IMPROVING PASSWORD USABILITY WITH VISUAL TECHNIQUES

by

Saranga Komanduri

A Thesis

Submitted to the Graduate College of Bowling Green
State University in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

December 2007

Committee:

Dugald Hutchings, Advisor

Ray Kresman

Laura Leventhal

ABSTRACT

Dugald Hutchings, Advisor

Current technology is more than capable of providing a secure authentication process, but users are limited in their ability to remember difficult passwords and make judgments about security. In this thesis, both of these problems were addressed in the development of a new picture-password system. Studies in computer security, password usability, and cognitive psychology were referenced throughout the design process.

The new system was tested against character passwords of equal complexity and recall was measured over an eight-day period. The results of the study show that a picture password system which accepts input without respect to order can be both secure and memorable. Both character and picture passwords were randomly assigned from a password space of almost 29 billion possible passwords. The picture passwords were remembered by all fifteen of the picture-group participants eight days later, whereas character passwords were not universally remembered.

In addition, I found that a substantial number of failed inputs by users are repeats of previous inputs in the same session. This reduces the number of distinct guesses users can make when authentication systems limit the number of allowed failed logins (known as *account lockout*). A protocol was developed to ignore these duplicate inputs in a way which incurs almost no security risk and improves usability.

ACKNOWLEDGEMENTS

It was Dr. Lancaster who first suggested that I persue the thesis option when I began to consider Ph.D. programs. I am certain that this thesis would never have happened without his advice.

Months later, I approached Dr. Kresman with a very rough idea for a picture password system after having done a small amount of research. Meetings with Dr. Hutchings and Dr. Leventhal soon followed and everyone was very supportive and full of ideas on how to proceed. So much so, that I was unsure of where to concentrate my efforts. I decided to discuss the matter with Dr. Dunning, who helped me to narrow my scope and decide on a broad structure.

Throughout the process, feedback from my committee was invaluable. Frequent meetings with Dr. Hutchings helped me to understand research, experiments, and academic writing in general, and his comments always led to improvements in the finished product. Dr. Leventhal introduced me to related work in the psychology literature and also helped me with materials for running experiments in the CHIL (Computer-Human Interaction Lab). Dr. Kresman helped me with security topics and in improving the material in Chapter 5. Everyone in my committee pushed me to improve the structure of this manuscript from a simple documentation of work to a (hopefully) more cohesive product. Though I struggled to meet deadlines, and often failed to do so, the final product is better for it.

I would also like to thank Dr. Klopfer. Though not a member of my committee, he passed along several relevant papers that piqued my interest in psychology.

My family and friends participated in pilot testing the experiment and supported me throughout. I'd especially like to thank Jason, Casey, and Colin who seemed genuinely interested in my research even though it often left me with very little free time. They have always supported me without reservation.

Finally, I'd like to thank Josh and the rest of the staff of the Technology Support

Center for allowing me time off when I really needed it. They haven't known me for very long but they've been extremely flexible and forgiving of my schedule these last two weeks. I am very lucky to have found such an enjoyable place to work.

I would **not** like to thank Time, which, for the entirety of this thesis, was never on my side.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION AND DEFINITIONS

1.1 Introduction

The "password problem" can be stated in the following way [Wiedenbeck et al. 2005; Sasse et al. 2001]:

- Passwords must be strong enough to be secure, **but**
- Passwords must be remembered by users.

These two requirements present users with conflicting constraints. As password policies require users to include numbers, uppercase letters, and special characters in their passwords, the resulting strings become less meaningful and more difficult to remember. In short, secure passwords are more difficult to remember because they incorporate more randomness [Wiedenbeck et al. 2005].

Using pictures in passwords is a promising solution to the problem due to the superior memorability of pictures as first studied by Nickerson [Nickerson 1965]. Several picture-based password systems have been designed and studied in the last few years [Dhamija and Perrig 2000; Brostoff and Sasse 2000; Wiedenbeck et al. 2005] with generally positive results.

At the same time, the emergence of *phishing*[1] has led to several studies examining user behavior with respect to security indicators[2]. Whether due to lack of user attention

[1] See section 2.1.1, "Phishing", page 7.

[2] Security indicators are screen elements that inform the user if a site or service is secure (or insecure). A common example is the padlock icon which appears in most browsers during an https connection. For

or inherent ineffectiveness, security indicators do not perform well enough to thwart attempts to steal user passwords.

I considered both the password and phishing problems in the design of my picture-password system. There were two important requirements:

- Creating a picture-based password system whose passwords meet or exceed the strength of character passwords.
- Making the system itself act as a security indicator through rearrangement of pictures.

The design decisions made in order to meet these requirements are explained throughout Chapters 2 and 3.

The rest of this chapter contains a few definitions which are relevant to any discussion of passwords. A survey and discussion of related work is presented in Chapter 2, along with my own contribution, "The Distinctiveness Dilemma," starting on page 25. The design of both the picture-password system and the research study are discussed in Chapter 3, with results given in Chapter 4. Chapter 6 outlines possible directions for future work and Chapter 5 presents another contribution of my own, a protocol to ignore duplicates of incorrect passwords.

