latin square.

The number of participants was determined based on *a priori* power analysis using G*Power 3. With conservative estimates of medium effect size for ANOVA (f=.25) (Cohen, 1988) and desired power of .95, the required sample size for within-subject repeated measures ANOVA is 44. For comparison, using a large effect size for ANOVA (f=.40) and desired power of .80 yields a required sample size of 16. We decided on 52 participants as a sort of buffer (power of .975) and with the knowledge that many study designs tend to be underpowered rather than overpowered.

We considered running this study using Amazon Mechanical Turk, the most popular crowdsourcing marketplace. However, there are several factors that make this study poorly suited to Mechanical Turk:

- Most importantly, the shoulder surfing data cannot be collected online and must be collected in a laboratory setting.
- Our implementation of user-created pictures is tightly coupled with the local experimental setup to support automatic opening, saving, and closing. Because the online drawing apps we found were determined to be poor and unfamiliar to our pilot testers, an online replacement would rely on the participant creating a picture on his or her own terms and then uploading the picture to our server.
- One of the largest advantages of Mechanical Turk is the rapid collection of data. This is greatly neutralized in our study due to the turnaround time required, although Mechanical Turk's other large advantages (ecological

validity and being able to easily recruit and pay tremendous numbers of participants) would still remain. In addition, the turnaround time would lead to particularly large levels of attrition when compared to a laboratory study. By standardizing each instance of account creation to five minutes, we were able to run multiple participants simultaneously in the laboratory, mitigating some of the differences in data collection speed.

Procedure

Participants were first greeted, given a brief overview of the study, and asked to complete an informed consent form. Notably, participants were informed that they could be asked not to return for the second session if they violated instructions such as not including the passphrase as text in a picture. Each condition of the within-subject design was standardized at five minutes per condition, with a one-minute warning given after four minutes had elapsed. Upon completing the creation phase for each condition, participants were asked to complete NASA-TLX and SUS questionnaires. Once all four accounts were created, participants answered some open-ended interview questions regarding the account creation task and completed a basic demographic questionnaire; because multiple participants were run at the same time, we employed a focus group format. The amount of time required to complete the creation phase was approximately thirty minutes.

Participants then moved to the recall phase. One of the four accounts was randomly selected and participants were prompted to sign in within a maximum of five

attempts; this process was then repeated for the remaining three accounts. After each condition, participants were asked to complete NASA-TLX and SUS questionnaires, this time regarding the recall task. Participants were also asked to provide a rating for the perceived level of security and an overall assessment. Once all four accounts were attempted, participants answered some open-ended interview questions about their likes, dislikes, suggestions, and any other relevant opinions in a focus group setting. The amount of time required to complete the recall phase was approximately fifteen minutes.

Participants were specifically instructed not to write down their passphrases after creating them. A week after each participant's first session, he or she was invited back for a second session to repeat the recall process. These recall tasks took approximately fifteen minutes, as before. Participants were asked to return even if they failed to recall one or more passphrases in the first session. Although they would be unlikely to recall a passphrase that was previously forgotten, we still found it useful to measure if their second-session attempt was farther away, the same, or even closer than their first-session attempt.

At the end of the recall phase of the second session, for which the procedure was largely identical to the recall phase of the first session, participants were also tasked with attempting to guess the passphrases of four other accounts as closely as possible. The system treated these adversaries in exactly the same manner as ordinary users:

displaying any mnemonic aids, utilizing the same input masking techniques, and allowing the same five attempts.

After attempting to guess the passphrases of other accounts, participants took on the role of attackers to assess the security of the conditions against shoulder surfing. After observing another user entering his or her passphrase – being allowed during that time to see the mnemonic picture if applicable – the attacker was then asked to enter the observed text as accurately as possible. The amount of time required to complete the guessing and shoulder surfing tasks was approximately fifteen minutes.

In total, the amount of time required to complete both sessions of the study was approximately seventy-five minutes: around forty-five minutes for the first session and around thirty minutes for the second session. To combat attrition, the $20 gift cards were not distributed until the second session was completed.

Results

Data Analysis

We used SPSS 19 for the data analysis. For the recall measures of time, Damerau-Levenshtein distance, Jaro-Winkler proximity, NASA-TLX indices, and SUS score, a repeated measures one-way ANOVA with a 95% confidence interval was used to determine significance. A one-way ANOVA was used for the appropriate dependent variables that were collected only once: creation NASA-TLX, creation SUS, perceived security, overall rating, shoulder surfing and cracking Damerau-Levenshtein distance and Jaro-Winkler proximity, and cracking resist attempts. We compared the combined

means of the control system conditions (#1 and #2) versus the new system conditions (#3 and #4) for recall Damerau-Levenshtein distance and Jaro-Winkler proximity using t-tests as *a priori* planned comparisons, based on our first primary hypothesis.

Welch's correction was used when variances were heterogeneous, which was determined using Levene's test. Mauchly's test was used to check for the assumption of sphericity. When sphericity was violated, the Greenhouse-Geisser correction was applied. Since the recall time did not follow a normal distribution, a log transformation was first applied to achieve normality. We also applied a $\log_2$ transformation to cracking resist attempts to match the scale of entropy bits, in addition to the consideration of normality.

Given significant results, Tukey's HSD test showed which authentication schemes differed significantly from one another. The Games-Howell test was used in place of Tukey's HSD when variances were heterogeneous. A simple effects test was used to evaluate any possible interactions involving condition and session.

Recall success was analyzed using a generalized estimating equation to examine the longitudinal effect across sessions. Shoulder surfing resist success, cracking resist success, and individual session recall success were analyzed using a binary logistic regression.

Machine Cracking

To determine cracking resist attempts, automated cracking was conducted on the passphrases using John the Ripper, considered one of the most well regarded

password cracking programs (Rao, Jha, & Kini, 2013). John the Ripper's wordlist mode was used with the largest dictionary provided by the developers. After eight hours of cracking on a 3.3 GHz Intel Core i5 desktop computer, not a single passphrase was identified.

We then tried John the Ripper's incremental mode, which uses a Brute-Force Markov (BFM) algorithm, as outlined in Kelley et al. (2011). The algorithm was trained with the same wordlist, which was used to build a Markov chain (with states and probabilistic transitions to subsequent states) for the password space. This Markov chain was then followed while cracking. After another eight hours of cracking on the same 3.3 GHz Intel Core i5 computer, again, not a single passphrase was identified. Unsurprisingly, John the Ripper and all the other password crackers currently available are known to perform poorly against passphrases (Rao, Jha, & Kini, 2013).

Finally, we built a simple deterministic passphrase cracker based on word frequency analysis, trained on the wordlist derived from the study. This passphrase cracker will always first attempt the combination of all the most common words for each position, and then proceed to guess the next most likely combination and repeat until the entire passphrase space is exhausted.

For example, if a total of three users selected *I love cats*, *I hate cats*, and *I love dogs* as their passphrases, the deterministic cracker would first attempt *I love cats* followed by *I love dogs*, *I hate cats*, and *I hate dogs*. The total number of possible passphrases in this simplified example is $1 \times 2 \times 2 = 4$, and the cracker would attempt

all four possibilities in descending likelihood. The specific order is determined because *love* appears more frequently than *hate*, and *cats* appears more frequently than *dogs*. Because certain words appear more frequently, passphrases containing these words would be considered less secure and would take fewer attempts to crack. Moreover, a set of passphrases would be considered less secure, the more overlap there is between individual passphrases.

Typical password crackers are designed to operate on password hashes since normally the passwords are unknown. On the other hand, our deterministic passphrase cracker can be fed the passphrases and simply calculate the required number of attempts without having to iterate through all the individual combinations. At this point, the cracking resist attempts could be determined after several minutes of running on the 3.3 GHz Intel Core i5 computer.

Objective Measures

Efficiency. The results indicate a significant difference for first-session recall time ($F=8.99$, $p<.001$, $\eta^2=.087$). The recall time in seconds was shorter for Special English ($M=28.9$) than all other conditions, as seen in Figure 25. For the second session, there was no significant difference for recall time ($F=1.88$, $p=.138$, $\eta^2=.025$). The average recall time in seconds for Special English increased from before ($M=44.9$) but remained the shortest condition. There was no interaction effect of condition and session on recall time ($F=1.23$, $p=.299$, $\eta^2=.018$).

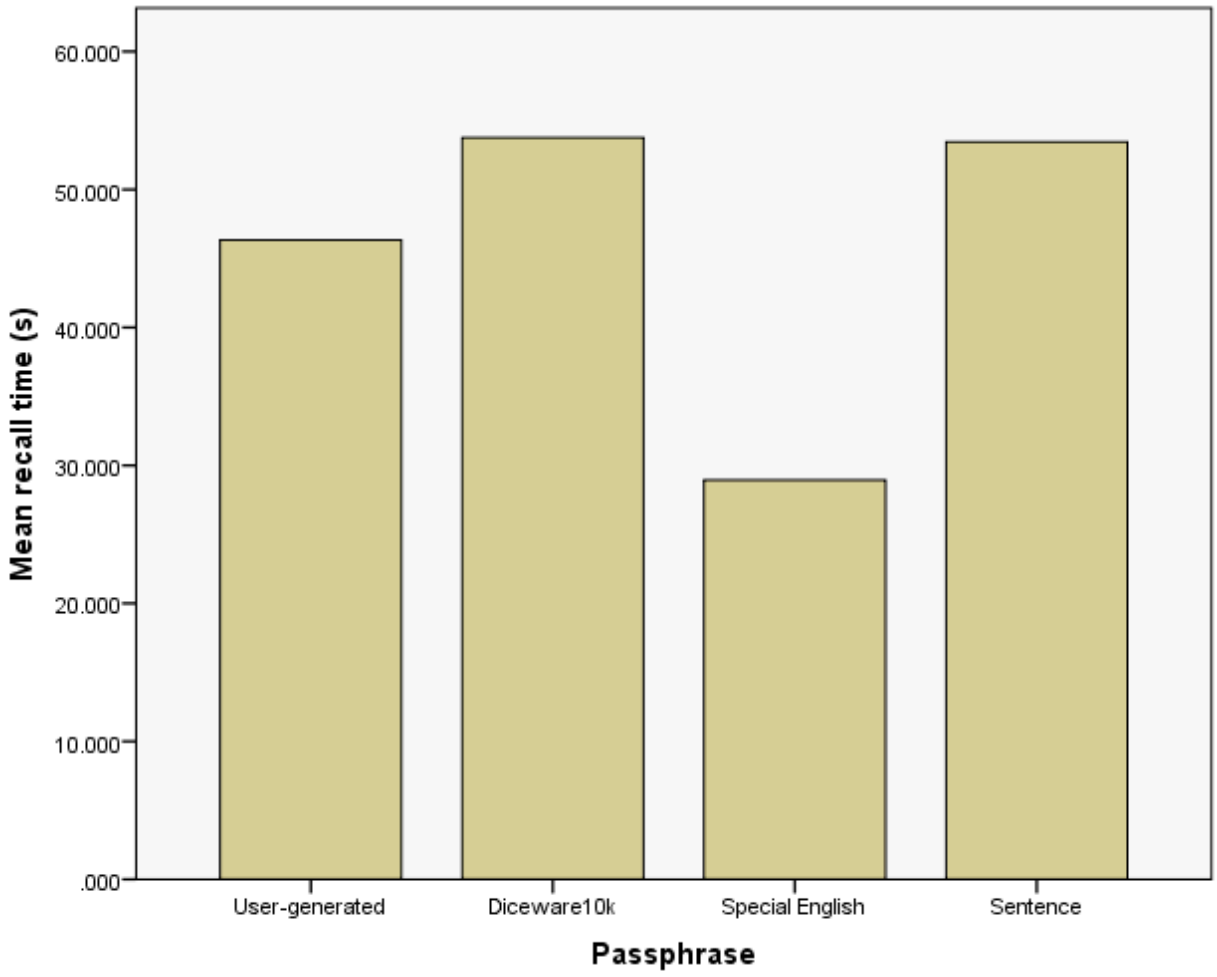

Figure 6.5. 1st-session recall time (smaller values indicate increased efficiency)

Effectiveness. The differences in recall Damerau-Levenshtein distance were statistically significant for both the first session ($F=4.95$, $p=.003$, $\eta^2=.045$) and second session ($F=6.26$, $p=.001$, $\eta^2=.077$). However, the interaction effect of condition and session on Damerau-Levenshtein distance was not significant ($F=0.940$, $p=.422$, $\eta^2=.014$). For the first session, Special English ($M=0.380$) was significantly shorter than Diceware10k ($M=3.23$), as shown in Figure 26. Although the mean of User-generated

(M=3.50) was even higher than that of Diceware10k, the greater variance in the User-generated condition meant that it did not differ significantly from Special English. We did observe a significant effect (t=3.05, p=.003, d=.424) when comparing the new system conditions (Special English and Sentence: M=0.740) with the control system conditions (User-generated and Diceware10k: M=3.37). For the second session, Special English (M=2.12) was significantly shorter than both User-generated (M=7.00) and Diceware10k (M=7.62). Sentence (M=2.94) was also significantly shorter than Diceware10k. These results can be seen in Figure 27. We also found a significant effect (t=4.02, p<.001, d=.568) when comparing the new system conditions (M=2.53) with the control system conditions (M=7.31).