1.2 Definitions

Before discussing passwords and related issues, a few terms need to be defined. Words like *complexity* and *entropy* have specific meanings in the context of passwords and the definitions given below are intended to help readers understand these terms in later

more information, see section 2.1, "Security Indicators", page 7.

chapters. I have deferred the definition of some terms, such as *phishing*, to Chapter 2 where they are presented in a more relevant context.

1.2.1 Passwords

The term *password* in this thesis will be used in both the familiar sense, where it describes a string of characters, and a more novel sense, in which it refers to the items selected in a graphical authentication input. This input may be an ordered/unordered set of particular clicks or keypresses, or an ordered/unordered set of items which the user must choose in response to on-screen prompts.

1.2.2 Complexity

Complexity in this paper is used to represent the number of possible combinations of user input relative to the number of inputs which the system will consider successfully authenticated. This is often referred to as the *strength* of the password. For example, a character-based password system which uses eight-character passwords taken from a set of 94 characters has a complexity of $94^8 = 6.1 \times 10^{15}$. Complexity, or strength, increases if users are allowed to use longer passwords or draw from a larger character set.

Complexity can also decrease if the number of acceptable inputs increases. For example, if the previously mentioned eight-character passwords were accepted in any order, the system would accept up to 40,320[3] possible inputs for each password. If the user's password is "software", the system would also accept "swearoft", "wastefor", "reawtfos", "sortafew", etc. This decreases the complexity to $\binom{94+8-1}{8} = 2.0 \times 10^{11}$ (the number of combinations with repetition of 94 items taken eight at a time.) Accepting passwords in any order seems to greatly improve their memorability and is discussed often in this thesis. It is next revisited in section 2.7.2 on page 24.

[3] $40,320 = 8!$

1.2.3 Entropy

Entropy was a term originally defined by Claude Shannon [Shannon 1948] as a measure of the amount of information in a given datum. When applied to passwords it is typically used to represent how easily a password might be guessed, and is sometimes referred to more specifically as *guessing entropy*. For example, with no strict password policy, user-selected 8-character passwords only carry about 18 bits of entropy [Burr et al. 2004]. This is substantially less than the over 52 bits of entropy available from a set of 94 characters[4].

It should be noted however, that password entropy has never been truly measured. Complete collections of real user passwords are not available for analysis due to the predominance of hashing in password systems[5]. In the example just cited, researchers at the NIST[6] used a set of heuristics to estimate the entropy of passwords.

1.2.4 Brute Force

Brute force refers to an attack in which possible passwords are guessed at random until a working password is discovered. Using truly random passwords, the time required to brute-force a password is proportional to the complexity of the password space. With user-selected passwords, the complexity of the password space is replaced by its guessing entropy. For example, an average user-selected password that carries 18 bits of entropy might be guessed after only $2^{16} \approx 66,000$ guesses[7], whereas a random password from the password space requires an average of $2^{53} = 3.0 \times 10^{15}$ guesses[8]. The low entropy value, of 18 bits, assumes that attackers[9] have perfect knowledge of the most likely user-selected

[4]The implications of this are discussed in the next section "Brute Force".

[5]See section 1.2.6, "Hashing", page 5.

[6]National Institute of Standards and Technology, a branch of the U.S. Department of Commerce.

[7]This is a lower bound on the average number of guesses as given by [Massey 1994]. The exact relationship between the number of guesses required and entropy is not as straightforward as it seems because it depends on the true frequency distribution of passwords [Malone and Sullivan 2004].

[8]Since the passwords are randomly chosen, their frequency distribution is uniform and the average number of guesses required is simply half of the size of the password space.

[9]An *attacker* is an entity, perhaps a person or program, which attempts to steal a user's password for illicit purposes.

passwords and will try them first. The assumption that attackers will always employ an optimal strategy is often used in security discussions and is used throughout this thesis. Low entropy in user-selected passwords is an often-cited problem in computer security and will be covered in section 2.2.

1.2.5 Shoulder Surfing

Of particular importance to picture-password systems is the problem of *shoulder-surfing*. Shoulder-surfing is the process by which an attacker steals a password by observing its entry. Since picture-based password systems often require use of an on-screen mouse cursor, an observer can watch a user's screen while the password is being entered and steal the password. Character-based authentication systems require the shoulder-surfer to watch the user's keyboard, which is partially obscured by the user's hands. Shoulder-surfing was attempted in my own study and is discussed in section 4.10, "Shoulder Surfing of Character-based Passwords", page 72.