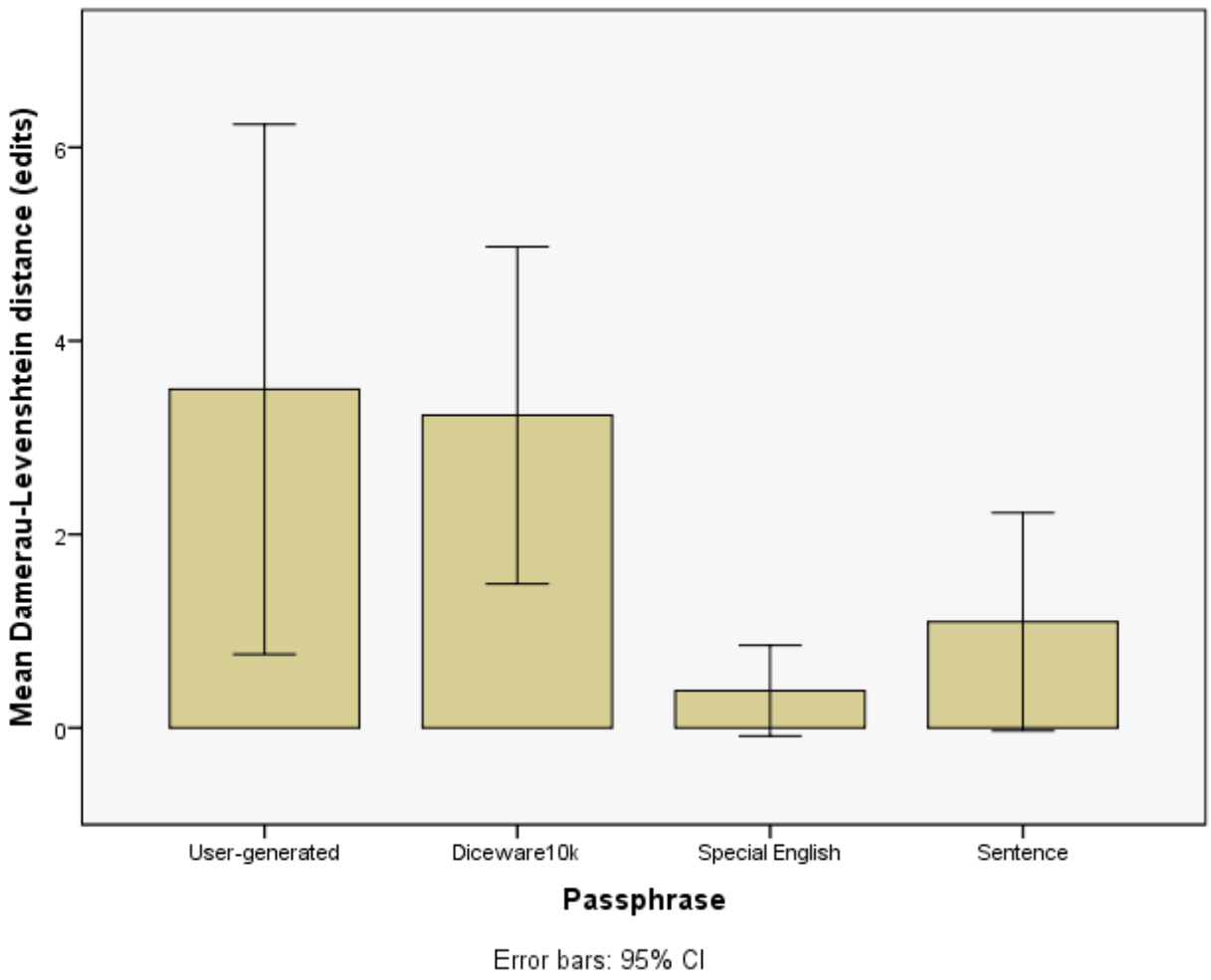

Figure 6.6. 1st-session Damerau-Levenshtein distance (smaller values indicate increased effectiveness)

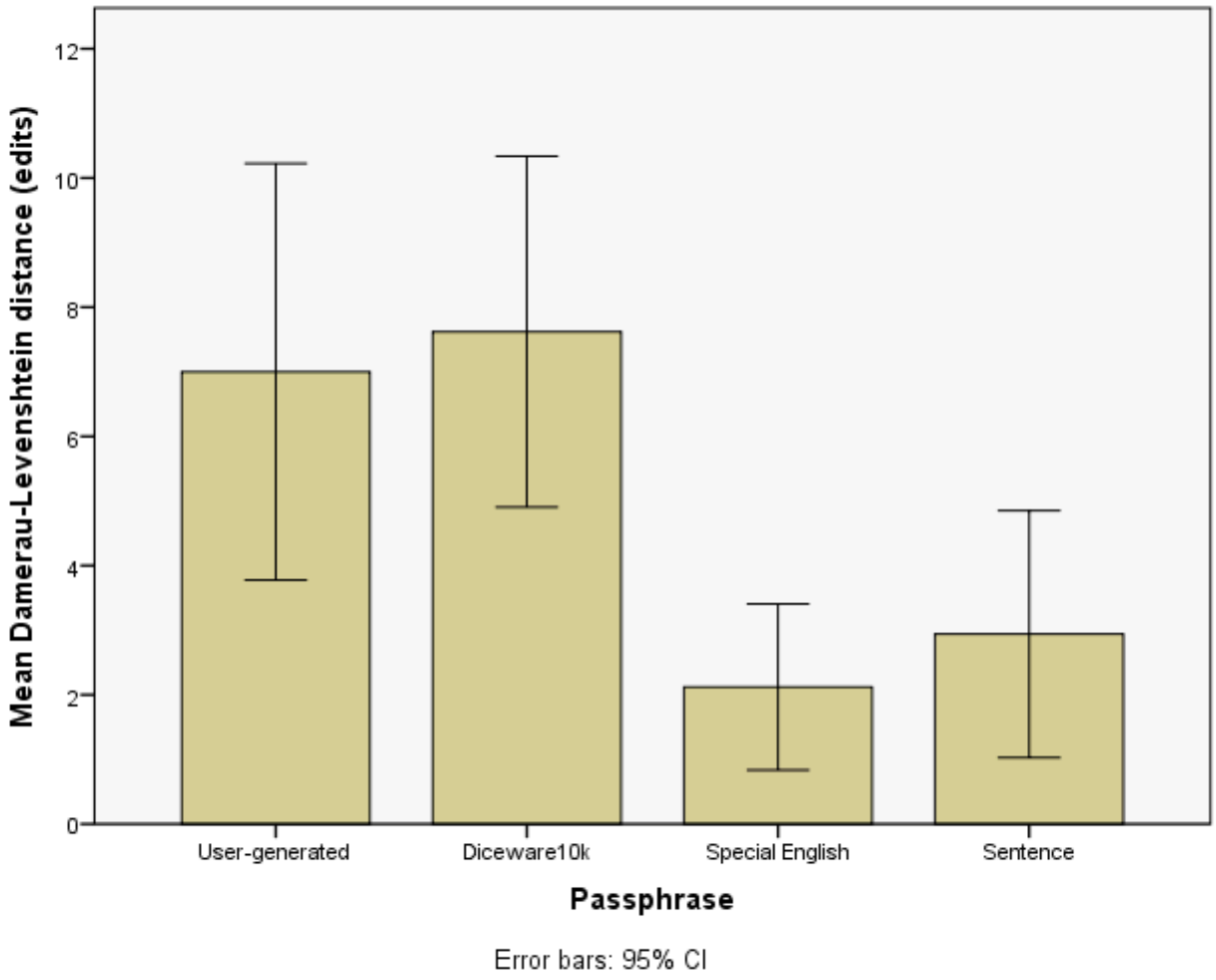

Figure 6.7. 2nd-session Damerau-Levenshtein distance (smaller values indicate increased effectiveness)

Likewise, the differences in recall Jaro-Winkler proximity were statistically significant for both the first session ($F=4.16$, $p=.008$, $\eta^2=.059$) and second session ($F=9.96$, $p<.001$, $\eta^2=.147$). Recall Jaro-Winkler proximity was strongly correlated with Damerau-Levenshtein distance for both the first session ($r=-.834$, $p<.001$) and second session ($r=-.834$, $p<.001$). Again, the interaction effect of condition and session on Jaro-Winkler proximity was not significant ($F=3.026$, $p=.079$, $\eta^2=.044$). For the first session,

Special English (M=.990) and Sentence (M=.985) were both closer than Diceware10k (M=.898), as seen in Figure 28. We also found a significant effect (t=-3.47, p=.001, d=.481) when comparing the new system conditions (M=.988) with the control system conditions (M=.911). For the second session, Special English (M=.956) and Sentence (M=.955) were again both closer than Diceware10k (M=.764), as seen in Figure 29. There was also a significant effect (t=-5.23, p<.001, d=.739) when comparing the new system conditions (M=.956) with the control system conditions (M=.811).

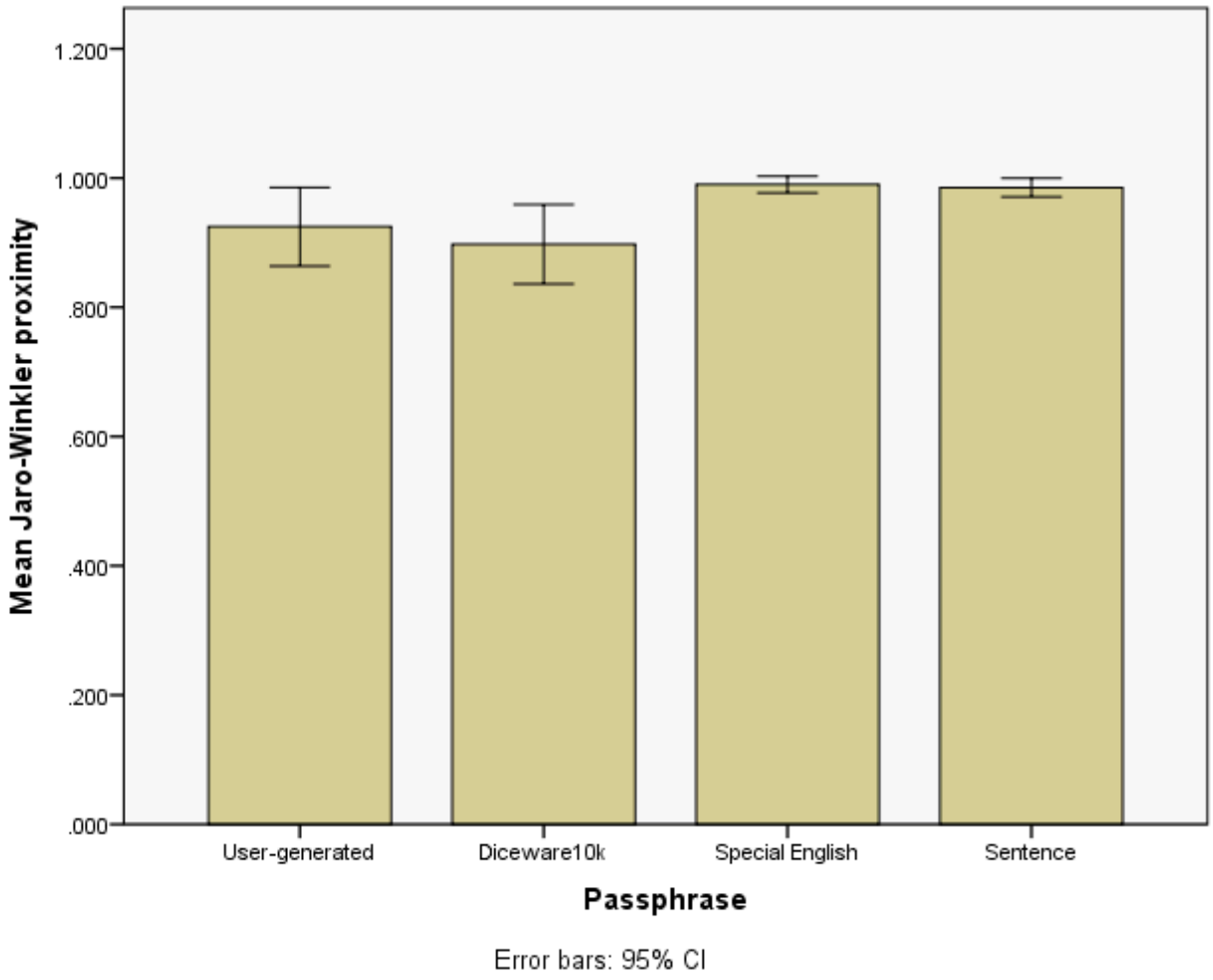

Figure 6.8. 1st-session Jaro-Winkler proximity (larger values indicate increased effectiveness)