1.2.6 Hashing

An algorithm employed by almost all character-based password systems is *hashing*. Hashing allows an authentication system to match user input to a previously chosen value without storing the value explicitly. This is accomplished by utilizing a one-way cryptographic hash function[10]. The user's password is given as input to the hash function, and its output (commonly referred to simply as the *hash*) is stored in a password file. From this point on, all user input is sent through the hash function and compared with the value stored in the password file. When the user's input matches their password, the hashes will also match, and the user is authenticated.

The main benefit of hashing is that the user's password cannot be readily retrieved from the stored hash. Even if the authentication server's filesystem is accessed by an

[10]See Preneel [Preneel 1993] for a comprehensive survey of hash functions.

attacker, the user's passwords will not be revealed since the hash function cannot be reversed. Some picture-password systems do not employ hashing, and this topic is explored in section 2.7, "Applicability of Hashing to Graphical Authentication Systems", page 21.

Chapter 2

RELATED WORK AND DISCUSSION

This chapter discusses previous studies of password and security-related topics. The discussion begins with security indicators, and then moves to studies in character-password security and usability, followed by research in cognitive psychology. The last half of this chapter discusses issues in picture-password systems.

In the final section of this chapter, I present "The Distinctiveness Dilemma" which places a theoretical limit on the design of picture-based password systems.

2.1 Security Indicators

2.1.1 Phishing

The rapid ascent of *phishing* has generated a lot of interest in security indicators. Phishing is a type of attack in which a fake webpage is sent to a user in an attempt to steal the user's password. It is believed that phishing became widespread starting in 2004 [Anderson 2007a]. In the past few years, phishers have become more sophisticated in their techniques and understanding of users.

In modern phishing, the fake site is an exact copy of the original site and may be served to the user in real-time using a *man-in-the-middle* approach. In a man-in-the-middle attack, the phisher connects to the original website and then passes a copy of it to the user. Almost always, the URL of the phishing site is in some way similar to that of the original site. Sophisticated phishers may even employ picture-in-picture techniques

in which a fake browser window with legitimate URL and security indicators is displayed within the real browser window [Jackson et al. 2007]. The goal of the phisher is to steal the user's password and, ultimately, the user's money. If the password is to an online bank or payment system, the phisher gains access to the user's account. It is estimated that phishing is a $200 million industry in the United States alone [Anderson 2007b].

Security indicators are meant to help users avoid phishing attacks by indicating whether a site is real or fake. Fake sites can be identified by checking for the presence (or absence) of an HTTPS connection[1] or comparing the site's URL to a known database. Unfortunately, research suggests that current security indicators perform poorly.

2.1.2 Security Indicators in Web Browsers

Several problems have been found with the current implementation of security indicators used in web browsers. A recent study in this field was done by Schechter et al. [Schechter et al. 2007]. In their study, they found that 100% of their participants continued to log in to an online banking site in the absence of HTTPS security indicators[2].

The sites visited by participants in their study also employed the SiteKey authentication system. In this system, the user chooses a picture and then is required to notice the absence of their picture. For example, the user may choose a picture of a duck as their security image. Every time they connect to the bank's website, they are shown the same picture of a duck while entering their password. If the "duck" picture is absent, users should not sumbit their password to the site. The SiteKey image is only served to trusted clients, and a phisher would not be able to access the SiteKey image since they are not a trusted client. Untrusted clients must answer a security question, such as "Mother's maiden name" before being trusted with the SiteKey image. When confronted with this style of system, the phisher does not use a man-in-the-middle approach, but instead

[1] The site's SSL certificate can also be checked for authenticity.

[2] In most browsers, the HTTPS indicator consists of a "lock" icon somewhere on the browser frame, and sometimes a green background to the URL bar, also part of the browser frame.

creates a fake website similar to the original with the SiteKey image removed. When these conditions were replicated in Schechter's et al. study, 96% of participants continued to log in to the site though the SiteKey image was absent. Some of the participants in the study were actual customers of the online bank, and 91% of those participants continued to log in.

This is not terribly surprising when you consider the fact that the security indicator, in this case, is not an indicator at all. It is the **absence** of an indicator. Instead of being shown an image which explains to users that a site is unsafe, users are expected to notice the absence of a safety image.

It should also be noted that since this study was conducted, the SiteKey system has been somewhat modified. In the new SiteKey system, users choose a security picture and then are encouraged to generate a password based on this picture. This is intended to draw user attention toward the SiteKey image when they login, and make its absence more noticeable. A study which incorporates this aspect of the SiteKey system has not yet been reported. Further, it is difficult to know the guessing entropy of such a system. If anecdotal evidence about user-selected passwords is true, the guessing entropy of such passwords will be very low. If users are shown a picture of a *white duck* and asked to base their password on it, how many users will choose "whiteduck" as their password? Due to the cognitive demands of multiple passwords, it is also possible that users will ignore the picture completely and use one of their standard passwords instead.

The final security indicator tested by the study was an HTTPS warning page. Before entering their passwords, users were redirected to a webpage explaining that the site was insecure, and needed to click on a link labeled "Continue to this website (not recommended)" to log in. 36% of actual bank users clicked the "not recommended" link and continued to log in to the site. An earlier study by Dhamija et al. discovered similar results of such warnings [Dhamija et al. 2006], even though users in that study were explicitly asked to determine if websites were real or fake.