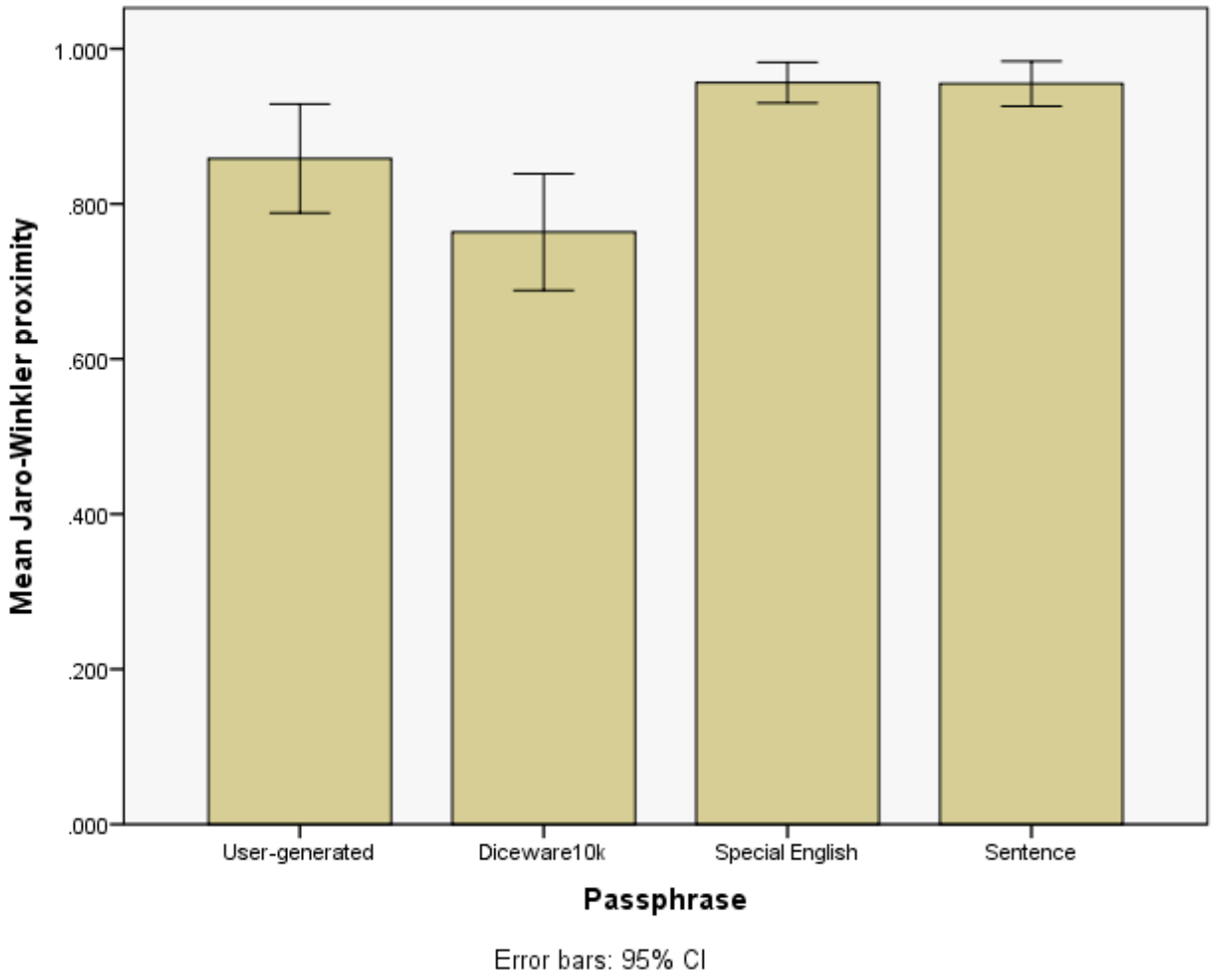

Figure 6.9. 2nd-session Jaro-Winkler proximity (larger values indicate increased effectiveness)

As with the other recall effectiveness measures, the effect of passphrase on success was also determined to be statistically significant for both the first session (χ^2=20.4, p<.001) and second session (χ^2=36.0, p<.001). There was no significant interaction effect of condition and session on recall success (χ^2=1.19, p=.275). Figure 30 shows that the first-session success rate was 94.2% for Special English, 92.3% for Sentence, 76.9% for User-generated, and 65.4% for Diceware10k. As seen in Figure 31,

the second-session success rate was 82.0% for Sentence, 80.0% for Special English, 50.0% for User-generated, and 34.0% for Diceware10k. Significance was also found when examining system type (control or new) instead of condition for both the first session (χ^2=18.5, p<.001) and the second session (χ^2=33.3, p<.001).

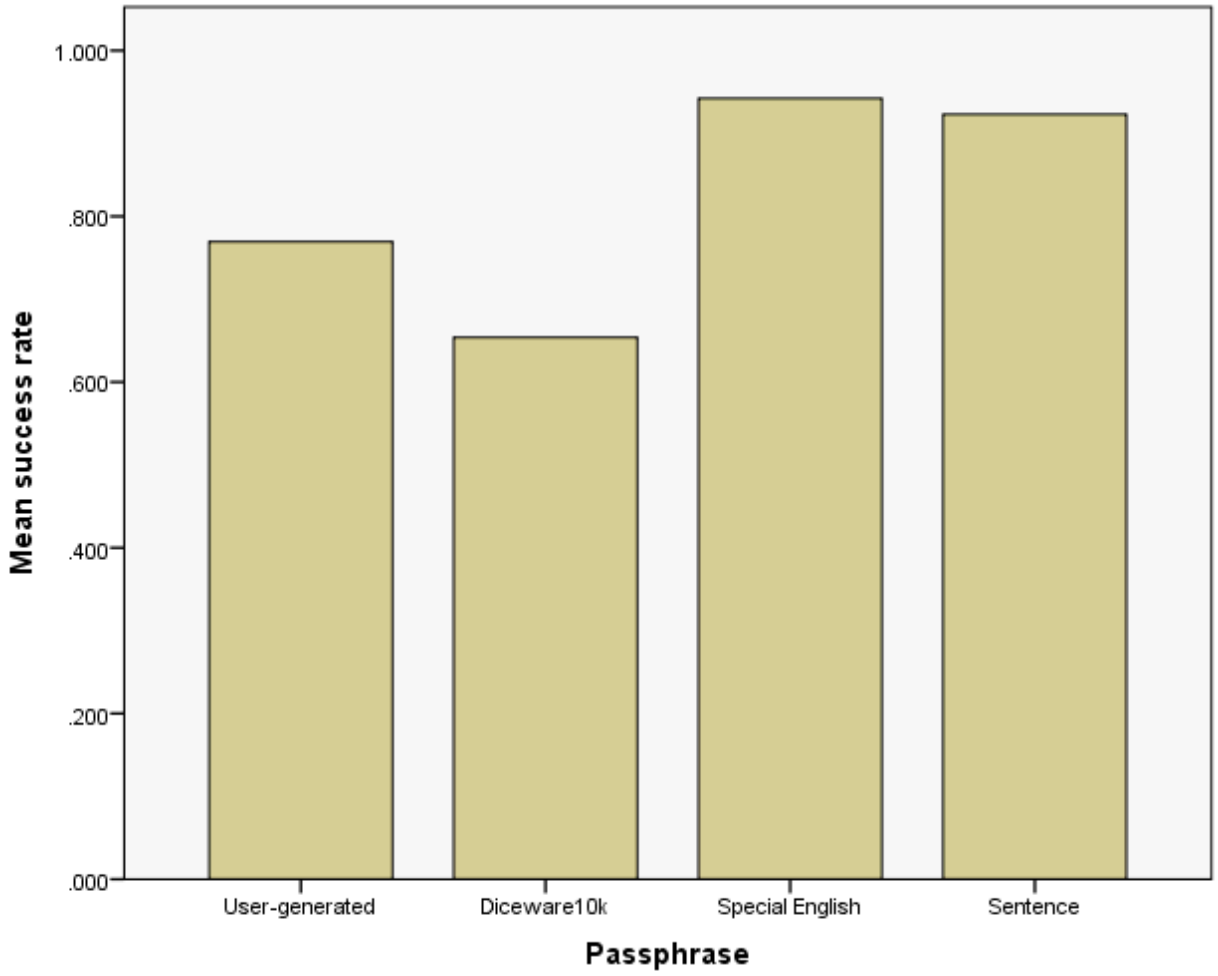

Figure 6.10. 1st-session recall success rate (larger values indicate increased effectiveness)

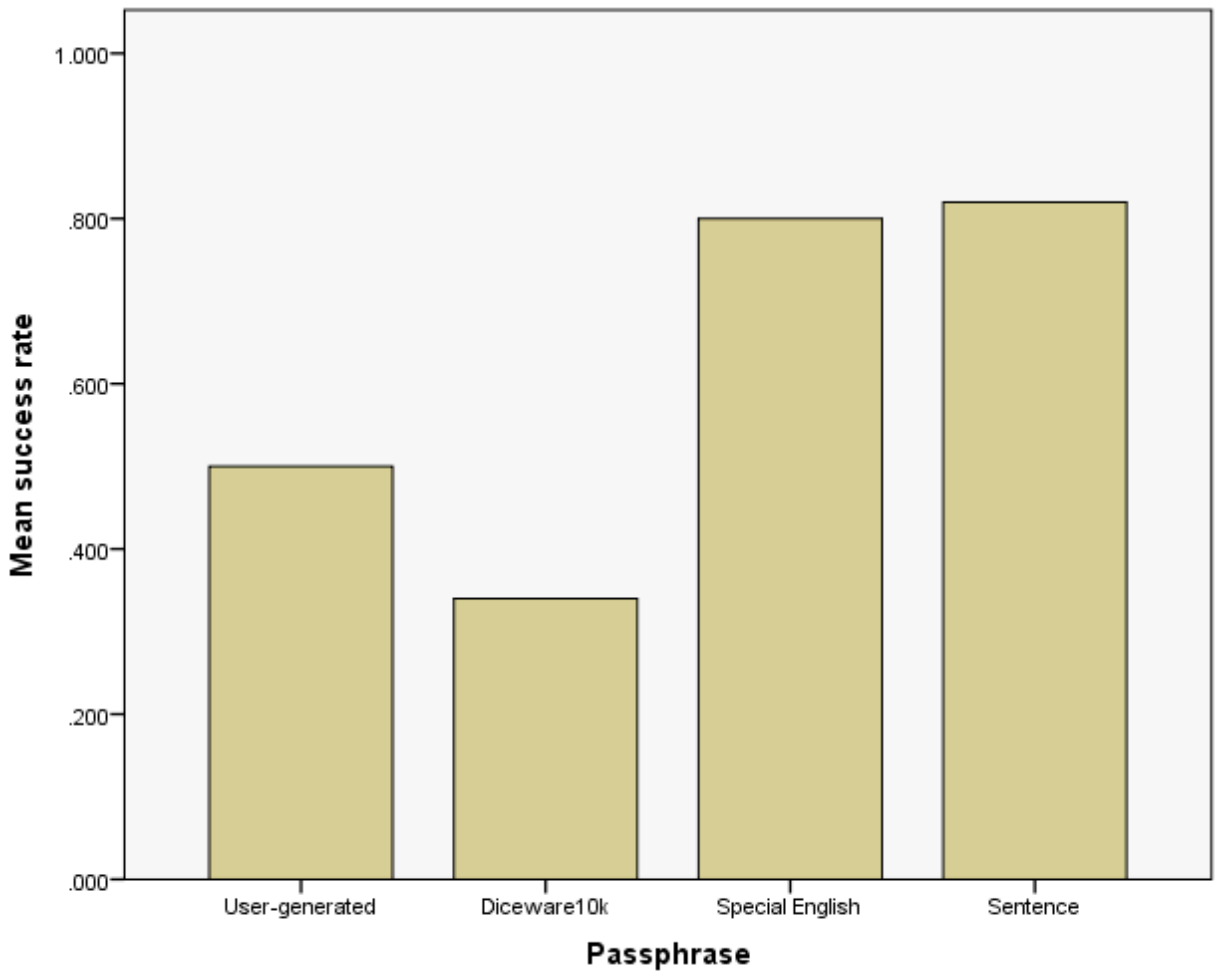

Figure 6.11. 2nd-session recall success rate (larger values indicate increased effectiveness)

Security. There was a statistically significant difference for the Damerau-Levenshtein distance between the shoulder surfers' guesses and the actual passwords ($F=43.0$, $p<.001$, $\eta^2=.397$). As shown in Figure 32, Sentence (M=30.5) was longer than User-generated (M=23.0) and Special English (M=19.9), and all three conditions were longer than Diceware10k (M=9.68). It is worth emphasizing that longer passphrases naturally result in larger Damerau-Levenshtein distances when the guesses are

unrelated. When looking instead at Jaro-Winkler proximity, as seen in Figure 33, there was no significant effect ($F=2.14$, $p=.097$, $\eta^2=.032$). However, Jaro-Winkler proximity did correlate with Damerau-Levenshtein distance ($r=-.518$, $p<.001$). No passphrase was guessed correctly, so the shoulder surfing resist success rate was 100% for all conditions.

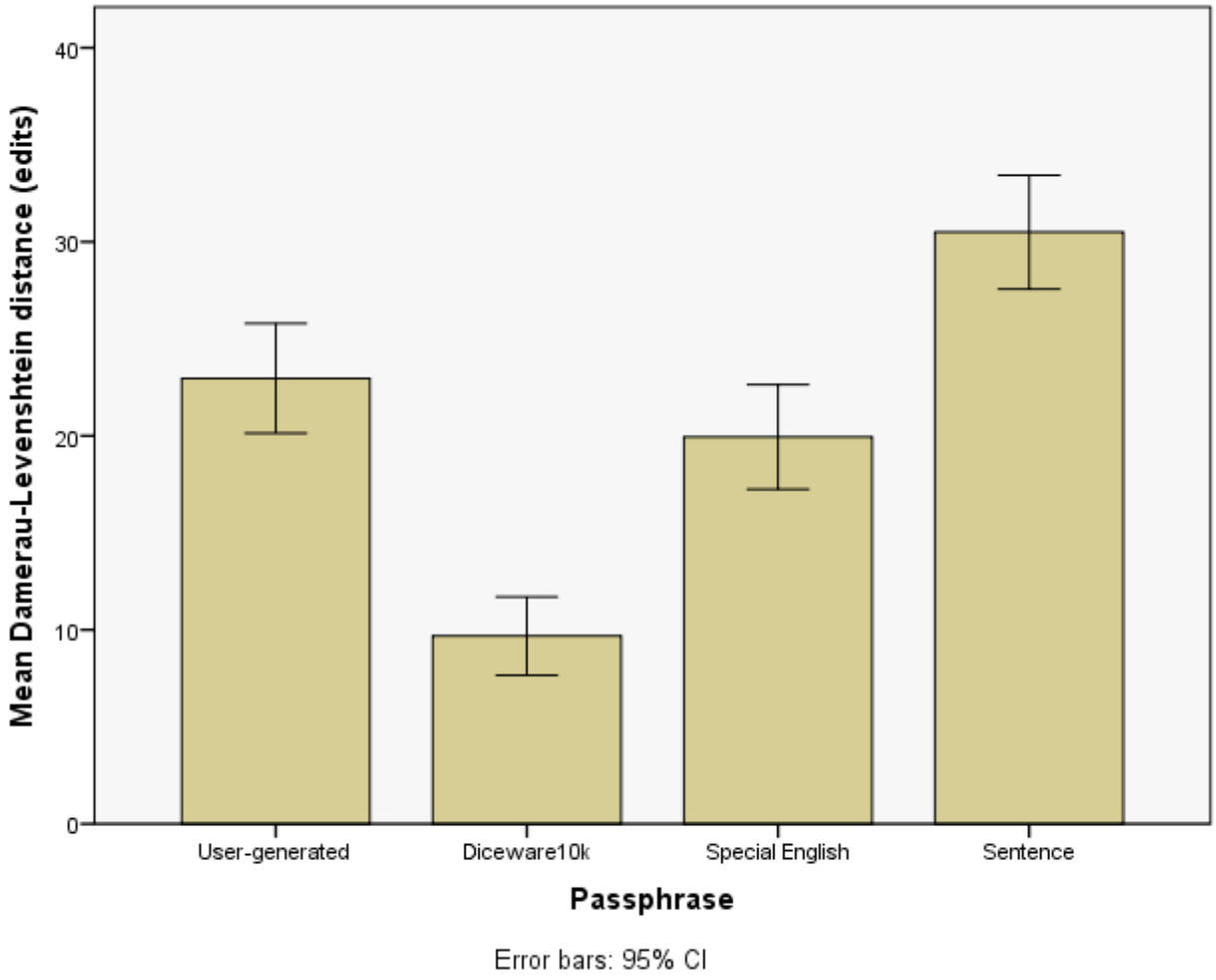

Figure 6.12. Shoulder surfing Damerau-Levenshtein distance (larger values indicate increased security)

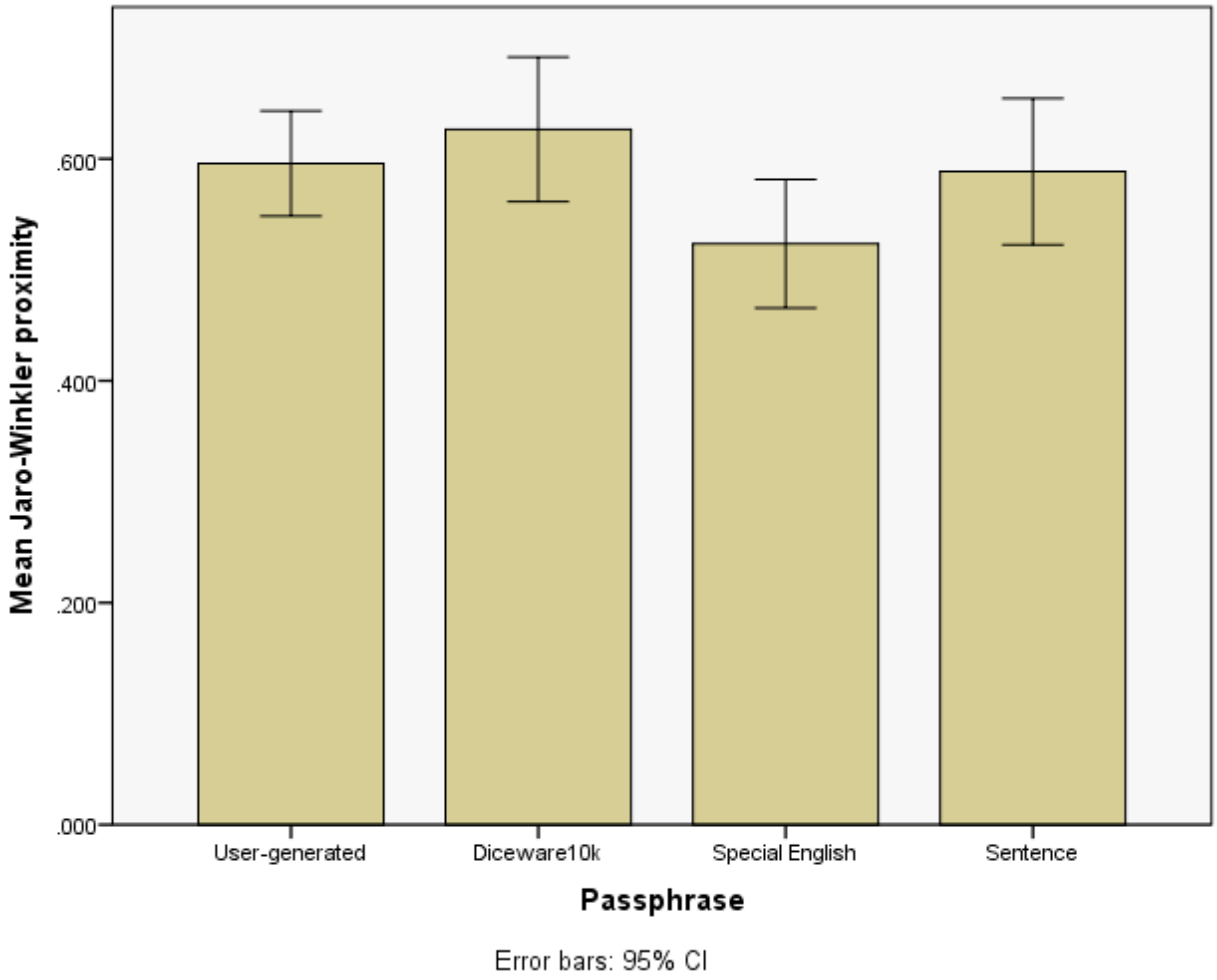

Figure 6.13. Shoulder surfing Jaro-Winkler proximity (smaller values indicate increased security)

For cracking security against human attackers, there was a statistically significant difference for the Damerau-Levenshtein distance between adversaries' guesses and the actual passwords ($F=43.4$, $p<.001$, $\eta^2=.399$). As seen in Figure 34, Sentence ($M=31.5$) and User-generated ($M=28.3$) were both farther than Special English ($M=20.8$), and all three conditions were farther than Diceware10k ($M=16.2$). There was also a statistically

significant difference for the Jaro-Winkler proximity, which correlated with Damerau-Levenshtein distance (r=-.518, p<.001), between guesses and the actual passwords (F=19.5, p<.001, η^2=.230). For Jaro-Winkler proximity, Diceware10k (M=.404) and User-generated (M=.457) were farther than both Special English (M=.568) and Sentence (M=.589), as seen in Figure 35. Again, no passphrase was guessed correctly, so the cracking resist success rate was 100% for all passphrases.

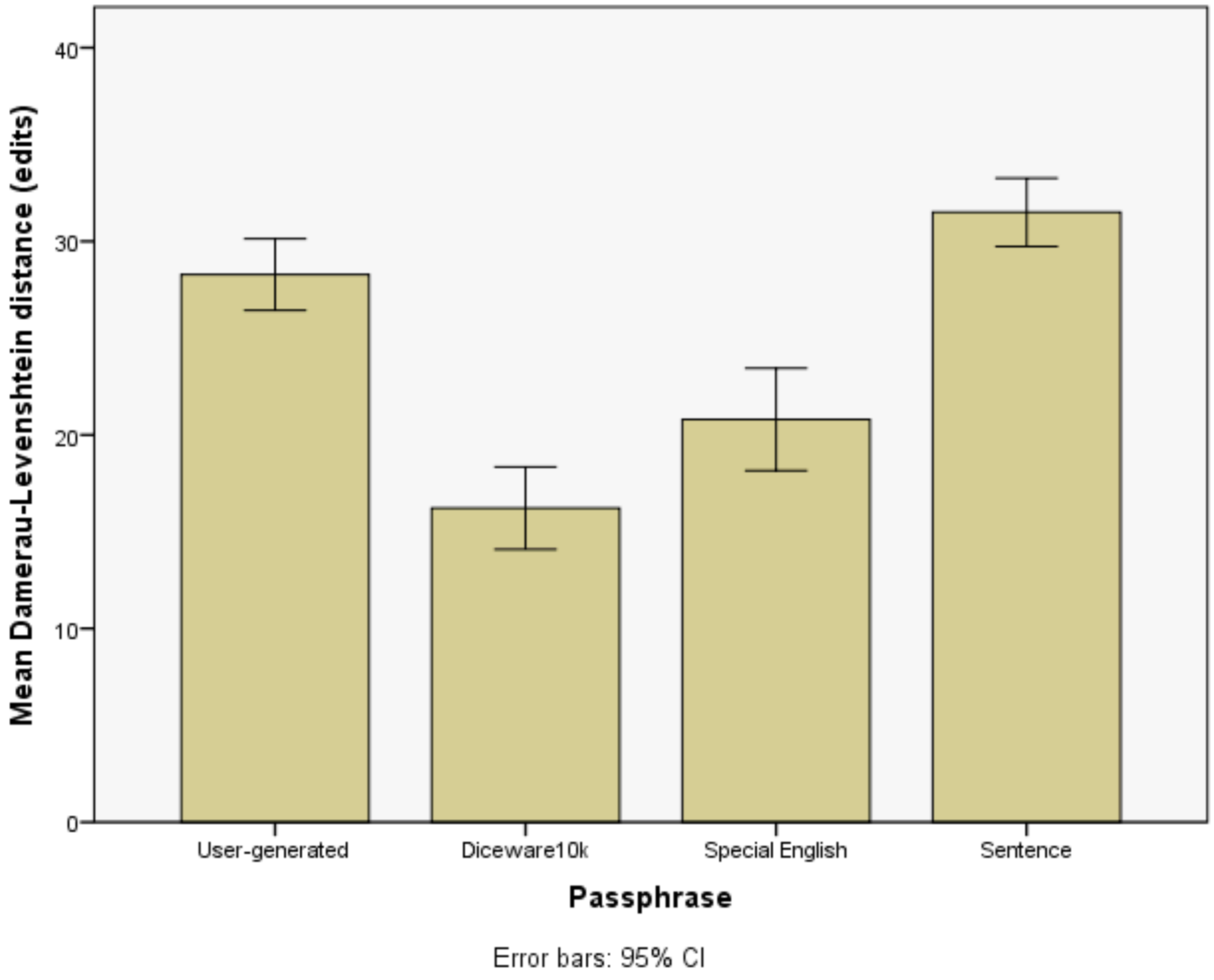

Figure 6.14. Cracking Damerau-Levenshtein distance (larger values indicate increased security)