It is clear that security indicators do not perform as expected. Previous research agrees on a single reason for this, which can be summarized simply:

- Users do not possess proper knowledge of computer security. This makes them unaware of insecure behaviors, and unsure of how to deal with security warnings.

Lack of understanding of insecure behaviors is supported by my own study (see section 4.8 on page 71). Unfortunately, education does not provide a simple answer. Insufficiently educated users develop paranoia about sites and will mark legitimate websites as illegitimate [Anandpara et al. 2007; Jackson et al. 2007]. It may be that the amount of education needed to produce a proper understanding of correct security behaviors is impractical for the casual user. Picture-in-picture phishing techniques present an even greater problem to security indicators, as all security indicators for the inside window (which could appear like a pop-up window) may appear legitimate since the entire window is fake.

Though not mentioned in related work, I believe there is another reason why users ignore security indicators:

- Security indicators are unreliable.

McAfee SiteAdvisor is unable to discern a phishing website from a legitimate one [Anderson 2007b]. Legitimate emails are often thrown into a "Junk" mailbox, while real spam still makes it into users' inboxes. If users mistrust security indicators, then the indicators become useless.

This causes users to become desensitized to security warnings. By default on most browsers, submitting any information on a web form (including a standard web search) brings up a security warning about "submitting unencrypted form data" the first time this task is performed. Users routinely dismiss such messages (with good reason, in the case of a web search), and it is easy to see how such practices can habituate users into ignoring security indicators.

Dhamija et al. also bring up the problem of "bounded attention", which was confirmed by Schechter et al. [Dhamija et al. 2006; Schechter et al. 2007]. Users do not pay attention to security indicators because they are irrelevant to their primary tasks[3]. Though this is an excellent observation, bounded attention may be a symptom of the ineffectiveness of security indicators rather than the cause.

2.2 User-Selected Passwords

In my picture-password system, passwords were assigned rather than user-selected. Issues with user-selected passwords are detailed in this section.

User-selected passwords are thought to have low entropy[4], and security professionals often criticize users for having "weak" passwords. In a study by Yan et al. of university students' passwords, password files were obtained and then cracked using a variety of mechanisms [Yan et al. 2004]. Note that since hashing[5] was used, the passwords could not be directly recovered from the password files. Instead, various "attacks" were attempted which produced a large number of guesses based on predefined rules. The most successful of these was a permuted dictionary attack, in which 32% of passwords in the control group (who were given no explicit instructions in creating their password) were cracked. In this attack, only dictionary words, dictionary words with common number substitutions (such as 0 for O), and permutations of dictionary words with 1, 2, or 3 digits were used as password guesses. Even though this requires a very large number of possibilities to be attempted, it is far less than the available password space. For example, given eight-character passwords taken from a set of 80 items, the password space contains 1.7×10^{15} possible passwords. Although Yan et al. does not provide details as to how many

[3]This idea informed the design of my picture-password system, which is designed to thwart phishing attempts by making the login task more difficult from an insecure location. This is discussed in section 3.6.2 on page 45.

[4]See section 1.2.3, "Entropy", page 4.

[5]See section 1.2.6, "Hashing", page 5.

guesses were generated, I think 10^8 would be a good estimate[6]. This is over 10,000,000 times smaller than our strong password space for 8 character passwords, yet 32% of user passwords were cracked. This illustrates why low entropy passwords are criticized by security professionals. However, as explained before[7], the true entropy of passwords has never really been tested. In Yan's et al. study, the passwords were created for accessing a class website, so security was probably not a user concern.

Of greater concern than the general entropy of passwords is the existence of "popular" passwords. In an analysis of 20,000 passwords which may belong to users of MySpace, Bailey found the most popular passwords to be "cookie123","iloveyou", and "password" [Bailey 2006]. Though anecdotally considered a common password, it is unknown just how often a word like "password" is actually used as a password. In Bailey's case, these three passwords only accounted for 37 passwords out of 20,000.

However, security professionals are concerned that a popular password could be used as a "weak link" through which to get access to an organization's intranet [Morris and Thompson 1979]. An attacker does not necessarily need to guess a particular user's password, but just one password out of the hundreds of users at an organization of interest. Once an attacker accesses one user account, they may be able to use other techniques to reach more sensitive files.

When all passwords are randomly-generated and computer-assigned, these considerations completely disappear. So why do user-selected passwords remain in use in almost all password systems and most organizations? There are two reasons for this:

- Assigned passwords are believed to be impossible for users to remember.
- Password resets must be handled securely.

In choosing the password policy for an organization, the latter reason (password resets) can be a difficult problem to overcome. In a study at BT Group (formerly known as

[6]This is a rough estimate based on the assumption of 50,000 words in the dictionary and 2,000 permutations per word.