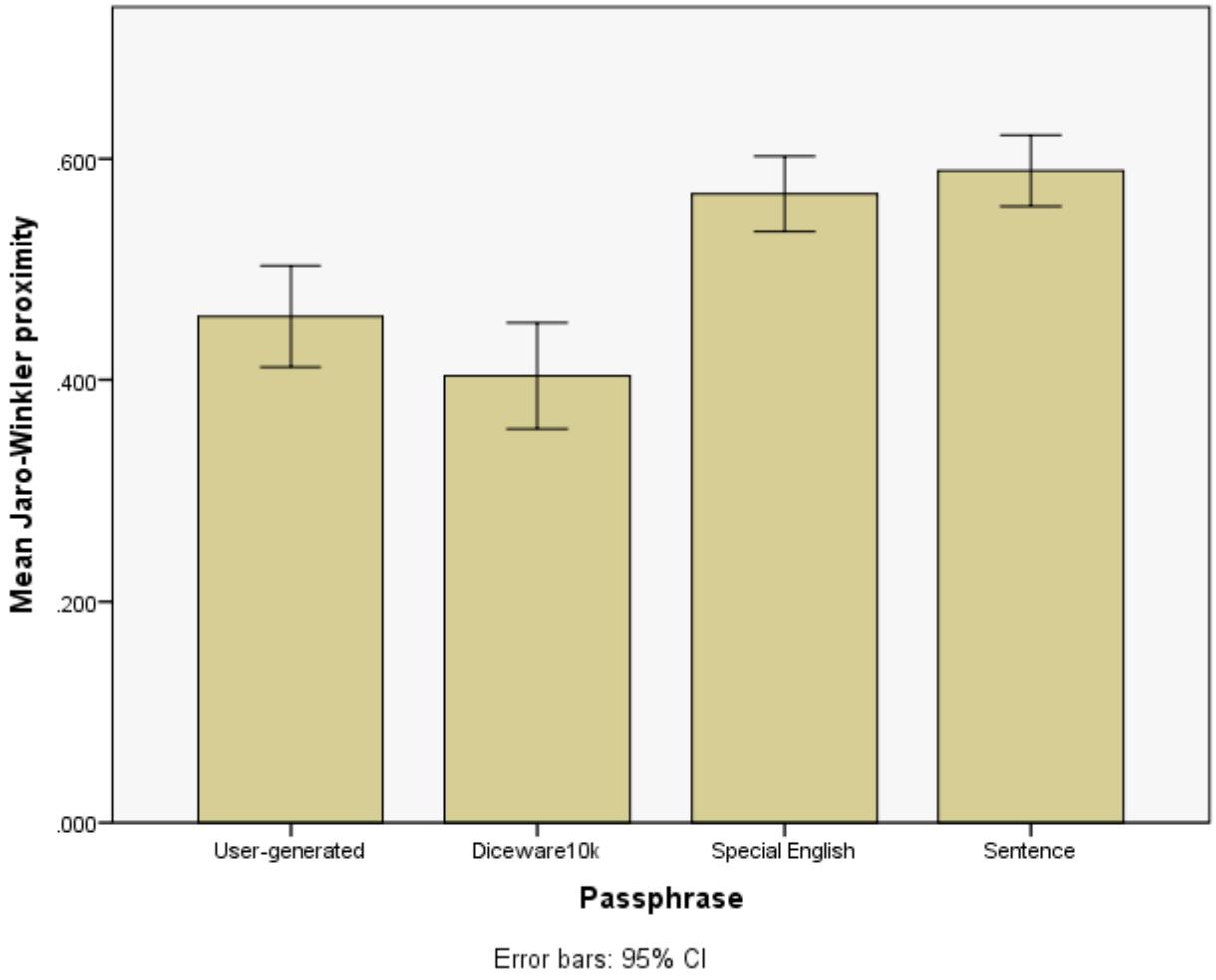

Figure 6.15. Cracking Jaro-Winkler proximity (smaller values indicate increased security)

For cracking security against machine attackers, the effect of passphrase on the $\log_2$-transformed cracking resist attempts was found to be statistically significant ($F=78.1$, $p<.001$, $\eta^2=.632$). Special English (M=36.0), Sentence (M=34.6), and Diceware10k (M=34.4), as the system-generated conditions, all took a greater number

of attempts to crack on average than did the User-generated condition (M=23.8). This effect can be seen in Figure 36.

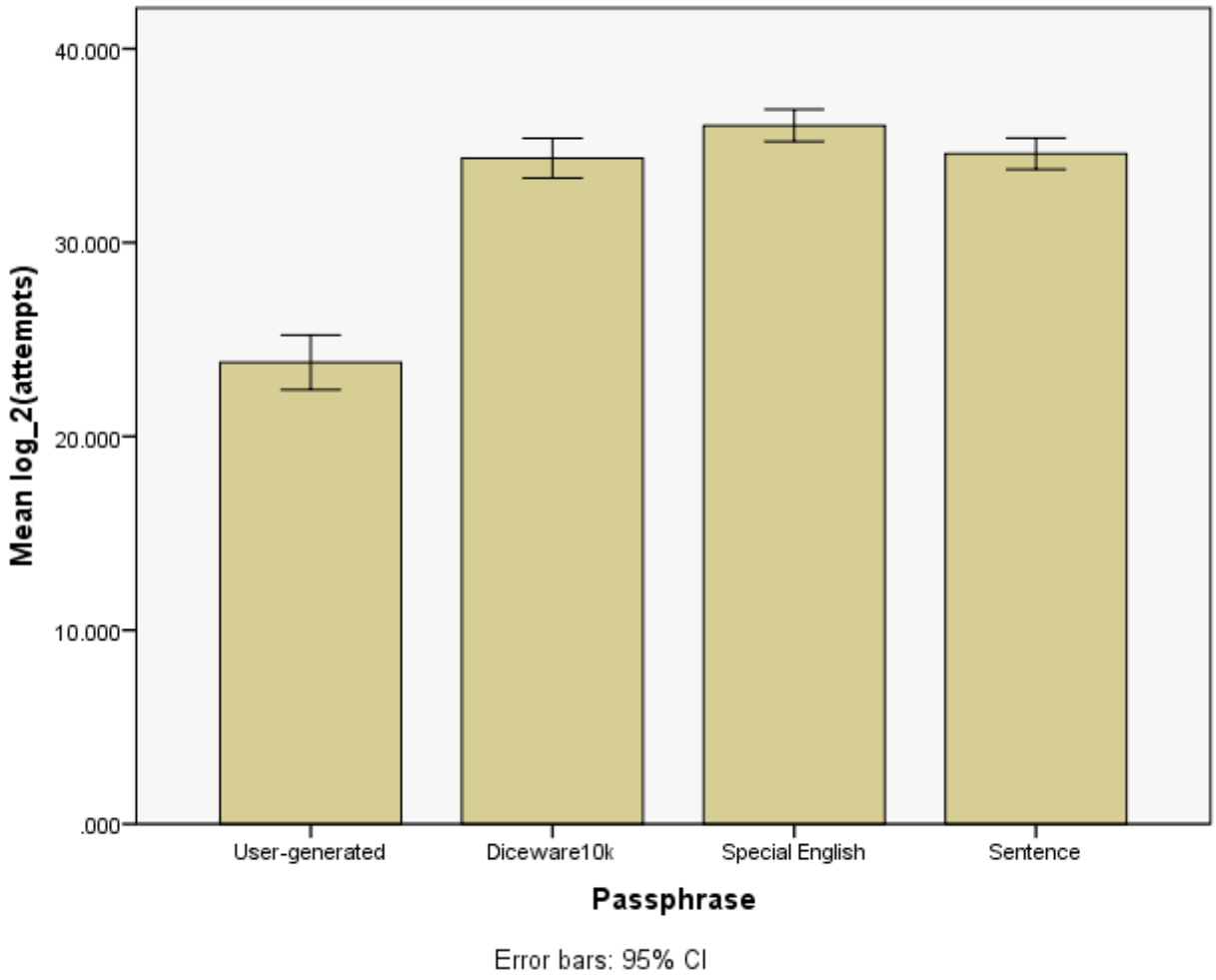

Figure 6.16. $\log_2$-transformed cracking resist attempts (larger values indicate increased security)

Subjective Measures

Workload. For the creation task, there was no significant effect of condition on any of the NASA-TLX indices: mental demand ($F=1.37$, $p=.253$, $\eta^2=.020$), physical demand ($F=2.39$, $p=.070$, $\eta^2=.034$), temporal demand ($F=2.40$, $p=.069$, $\eta^2=.034$), performance ($F=1.31$, $p=.272$, $\eta^2=.019$), effort ($F=2.04$, $p=.109$, $\eta^2=.029$), and frustration ($F=0.399$, $p=.754$, $\eta^2=.006$). During the first recall session, the only index that was

statistically significant was mental demand (F=3.44, p=.019, η^2=.049). Special English (M=2.29) was rated as being less mentally demanding than Diceware10k (M=3.60). For the second recall session, there was no significant effect of condition on any of the indices, including mental demand (F=2.178, p=.092, η^2=.032), which was now rated at M=3.44 for Special English. In addition, there was no interaction effect of condition and session on any of the NASA-TLX indices.

Usability. During the creation task, there was no significant effect of condition on SUS (F=1.50, p=.216, η^2=.022), as seen in Figure 37. For first-session recall, the effect of condition on SUS was significant (F=4.29, p=.007, η^2=.058). Figure 38 shows that Special English (M=72.9) was rated as more usable than Diceware10k (M=59.9) and User-generated (M=58.0). As seen in Figure 39, the effect of condition on SUS was also significant for second-session recall (F=7.62, p<.001, η^2=.110). Sentence (M=76.0) and Special English (M=71.7) were both considered more usable than User-generated (M=56.0). Sentence was also found to be significantly more usable than Diceware10k (M=62.1). There was no interaction effect of condition and session on SUS (F=1.652, p=.179, η^2=.025).

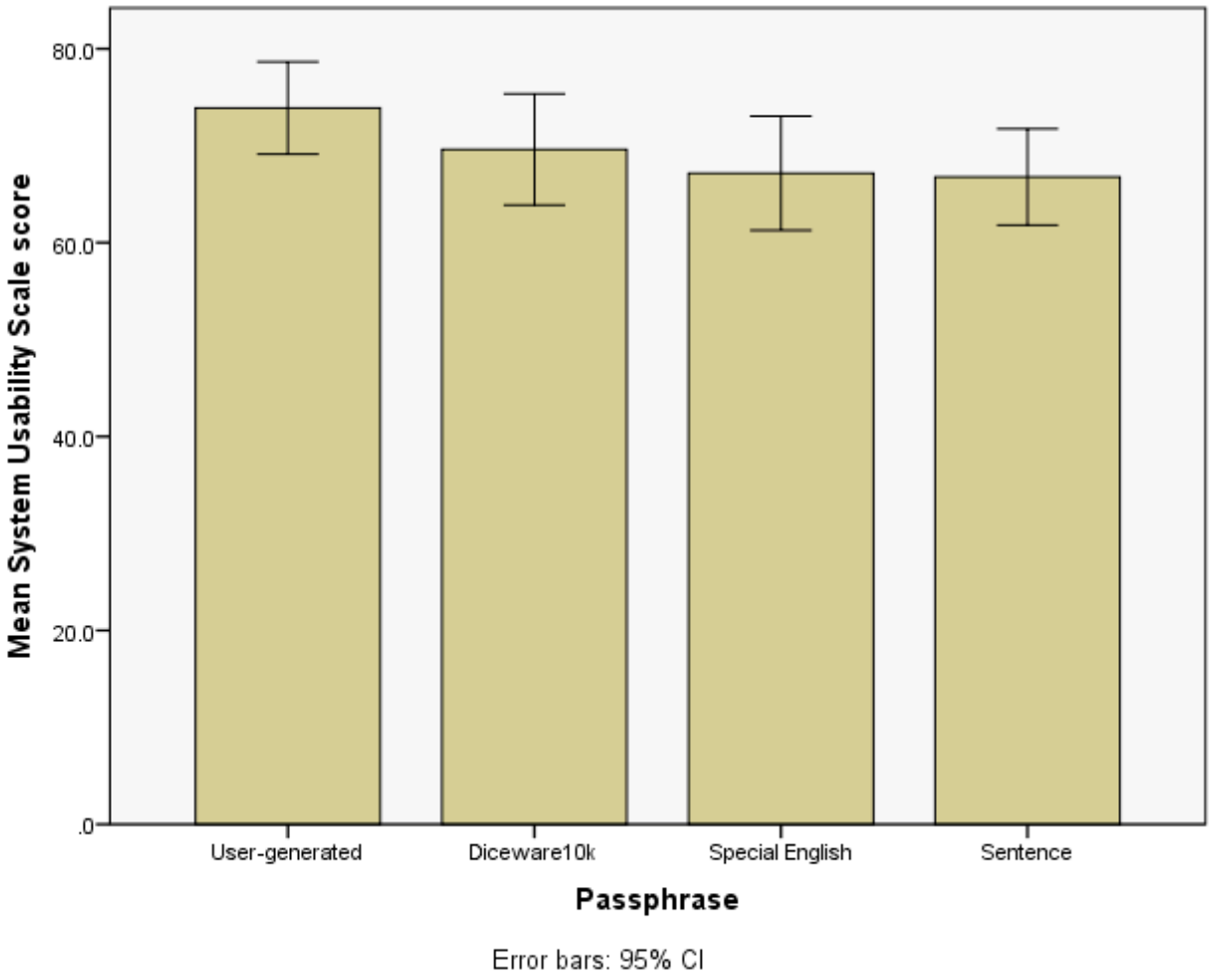

Figure 6.17. Creation SUS score (larger values indicate increased usability)