[7]See section 1.2.6, "Hashing", page 5.

British Telecom) 91.7% of users requested password resets in a 6-month period [Sasse et al. 2001]. Though this might be an extreme example, users inevitably need password resets and these resets often require two steps. The first step involves setting the user's password to some arbitrary string. The second step occurs once the user logs in with their arbitrary string and resets their password to something secret. This is required because it is difficult to securely reset and notify a user of a new password. If the password is transmitted over email, it is incredibly insecure. If a helpdesk person resets a password and communicates it over the phone, it is not a good security practice to leave the password in place, since the helpdesk worker now knows the password. The password could be transmitted directly to the user through a secure webpage, but most organizations do not have such password systems in place. Further, it is often considered important that a human be part of the password reset procedure. When the process is not supervised by a human, attackers might be able to reset passwords and gain access to user accounts (although there is plenty of evidence that hackers can impersonate employees easily and effectively [Granger 2001].) The current solution to these password reset issues is to give users the ability to set their own passwords.

However, technology (like secure webpages) could be used to transmit passwords to employees easily and securely. Humans could still supervise the process, but instead of giving the user their password over the phone, they could initiate the generation and transmission of the password to the user through a company web-portal or secure application.

While this addresses the problem of securely resetting passwords, the memorability of assigned passwords is still in doubt. Our own study found that when passwords were accepted without respect to order, assigned picture passwords can be highly memorable. These results are presented in Chapter 4.

2.3 Password Usability

Password memorability and ease of entry are arguably the most important elements of password usability. These topics are covered in the following sections.

2.3.1 Memorability

The most important issue in password usability is memorability. Most usability researchers support *mnemonic* passwords [Yan et al. 2004; Sasse et al. 2001]. A mnemonic password is a strong password that the user has associated with a meaningful phrase. For example, a user could select the phrase "Each week I get a seven dollar car wash." From this phrase is produced the password: EwIga7$cw. Note that this is a very strong password and should pass any organization's password requirements.

However, Yan et al. report that mnemonic passwords do not actually improve memorability [Yan et al. 2004]. While mnemonics do improve the memorability of strong passwords, they do not produce passwords which are more memorable than user-selected, weak passwords. Further, about 10% of users will ignore advice about mnemonic passwords and will use their own methods to generate passwords. The passwords generated by these users tend to be weak and relatively easy to guess. Therefore, mnemonic passwords are not applicable as a solution to the password problem. However, in organizations which use standard authentication systems, user education about mnemonic passwords may be beneficial by increasing the average entropy of user passwords.

2.3.2 Ease of Entry

Passwords tend to be short, so ease of entry is usually not a problem. However, many security professionals espouse the use of *passphrases*. It is important to note that the term passphrase in the literature can refer to mnemonic passwords [Yan et al. 2004] or long-form passwords consisting of several words [Keith et al. 2007]. It is the latter definition which is typically used by security professionals.

Passphrases are seen as a solution to the password problem because they combine extremely high complexity with memorability. A passphrase is simply a sequence of words with the spaces between words removed. For example, "ilikeicecreamandpie" is a valid passphrase. Note that this passphrase is 19 characters, compared to the 6-8 character passwords considered previously. An attacker who simply guesses combinations of characters will take far too long to crack this password. However, it is reasonable to assume that the attacker would be aware of the fact that an organization is using passphrases and will modify their attack to guess combinations of words. Even so, given the amount of words available for users to choose, the complexity of passphrases is very high.

Passphrases are thought to be highly memorable since they do not require remembering special characters or numbers[8]. A phrase composed of six words should be quite memorable since both the phrase and individual elements have semantic meaning [Nelson 1977].

Keith, Shao, and Steinbart [Keith et al. 2007] recently published a study in which they compare passphrases with traditional strong passwords and unrestricted passwords. Their results conclusively show that passphrases should not be adopted as an alternative to traditional passwords. Memorability was equivalent to that of regular passwords, but ease of entry was far lower because the extra length of passphrases brings an increased probability of typing mistakes. Users found this aspect of passphrases very frustrating. Since memorability was no better than regular passwords, there seems to be no benefit to the use of passphrases. This is an important result for future authentication system design: increasing the amount of keyboard input can cause a larger number of failures due to user error.

Another factor considered with ease of entry should be the problem of duplicate entries while authenticating. This was discovered in the results of my study and does not

[8]Some security professionals do recommend the addition of numbers or special characters to passphrases [Reinhold 2007], but I do not believe this is necessary given their already high complexity.

appear to have been addressed elsewhere. An approach to improving usability by ignoring duplicate password inputs is presented and analyzed in Chapter 5.