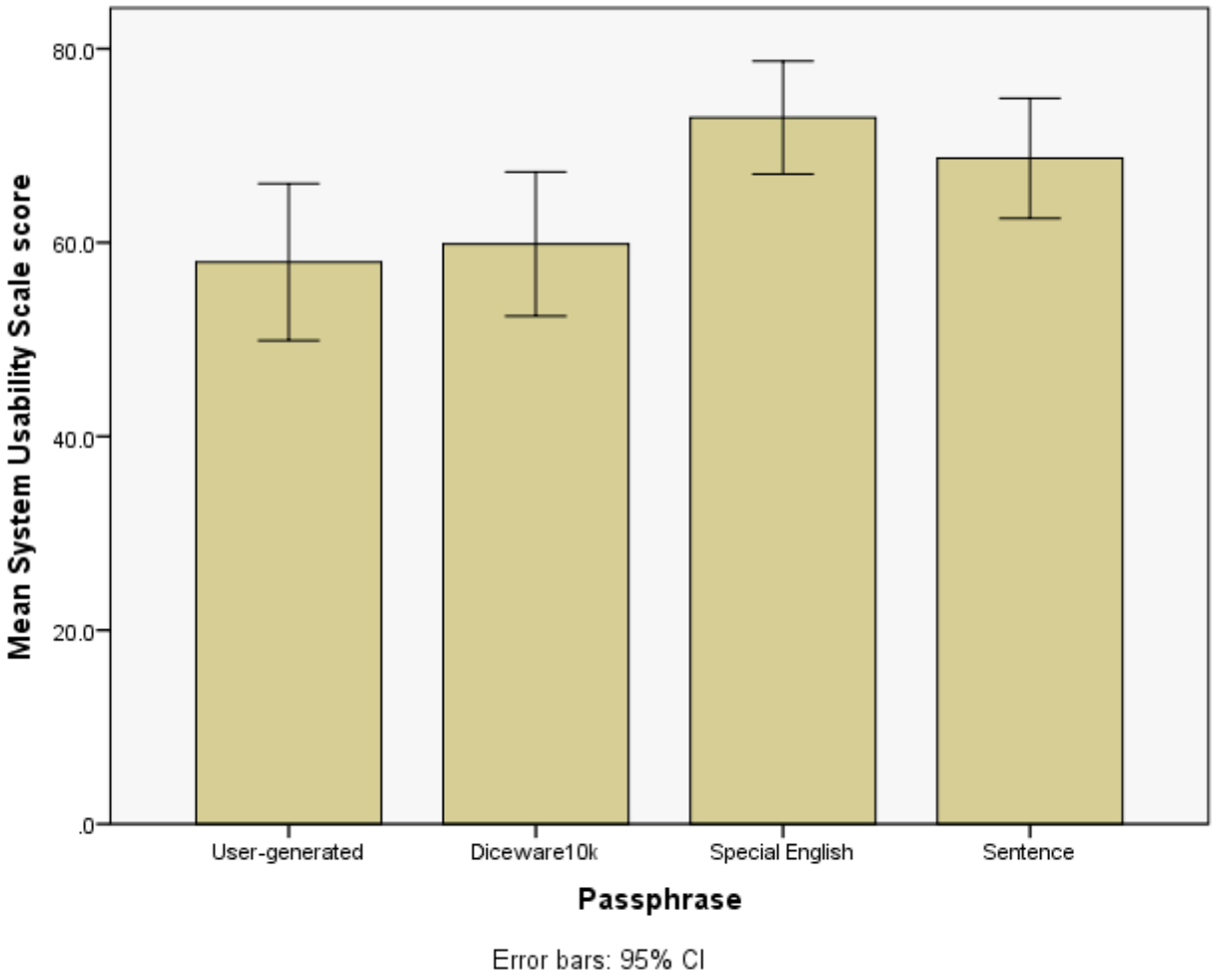

Figure 6.18. 1st-session recall SUS score (larger values indicate increased usability)

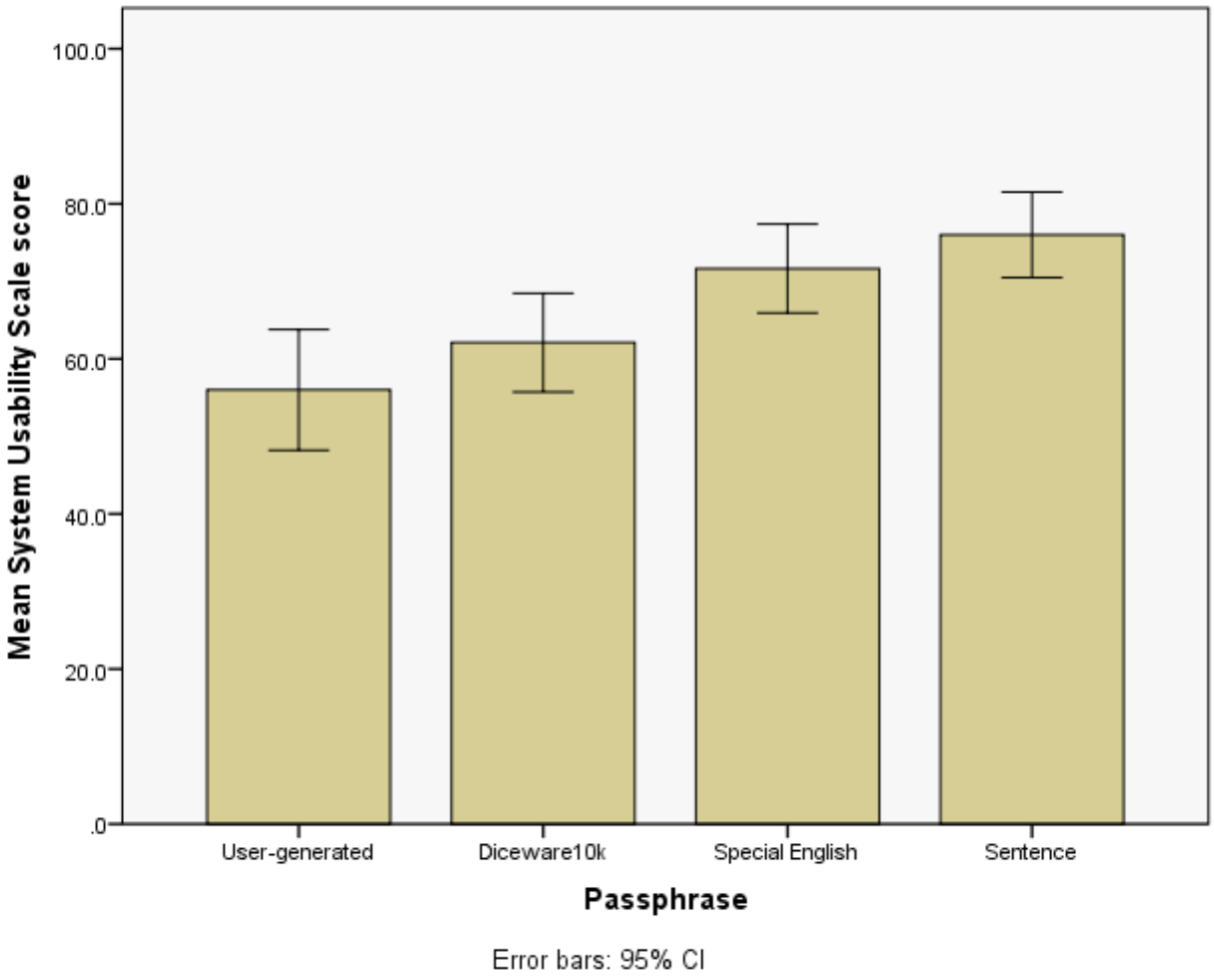

Figure 6.19. 2nd-session recall SUS score (larger values indicate increased usability)

Security. There was no significant difference for the effect of condition on perceived level of security ($F=0.638$, $p=.591$, $\eta^2=.010$), as seen in Figure 40.

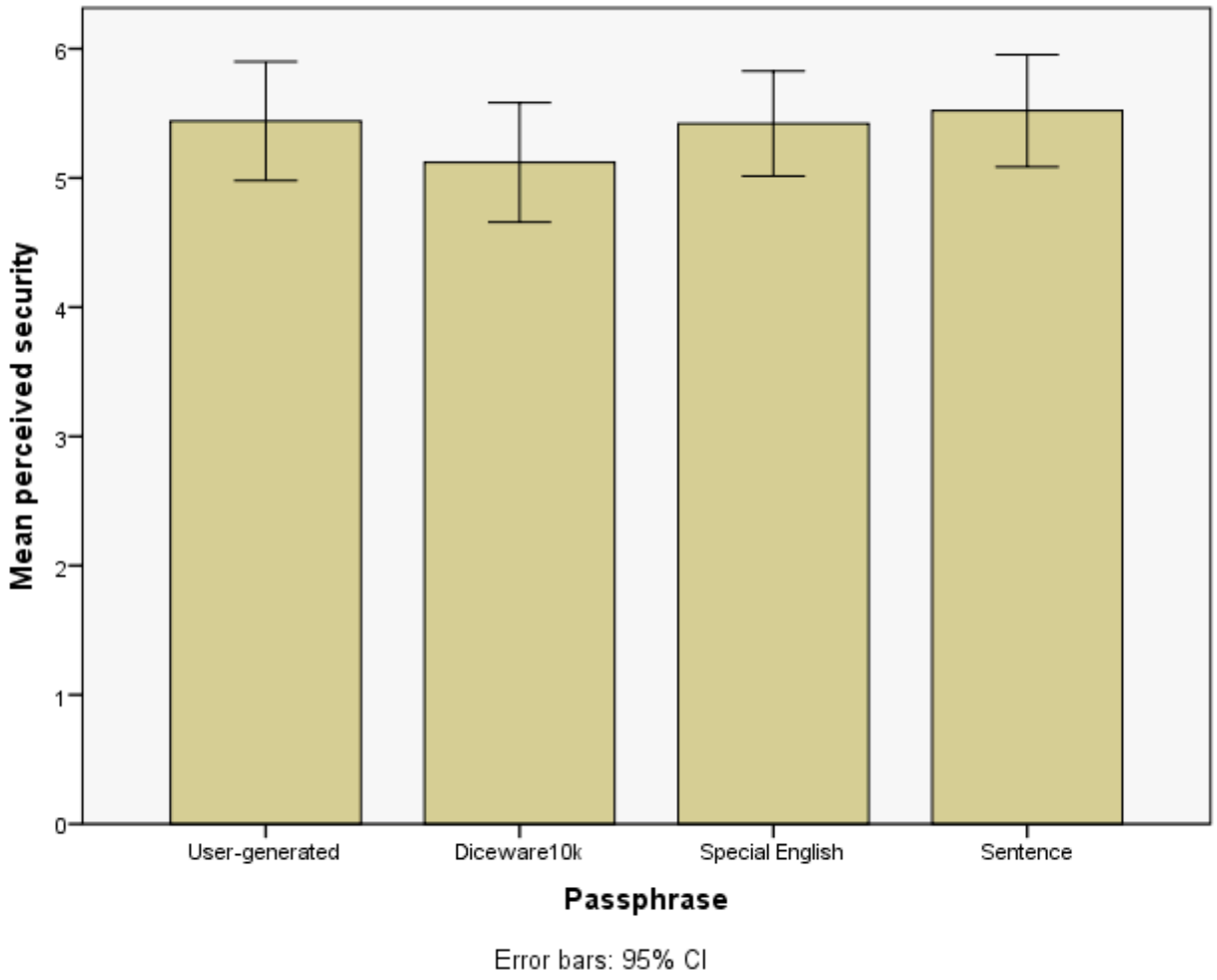

Figure 6.20. Perceived security on a 7-point Likert scale (larger values indicate increased security)

Overall. There was a significant difference for the effect of condition on overall rating ($F=9.73$, $p<.001$, $\eta^2=.130$), as seen in Figure 41. Special English ($M=5.02$), Sentence ($M=4.98$), and User-generated ($M=4.88$) were all rated more highly overall than was Diceware10k ($M=3.52$).