2.4 Picture Superiority

The *picture superiority effect* is repeatedly confirmed in psychological studies. It is normally attributed to Standing [Standing 1973] who performed a study in which participants learned up to 10,000 pictures and were then asked to perform forced-choice recognition tests[9]. Pictures were found to have superior memorability to words in a large battery of tests. There are some tasks for which pictures are not superior to words but these involve picture fragments [Weldon and Roediger 1987], rapid presentation [Paivio and Csapo 1971], and tasks involving "similar" pictures [Snodgrass and McCullough 1986]. When used in authentication systems, pictures are seen as an easy way to improve the memorability of passwords.

Though various theories accounting for the picture superiority effect exist, the models which most informed the design of my picture-password system were Nelson's "Sensory-Semantic" or "Levels of Processing" model [Nelson 1977] combined with Snodgrass's findings related to polysemy [Kinjo and Snodgrass 2000]. All modern theories seem to agree that pictures are readily stored semantically and undergo more elaborate cognitive processing than words, which increases memorability. They also agree that where polysemy occurs (when pictures or words have multiple meanings) memorability is reduced.

Unfortunately, the problem of picture "similarity" is a difficult one, because pictures may be similar along any one of several dimensions: conceptually, verbally, schematically[10], etc. These ideas are expanded and provided context throughout this

[9] In a two-alternative, forced-choice test the participant is required to choose between a previously learned image and a previously unseen image. Standing performed his tests with up to 32 alternatives and still confirmed a picture superiority effect [Standing 1973].

[10] Schematic similarity occurs when two pictures have spatially similar representations. For example, a tire and a doughnut, though not conceptually similar, are schematically similar if scaled to the same

thesis.

2.5 Overview of Graphical Authentication Systems

The rest of this chapter discusses graphical authentication systems (which include picture-password and shoulder-surfing resistant systems) and important aspects of their implementation.

2.5.1 Recognition vs. Recall

One of two memory tasks may be used by an authentication system: *recognition* and *recall*. A recall task involves the reproduction of previously learned material from a basic stimulus. For example, a password must be retrieved from memory into a blank textbox, or a list of items must be repeated in serial order. This is the task employed by current, character-based authentication systems. A recognition task involves the correct identification of previously learned material upon encountering it. For example, choosing the correct option on a multiple choice test, or identifying a previously seen picture among a group of previously unseen pictures.

The use of recognition in graphical authentication systems is very beneficial for memorability, since recognition performance for pictures is very high. However, the picture superiority effect still holds for recall activities [Standing 1973]. Though memory performance is lower for recall activities compared to recognition, the relative advantage of *recall* for pictures, when compared to text, is higher than the relative advantage of *recognition* for pictures compared to text. Further, there are several obstacles in the design of recognition tasks when applied to picture-password systems.

---

apparent size.

2.5.2 Distractor Images

Recognition tasks require the use of constantly changing *distractor images*. Generally, distractor images are any images on a screen which are not part of the users password. In some recognition-based systems these images change each time the user begins the authentication process[11], but in other systems they are always the same[12] or only change after each successful authentication[13].

The use of distractor images requires a large set of images from which to pull the distractors. One practical problem with this is the compilation of a set of images. The Déjà Vu system [Dhamija and Perrig 2000] uses "Random Art" images. These are generated by a computer algorithm designed to create random, yet structured, images. However, the pictures still go through a manual selection process to remove weak images and ensure that all images possess a recognizable structure. This leads to "The Distinctiveness Dilemma" which is discussed in section 2.8.

Repeated Exposure and Intersection Attacks

Another practical problem with distractor images is the following:

- For security, the distractor images should not change each time the authentication process is initiated.

An attacker could compare two authentication screens and determine the user's password as the intersection of the images on both screens. If the distractor images were to change on successful authentication, the system would be susceptible to the same attack over a relatively longer span of time. Therefore, from a security perspective, it seems necessary to **never** change the user's distractor images. A solution to this problem is discussed in the following paragraph, "Multiple Authentication Screens." Note that this has grave

[11] Déjà Vu [Dhamija and Perrig 2000]
[12] PassFaces [Tari et al. 2006]
[13] Proposed improvement to Déjà Vu, see the paragraph in section 2.5.2 titled "Multiple Authentication Screens."

implications for recognition, as the user could become accustomed to seeing the same images over time and confuse them with their own password images. The performance would eventually deteriorate to that of a recall task, with the benefits of recognition lost entirely. The long-term effect of repeated exposure to the same distractor images is something that requires further study.

Multiple Authentication Screens

The authors of the Déjà Vu system [Dhamija and Perrig 2000] identify the problem of repeated exposure and propose a two-step solution. The first step is to enlarge the user's password set to be greater than the number of password images required to authenticate. For example, the user needs to select six images to authenticate, but has 20 images in their "portfolio" [14]. When authenticating, the user will see six of the 20 images, selected at random, and must choose all six to authenticate. In this way, the intersection of multiple authentication screens is unlikely to yield enough items to authenticate because a different combination of six images from the portfolio is shown each time. Second, the user must click through several challenge screens to authenticate, each screen containing only one image from their portfolio. If an incorrect image is chosen, the following screen will show only distractor images, with no indication to the attacker that a password image is not being shown. Thus, the number of authentication attempts and screen intersections the attacker must make becomes quite high.

This is an effective solution to the distractor image problem, but leads directly into a new problem, "The Distinctiveness Dilemma" (see 2.8). It should be noted that use of multiple screens increases the authentication time of users, as they must search for their images on each challenge screen. The time to authenticate with the Déjà Vu system is unknown but, in the convex hull click system [Wiedenbeck et al. 2006], users required 72 seconds to authenticate over five challenge screens. Compare this to the 13.7 seconds

[14]The term portfolio was used by Dhamija and Perrig, but concrete numbers such as 20 and six were not given.

picture-password participants spent authenticating in my study[15] and users may not be willing to wait through multiple authentication screens. Finally, the nature of this type of authentication system makes hashing impractical[16].

2.6 Ambiguity and Variability of User Input

Here, I define **ambiguity** as a property of authentication systems in which the user is capable of authenticating with several possible inputs, but only one such method is chosen (at random) by the server and presented to the user. An example is the proposed Déjà Vu system mentioned above (section 2.5.2). **Variability** in user input occurs when the server always presents the same screen to the user, but multiple inputs are allowed for successful authentication. Allowing ambiguity or variability in user input can have several benefits, such as allowing secure authentication while being observed and even recorded [Wiedenbeck et al. 2006] or improving user performance by employing tolerances [Wiedenbeck et al. 2005]. However, there are two practical problems with ambiguity or variability in user input: a reduction in password complexity, and, for some systems, difficulty in hashing. These issues will be discussed in section 2.7 on page 21.

2.6.1 Resistance and Immunity to Shoulder-Surfing

Shoulder-surfing resistant password systems allow for secure authentication while being observed by a live observer, but are not secure to recording devices. An example of this is the Spy-resistant Keyboard [Tan et al. 2005]. By the time user input is observed, it is too late for an attacker to map that input to on-screen elements. However, if a shoulder-surfing resistant input is recorded, and then played backwards or forwards multiple times, the password items can be stolen[17].

[15] 22.4 seconds including failed logins. See section 4.3, "Comparison of Entry Times", page 61.

[16] See section 2.7, "Applicability of Hashing to Graphical Authentication Systems", page 21.

[17] See section 3.6.2, "Shoulder-Surfing Resistant Input", page 47, for an explanation of our shoulder-surfing resistant system and a more complete description of the authentication process.

Truly ambiguous authentication systems can prevent an observer from authenticating even if they have recorded and analyzed a single input. Such systems may be considered *shoulder-surfing immune*. Since the user is capable of authenticating through different methods, and only one of these methods was recorded, an attacker can only successfully authenticate if they are presented the exact same authentication method by the server.

For example, the convex hull click system [Wiedenbeck et al. 2006] has the user click within the convex hull created by a subset of icons the user has previously learned. The points the user chooses may be within any of the potential convex hulls created by hundreds of icon triplets, and the arrangement of the icons is randomized for each input. It is unlikely that an observer could ever gather enough data to successfully authenticate, even after recording successful authentications.

There are some difficulties with this scheme. If users always click near the geometrical center of the convex hull, it may be possible to analyze the user screens and click points to determine the user's icons over a number of authentications. The authors cite this as a potential problem and feel the number of icons on the screen (currently 112) should be increased. However, this system is still much more secure against observation than traditional or shoulder-surfing resistant authentication systems, due to ambiguity in user input. Unfortunately, this ambiguity can make hashing impractical.

2.7 Applicability of Hashing to Graphical Authentication Systems

As explained in section 1.2.6, hashing allows authentication systems to match a user's input to their password without storing the password explicitly. Many existing graphical authentication systems do not use hashing [Man et al. 2003; Brostoff and Sasse 2000; Wiedenbeck et al. 2005] instead assuming the existence of a highly secure authentication

server. Though this is not a bad assumption, the use of hashing in character-based password systems presents an argument against graphical passwords which cannot be readily rejected. Further, such systems become less secure in situations requiring local authentication. These include logging on to a shared PC, or systems which locally cache server authentication credentials for use when the server is unavailable [Microsoft Corporation 2007].

These systems might not employ hashing because they rely on variable user input. For example, the PassPoints system [Wiedenbeck et al. 2005] requires the user to click on specific points in a shown image, within a tolerance of 5 mm. There are a large number of combinations of pixels that the user could choose and still successfully authenticate. Successful user input for this system is not a predictable datum as in character-based password schemes but a large set of possible inputs. Hashing all possible inputs and then comparing them to user input is impractical. Assuming there are 20 pixels within the tolerance range of 5 mm, and six points in the users password, there are 20^6 possible inputs. Storing and comparing this many hashes is impractical and lessens the security benefit of hashing. If an attacker obtains the password file and attempts a brute force attack on the hash algorithm (computing hashes for all possible combinations of user input), they will now succeed in $\frac{1}{20^6}$th of the time, since they could successfully authenticate with only one of the possible inputs.