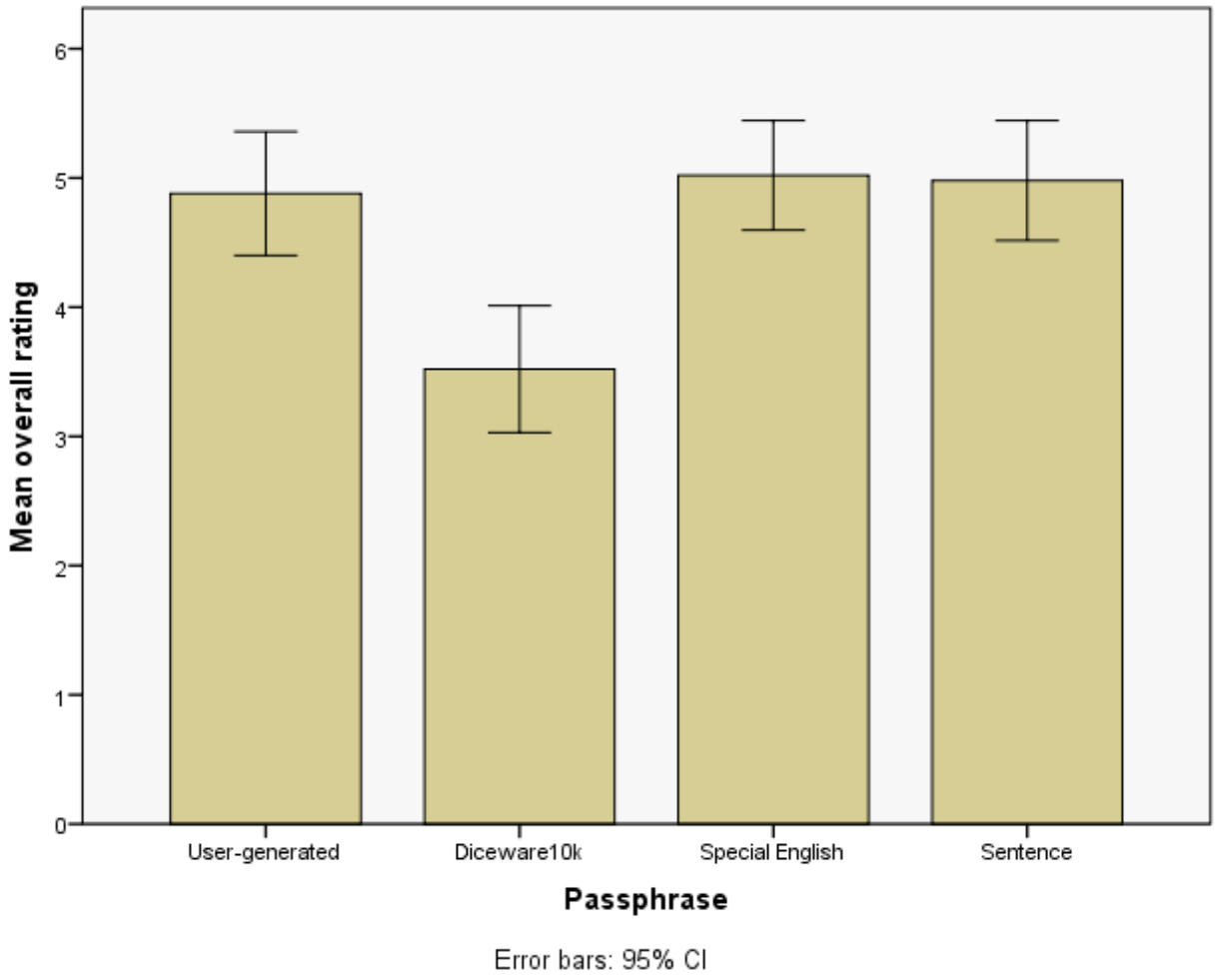

Figure 6.21. Overall rating on a 7-point Likert scale (larger values indicate increased usability and security)

Discussion

Hypotheses

Our first primary hypothesis, that recall effectiveness measures would be most favorable for the new system conditions, was supported by the results. This was the case when comparing between conditions and also when comparing control and new system conditions. As a whole, the second-session recall success rates (82% for

Sentence, 80% for Special English, 50% for User-generated, and 34% for Diceware10k) were somewhat higher than expected, given our results from previous studies in Chapters 3 and 4 on passwords and mnemonics.

Nevertheless, success rates around 80% would not initially seem to inspire tremendous confidence in the practicality of real-world implementation. However, for the sake of experimental design, we intended for our tasks to be more difficult than in typical scenarios. Participants were tasked to create four accounts in a row (resulting in interference between passphrases), could not write down or practice their passphrases, and were not incentivized for higher performance. Therefore, we believe that in standard real-world situations, long-term success rates for our new system would increase past the 80% level.

For the control system conditions, our finding that the success rates were somewhat higher than expected was likely due to the controls in this study being a bit more memorable than the previous control condition of passwords generated with random letters. For the new system conditions, higher success rates were likely due in large part to the login benefits of error correction and decoy masking. There were also fewer words to remember with the new system conditions than with the previous 7-word sentence structure.

Our second primary hypothesis, that shoulder surfing security measures would not differ between authentication schemes, was also supported by the results. Although there were significant effects as measured by Damerau-Levenshtein distance, this was

largely due to the differences in length between passphrases. (Regardless, the fact that our new system results in longer passphrases is a valid contributor to increased security against shoulder surfing combined with machine attack.) Because of the need for a length-independent baseline, Jaro-Winkler proximity was a more useful measure, and it showed no significant differences between conditions. This indicates that the shoulder surfing security of decoy masking was preserved through the transition from passwords to passphrases.

Our third primary hypothesis was that cracking security measures would be least favorable overall for User-generated, which we argue was also borne out by the results. For cracking security against human attackers, the Jaro-Winkler proximity indicated that User-generated and Diceware10k performed better than Special English and Sentence; in other words, the mnemonic pictures leaked some information to attackers. However, the average proximity for all conditions fell between .4 and .6. These values are not very high, and the new system conditions were indeed far closer to the guesses of the control system conditions than insecurity. To put it in another perspective, attackers came closer to guessing the legitimate passphrase while shoulder surfing with the bullet masking of the control system conditions than they did while guessing passphrases based on the pictures of the new system conditions. Moreover, due to the nature of passphrases, the closest by Damerau-Levenshtein distance that any attacker got to a legitimate passphrase was farther away than the entire password length in our previous studies in Chapters 3 and 4.

For cracking security against machine attackers, User-generated was found to take 2^{24} attempts on average, over 10 bits less secure than any other condition. (Since the entire passphrase space will only be exhausted by machine guessing in the worst case scenario, and because all possible passphrases are far from equally likely, this does not mean that User-generated was found to exhibit 24 bits of entropy.) Although this effect is extremely significant from a statistical standpoint, and over 1000 times fewer attempts required also seems significant from a practical standpoint, we cannot say with absolute certainty that the difference matters at this point in time.

For the fourth primary hypothesis, that subjective usability measures would be least favorable for Diceware10k, the results were different than expected. While Diceware10k did indeed perform worse than the new system conditions for first-session and second-session recall SUS, User-generated performed just as poorly. This was surprising to us because users typically prefer to create their own passwords or passphrases (Proctor et al., 2002). Since there were no significant differences for creation SUS, it is likely that recall SUS was highly affected by lower performance for the control system conditions in the recall tasks. While Shay et al. (2012) found that participants generally rated system-generated passphrases as difficult and annoying, we believe that the mnemonic pictures, error correction, and decoy masking in our new systems improved their usability.

It came as no surprise to us that we found no significant differences for participants' ratings of perceived security, as users would not be expected to

understand the underlying implications of various design decisions on security. Interestingly, Diceware10k did perform the worst in an overall rating that combined usability and security. User-generated outperformed Diceware10k for the overall rating, even though they performed at the same level for creation SUS, recall SUS, and perceived security. Perhaps this is because, while participants could not process their greater dislike for Diceware10k in terms of specific dimensions such as usability and security, they did feel that they disliked it the most, overall.

Observations

We directly observed a wide range of reactions from participants, demonstrating the effect of individual differences. Some participants were smiling and laughing the entire time, while others appeared to take the experimental tasks very seriously or became frustrated more easily. From what we could tell, these differences were mostly unrelated to success at the experimental tasks and were more likely related to personality.

There was a good deal of uniformity in passphrase creation strategy for the user-generated condition, even though the only requirement was to be at least 24 characters in length. All accounts were created using true phrases (e.g., unlike *Samantha1234567890!!!!!!*). The vast majority of user-created passphrases included spaces between words. Symbols, numbers, and upper case letters were almost never used; when they were, they would inevitably be capitalized first words or proper nouns, apostrophes, or trailing exclamation points. Very occasionally, participants used

intentional misspellings or letter-to-number obfuscations (e.g., *0* instead of *o*). But overall, participants jumped at the opportunity to create an authentication secret with the simplicity of plain language.

One participant selected a phrase in a foreign language, and another strung together four names that are not commonly seen in the United States. While these may have been admirable attempts to bolster security, passphrase crackers can be trained on foreign phrases and names just as simply as English ones. Any additions past plain English, such as capitalization, punctuation, and letter-to-number obfuscations, are particularly poor from both usability and security perspectives, as they introduce extra information that must be remembered, but these additions tend to be easy for passphrase crackers to predict.

Other participants selected phrases based on movie quotations, song lyrics, and online memes. These insecure selections typically occur when users are allowed to select their own phrases (Kuo et al., 2006). While Kuo et al. (2006) found that 65% of user-generated passphrases in their study were found when submitted as exact quotes to Google, this held for 42% of our user-generated passphrases. In any case, this idea suggests the possibility of preventing users from creating passphrases that appear in an online search.

In contrast to passphrase creation, picture creation strategies proved to be diverse. Most participants started drawing relatively quickly, but some adopted a different strategy of thinking about the passphrase for a minute or more before

beginning to draw. The pictures themselves demonstrated a mix of black and white, colors, shapes, symbols, freeform, text or letters (not taken directly from the passphrase), and different levels of artistry or drawing ability (or more charitably, abstractness). Some tried to paint a holistic picture, while others depicted distinct sub-pictures for individual words in sequence. Participants also finished at different times. Some were completely finished before the one-minute warning and spent the rest of the allotted time looking at their picture. Others used the entire five minutes and had to be explicitly told to stop drawing.

When it came time to guess the passphrases of other accounts, participants almost universally seemed amused or exasperated at the task. We made sure to stress that we did not expect anyone to be able to completely guess another passphrase, even with picture support. Some participants were quite excited to guess others' pictures and laughed or said they were cute but had no idea what they were supposed to be.

Participants mostly exhibited similar reactions of exasperation when asked to shoulder surf, although several participants claimed they caught part of an entered passphrase, usually with decoy masking. A few participants even claimed they thought they might have caught the whole thing, but as no passphrase was correctly guessed, the results did not support any of these claims.

We also observed that the defense against shoulder surfing seemed to have more to do with the person typing than with the attacker. Behaviors such as pausing after every word, double checking before submitting, and typing slowly in general would

create potential opportunities for attack. This is why we instructed participants to prioritize speed over accuracy while using decoy masking, reminding them of the built-in error correction.

We noticed several different deducible reasons why participants would fail to sign in successfully, rather than simply forgetting entirely. Many participants expressed disbelief that they had, in particular, forgotten the passphrase they had previously chosen. Memory is fallible, which is why mnemonic aids are so valuable.

For user-generated passphrases, sometimes it was as simple as neglecting an exclamation point at the end, leaving out *the* at the beginning of a phrase, mixing *one* and *1*, forgetting capitalization, or making a typographical error. In this last case, since the user doesn't realize the mistake, he or she will try incorrect variations of the legitimate passphrase on subsequent attempts, proceeding down a cycle of failure. Besides the typographical errors, these other types of mistakes suggest that from an information processing perspective, the memory failure was possibly due to an encoding error rather than a retrieval error.

Since we did not ask participants to confirm their passphrases during creation, we thought that perhaps some users might have made a typographical error during creation instead of login. However, when we examined the passphrases, we noticed that this phenomenon only occurred for one participant. Interestingly, a couple of participants intentionally introduced a misspelling into a passphrase and successfully remembered this modification later during recall.

For Diceware10k, unfamiliar words were troublesome. Symbol-based “words” also caused difficulty, and participants would also forget the order of the words. Unsurprisingly, the single most common reason why participants had trouble with this condition was simply that they were assigned a randomly generated passphrase without any mnemonic aids.

For the new system conditions, because of the mnemonic pictures, participants very rarely experienced a complete inability to remember any aspect of the passphrase. Participants were frequently off by one word, substituting another word with a similar meaning. For the sentence-based passphrase, the animal and the food tended to be easier to remember, while the food adjective and verb tended to give the most problems. This same pattern also held for attackers’ guesses of pictures.

Responses

One of the most consistent statements from participants answering the open-ended interview questions was just how much they enjoyed taking part in the study, compared to other studies they had participated in. Several participants exclaimed that they were going to go home and play a game of Pictionary. Another responded that the Sentence passphrase condition reminded him of the Apples to Apples card game, based on intriguing or unexpected word combinations; others compared the condition to Mad Libs.

While participants almost unanimously liked the mnemonic pictures, perhaps the greatest positive response was actually for the typographical error correction.

Participants felt relieved that they did not have to type their passphrase absolutely correctly, although some felt the need to correct their mistakes regardless. Sometimes, we observed users making a mistake and never realizing it; without error correction, they would have failed the login attempt and not understand why they had failed.

Participants also expressed a great deal of appreciation for decoy masking. This was demonstrated primarily not during the decoy masking tasks but during the standard bullet masking tasks, as the lack of feedback in those cases proved frustrating. Several participants particularly liked how the color and position of the legitimate entry was consistent for every recall task for a particular passphrase.

Another potentially underappreciated feature of our new system was allowing participants a choice of four passphrases for Special English. Some participants remarked that they liked having the feeling of choice in that condition rather than simply the luck of the draw as in the other system-generated conditions.

The passphrase scheme that participants disliked the most was Diceware10k. There was certainly the highest degree of variance in assigned or created passphrases with this condition, as the system could randomly assign three easy words or three difficult words. Naturally, participants who received words that they did not recognize professed a major dislike for the system; a few participants wondered why there couldn't have been definition tooltips for this condition rather than for Special English, where the definitions weren't typically needed. Participants also disliked how symbol characters (e.g., in *50%*, *1/8*, or *!!*) are included in the Diceware10k wordlist. Most

importantly, participants disliked that no mnemonic picture was available for this condition.

The single most insightful suggestion from a participant was an enhancement to strengthen the phrase-based decoy masking when a typographical error has been made that the system has not yet corrected. In this situation, the system would also introduce a typographical error in the decoys and then correct all lines of displayed text simultaneously. The system would know what the correct passphrase is, and while it would not be desirable to correct all errors immediately (or else it would be impossible for an attacker to fail during login), the proposed idea should be feasible.

Later, however, we realized that this change would likely be insecure against passphrase guessers, as an attacker could simply try each character one letter at a time until identifying the correct character as the one that does not induce a typographical error in the decoys. The threat of this type of attack could possibly be mitigated by detecting if an attacker is employing this tactic and then locking the account, but legitimate users may then be caught in the crossfire. Along with the consideration of selecting decoys similar to the legitimate passphrase, these examples demonstrate that there is often a tradeoff not only between usability and security, but also between security measures against different types of attacks.

Finally, several participants noted that getting to play the role of an attacker was both fun and informative. A few participants suggested that this new perspective would probably enable them to draw more effective and secure pictures in the future.

Conclusion

We developed a new authentication system based on passphrases instead of passwords. Our new system incorporates a user-generated mnemonic picture displayed during login, definition tooltips, error correction to reduce typographical errors, a decoy-based input masking technique, and the choice of utilizing either a specialized wordlist or a sentence template. The idea is that these added features work particularly well with passphrases and help to address the usability shortcomings that have slowed the adoption of passphrases.

We conducted a study to evaluate our new system with a customized 1450-word list and our new system with a 6-word sentence structure against the control conditions of a user-created passphrase of at least 24 characters and a system-generated passphrase using a 10326-word list. We found that using the new system conditions, memorability was improved and security was equivalent to or better than the control conditions. Usability and overall ratings also favored the new system conditions over the control conditions.

CHAPTER 7: CONCLUSION

The major contribution of our research to the field of digital authentication is in facilitating a transition from passwords to usable passphrases, resulting in increased security and, potentially, usability. The key lies in addressing the traditional usability issues of passphrases, which have hindered wider adoption. This was accomplished through the integration of passphrases with our component innovations: a specialized wordlist and sentence template, definition tooltips, user-generated mnemonic pictures for recall, error correction while typing, and a decoy-based input masking technique.

We recommend that current authentication systems be steadily adapted to our research contributions and findings. In computer security, drastic changes should never happen overnight, but regular progress is always necessary. Altogether, our contributions represent individual techniques for optimizing usability and security, but one practical takeaway is that careful thought and inventive ideas can overcome obstacles in the constantly evolving area of computer security. As new challenges will always arise, we must be willing to face these challenges with an open mind.

One topic for future work is separating out which aspects of our new system contribute the most beneficial effects. However, several different aspects of our new system would be difficult or infeasible to integrate with existing passphrase schemes. For example, mnemonic pictures are only effective when the meanings of those words are well known. Error correction only works well when the components of a passphrase are clearly defined words, with a low amount of overlap between possible words (e.g.,

bun and *bin* would not be desirable to correct to one another, as they are too similar). Still, knowing what the more or less significant features are would help us to improve them and integrate them into a more effective combination.

Another interesting research angle would be to break down the entire range of memory failures into its component pieces: encoding errors, storage errors, and retrieval errors. By examining the different ways in which users can fail to recall their passphrases, we can better understand how to provide specially tailored mnemonic aids that help them remember those passphrases or other security information.

Studying performance in the wild versus in a laboratory setting would also be important for assessing ecological validity. It may be possible to conduct studies of real accounts, over long time periods, or using Amazon Mechanical Turk. In this situation, we could also measure how much time users would really spend on account creation or login if left to their own devices. More rigorous research into how users actually create real passphrases would also have benefits in understanding passphrase memorability and security.

We can also work on developing more sophisticated methods of passphrase cracking to better evaluate the security of passphrase systems. Not much attention has been given yet to passphrase cracking because the vast majority of authentication secrets in the wild are currently passwords, not passphrases. As the situation changes in the future, passphrase cracking will become more lucrative and will inevitably improve.

Our final consideration is especially interesting from a user experience standpoint: how to foster understanding of best practice using our new system or any other novel innovations in computer security. As system designers, how do we prevent users from defeating the security features imposed or suggested by the system? This is a classic question in human factors research, and part of the answer could involve training, motivation, or simply understanding some of the reasons why users behave the way they do regarding computer security.

Our research presents innovative techniques that improve on the usability and security of existing authentication systems. We feel that further development of these techniques will lead to even greater improvements in the future. In the end, we hope that by focusing attention on the usability of security-critical computer systems, we will inspire other researchers and practitioners to strive to design systems that are both secure and usable.

REFERENCES

Avoine, G., Junod, P., & Oechslin, P. (2008) Characterization and improvement of time-memory trade-off based on perfect tables. *ACM Transactions on Information and System Security*, *11*(4), No. 17.

Bertsch, S., Pesta, B. J., Wiscott, R., & McDaniel, M. A. (2007). The generation effect: A meta-analytic review. *Memory & Cognition*, *35*(2), 201-210.

Biancuzzi, F. (2006, February 22). John the Ripper, by Solar Designer. Retrieved October 14, 2013, from http://www.securityfocus.com/columnists/388/2

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), Usability evaluation in industry (pp. 189-194). London, England: Taylor and Francis.

Burr, W. E., Dodson, D. F., & Polk, W. T. (2006). Electronic authentication guideline: Recommendations of the National Institute of Standards and Technology. Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). New York, NY: Academic Press.

Craik, F. I. M., & Tulving, E. (1975). Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology: General*, *104*(3), 268-294.

Damerau, F. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, *7*(3), 171-176.

Forget, A., Chiasson, S., & Biddle, R. (2010). Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, 1107-1110.

Fraune, M. R., Juang, K. A., Greenstein, J. S., Chalil Madathil, K., & Koikkara, R. (2013). Employing user-created pictures to enhance the recall of system-generated mnemonic phrases and the security of passwords. *Proceedings of the 57th Annual Meeting of the Human Factors and Ergonomics Society*, 419-423.

Gaw, S., & Felten, E. W. (2006). Password management strategies for online accounts. *Proceedings of the Second Symposium on Usable Privacy and Security*, 44-55.

Goverover, Y., Basso, M., Wood, H., Chiaravalloti, N., & DeLuca, J. (2011). Examining the benefits of combining two learning strategies on recall of functional information in persons with multiple sclerosis. *Multiple Sclerosis Journal*, *17*(12), 1488-1497.

Greenwald, A. G., & Banaji, M. R. (1989). The self as a memory system: Powerful, but ordinary. *Journal of Personality and Social Psychology*, *57*(1), 41-54.

Hart, S. G., & Staveland, L. (1988). Development of NASA-TLX: Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati (Eds.), Human mental workload (pp. 239-250). Amsterdam, Netherlands: North-Holland Press.

Herley, C., & van Oorschot, P. C. (2012). A research agenda acknowledging the persistence of passwords. *IEEE Security & Privacy*, *10*(1), 28-36.

Hoanca, B., & Mock, K. (2005). Screen oriented technique for reducing the incidence of shoulder surfing. *Proceedings of the 2005 International Conference on Security and Management*, 334-340.

Hruska, T. (2011, November 3). How to calculate password strength. Retrieved October 14, 2013, from http://cubicspot.blogspot.com/2011/11/how-to-calculate-password-strength.html

Jeyaraman, S., & Topkara, U. (2005). Have the cake and eat it too – infusing usability into text-password based authentication systems. *Proceedings of the 21st Computer Security Applications Conference*, 482-491.

Juang, K. A., & Greenstein, J. S. (2011). Evaluating the usability and security of input masking techniques. *Proceedings of the 55th Annual Meeting of the Human Factors and Ergonomics Society*, 1120-1124.

Juang, K. A., Ranganayakulu, S., & Greenstein, J. S. (2012). Using system-generated mnemonics to improve the usability and security of password authentication. *Proceedings of the 56th Annual Meeting of the Human Factors and Ergonomics Society*, 506-510.

Karole, A., Saxena, N., & Christin, N. (2010). A comparative usability evaluation of traditional password managers. *Proceedings of the 13th International Conference on Information Security and Cryptology*, 233-251.

Keith, M., Shao, B., & Steinbart, P. J. (2007). The usability of passphrases for authentication: An empirical field study. International Journal of Human-Computer Studies, *65*(1), 17-28.

Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., Bauer, L., . . . Lopez, J. (2012). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, 523-537.

Kelly, C. (2010, January 1). VOA Special English word book. Retrieved March 13, 2013, from http://www.manythings.org/voa/words.htm

Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., . . . Egelman, S. (2011). Of passwords and people: Measuring the effect of password-composition policies. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2595-2604.

Kuo, C., Romanosky, S., & Cranor, L. F. (2006). Human selection of mnemonic phrase-based passwords. *Proceedings of the Second Symposium on Usable Privacy and Security*, 67-78.

Kurzban, S. A. (1985). Easily remembered passphrases: A better approach. *ACM SIGSAC Review*, *3*(2), 10-21.

Leonhard, M. D., & Venkatakrishnan, V. N. (2007). A comparative study of three random password generators. *IEEE International Conference on Electro/Information Technology*, 2007, 227-232.

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human Computer Interaction*, *7*(1), 57-78.

Mäntylä, T. (1986). Optimizing cue effectiveness: Recall of 500 and 600 incidentally learned words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *12*(1), 66-71.

Nelson, D., & Vu, K. L. (2010) Effectiveness of image-based mnemonic techniques for enhancing the memorability and security of user-generated passwords. *Computers in Human Behavior*, 26, 705-715.

Nielsen, J. (2009, June 23). Stop password masking. Retrieved August 29, 2012, from http://www.useit.com/alertbox/passwords.html

Percival, C. (2009). Stronger key derivation via sequential memory-hard functions. *Proceedings of BSDCan 2009*, 1-16.

Porter, S. N. (1982). A password extension for improved human factors. *Computers and Security*, *1*(1), 54-56.

Proctor, R. W., Lien, M. C., Vu, K. P. L., Schultz, E. E., & Salvendy, G. (2002). Improving computer security for authentication of users: Influence of proactive password restrictions. *Behavior Research Methods, Instruments & Computers*, *34*(2), 163-169.

Rao, A., Jha, B., & Kini, G. (2013). Effect of grammar on security of long passwords. *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*, 317-324.

Reinhold, A. (2012, March 8). The Diceware passphrase home page. Retrieved December 28, 2012, from http://world.std.com/~reinhold/diceware.html

Renaud, K. (2005). Evaluating authentication mechanisms. In L. F. Cranor & S. Garfinkel (Eds.), *Security and usability* (pp. 103-128). Sebastopol, CA: O'Reilly.

Rogers, T. B., Kuiper, N. A., & Kirker, W. S. (1977). Self-reference and the encoding of personal information. *Journal of Personality and Social Psychology*, *35*(9), 667-688.

Roth, V., Richter, K., & Freidinger, R. (2004). A PIN-entry method resilient against shoulder surfing. *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 236-245.

Sasse, M. A., & Flechais, I. (2005). Usable security. In L. F. Cranor & S. Garfinkel (Eds.), *Security and usability* (pp. 13-30). Sebastopol, CA: O'Reilly.

Scarfone, K., & Souppaya, M. (2009). Guide to enterprise password management (draft): Recommendations of the National Institute of Standards and Technology. Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology.

Shay, R., Kelley, P. G., Komanduri, S., Mazurek, M. L., Ur, B., Vidas, T., . . . Cranor, L. F. (2012). Correct horse battery staple: Exploring the usability of system-assigned passphrases. *Proceedings of the Eighth Symposium on Usable Privacy and Security*, No. 7.

Shiner, J. (2013, April 16). On hashcat and strong master passwords as your best protection. Retrieved October 14, 2013, from http://blog.agilebits.com/2013/04/16/1password-hashcat-strong-master-passwords/

Slamecka, N. J., & Graf, P. (1978). The generation effect: Delineation of a phenomenon. *Journal of Experimental Psychology: Human Learning and Memory*, *4*(6), 592-604.

Stobert, E., & Biddle, R. (2013). Memory retrieval and graphical passwords. *Proceedings of the Ninth Symposium on Usable Privacy and Security*, No. 15.

Suo, X., Zhu, Y., & Owen, G. S. (2005). Graphical passwords: A survey. *Proceedings of the 21st Annual Computer Security Applications Conference*, 472-481.

Tari, F., Ozok, A. A., & Holden, S. H. (2006). A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. *Proceedings of the Second Symposium on Usable Privacy and Security*, 56-66.

Teat, C., & Peltsverger, S. (2011). The security of cryptographic hashes. *Proceedings of the 49th Annual Southeast Regional Conference*, 103-108.

Tognazzini, B. (2005). Design for usability. In L. F. Cranor & S. Garfinkel (Eds.), *Security and usability* (pp. 31-46). Sebastopol, CA: O'Reilly.

Whitten, A., & Tygar, J. D. (1999). Why Johnny can't encrypt: A usability evaluation of PGP 5.0. *Proceedings of the 8th USENIX Security Symposium*, 169-184.

Wiedenbeck, S., Waters, J., Sobrado, L., & Birget, J. C. (2006). Design and evaluation of a shoulder-surfing resistant graphical password scheme. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 177-184.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. *Proceedings of the Section on Survey Research Methods*, 354-359.

Wright, N., Patrick, A. S., & Biddle, R. (2012). Do you see your password? Applying recognition to textual passwords. *Proceedings of the Eighth Symposium on Usable Privacy and Security*, No. 8.