For ambiguous authentication systems, such as the convex hull click system [Wiedenbeck et al. 2006], hashing does not appear to be applicable. This topic is covered on page 24.

2.7.1 Reduction in complexity

The problem of storing a huge number of hashes for variable user input can be solved by using some function to encode all user-input so that successful inputs map to a single value. For the PassPoints scheme, the screen could be broken into squares of adequate

size to contain the tolerance radii for all passpoints. Since multiple points are involved, and points may fall near the edge of a square, it is necessary to also store an offset, so that tolerance ranges around points fall entirely within square boundaries. Note that this increases the tolerance area to the entirety of these squares. User input can be considered an ordered list of the squares the user clicked inside (note that users do not see the actual squares) and successful user input will always be the same list of squares. The square size, offset, and hash can be stored as a single entry in the password file. Though this deals with the hashing problem, it greatly reduces the complexity of the password. Password inputs which would have been previously unsuccessful would now be successful due to an increase in the tolerance area.

The convex hull click system also suffers this reduction in complexity, but in a far more difficult-to-measure way. The complexity of the password can be given by the average percentage of screen space occupied by a convex hull multiplied by the number of screens an attacker needs to go through. The convex hull may take up to half the screen space [Wiedenbeck et al. 2006], and an attacker can click any point within the hull to authenticate. The average percentage of screen space taken by a convex hull is unknown, but will exceed 50% only in rare circumstances. The system is also set up not to allow convex hulls which are too narrow because they are difficult to click. Hypothetically, an attacker could brute-force the system by clicking at random points and attempting to authenticate. If the average screen space of a convex hull is taken at 10%, the complexity is 10^5 which gives about 17 bits of guessing entropy. This is weaker than a weak 8-character password [Burr et al. 2004]. Note that even if the number of icons on the screen is increased, the guessing entropy will not increase unless the number of screens is increased, since the icons are still randomly distributed over the screen space.

2.7.2 Serial vs. Unordered Recall

A simple way to gain some of the memory benefits of recognition for a recall-based picture-password system is to remove the requirement that the password be an *ordered* collection of items. This is a task known as *unordered recall*[18]. Allowing users to select their password images in any order greatly reduces the cognitive demand of the password [Sasse et al. 2001] and should result in a lower rate of unsuccessful logins. In Chapter 4, the system is analyzed as if it accepted unordered input and a large increase in successful logins is found.

When unordered input is acceptable, the system can implement hashing with no penalty. User input (a set of selected pictures) could be reordered according to some predetermined ranking, say alphabetical order, and the resulting string could be compared to a single, stored hash. Though the hashing problem is easily dealt with, there is an unavoidable reduction in complexity. In my password system, complexity would fall from 1.17×10^{15} to 2.9×10^{10} as the password space is divided by 8!. This is a general result of removing order from password input, since the complexity of the password must be divided by the number of possible successful inputs. It should also be remembered that even though a system like this provides variability in user input, it does not provide the immunity to shoulder-surfing of an ambiguous authentication system.

2.7.3 Hashing in Ambiguous Authentication Systems

When the server presents an authentication challenge to a user, it must be able to validate the user's response. This is typically done by hashing the user's input and comparing it to some stored value. However, in the case of ambiguous authentication systems, the server's challenge is based on the user's password and is generated in real time. This requires that the server maintain a copy of the user's password; a hash of the password cannot be used

[18] In the psychology parlance of list memory, we are forgiving transposition errors, but not insertion, deletion, or substitution errors.

to generate such a challenge. Therefore, truly ambiguous authentication systems, such as the convex-hull click system [Wiedenbeck et al. 2006], cannot implement hashing when storing user passwords.

2.8 The Distinctiveness Dilemma

Regardless of whether an authentication system is recognition-based or recall-based, the use of a large number of images creates a "distinctiveness dilemma." In this section I present the problem formally.

To begin with, let us reexamine the multiple authentication screen scenario of section 2.5.2, where the user must proceed through L screens (where L is the length of a password) containing N images per screen. In order to maintain complexity comparable to a character-based system, a recognition-based system should not indicate a proper choice immediately. That is, the attacker should not get kicked out of the system when an incorrect choice is made, as this reveals information about the password (The maximum number of tries required to crack the password drops precipitously from N^L to $N \times L$.) Therefore, the potential hacker can peruse all L screens on each attempt. From this it follows that the distractor images cannot repeat over multiple screens, as this reveals additional information which can be used towards cracking the password. Therefore, a system with N pictures per screen, and a password of length L, must contain at least $N \times L$ different pictures[19]. Thus, a multi-screen password system of comparable complexity to my system would require $80 * 8 = 640$ images[20].

Further, it must be assumed that all images are equally likely to be in the user's password. We cannot randomly choose images from outside the set (such as from the internet) to use as distractor images, since attackers would quickly discover this flaw in the password system. Therefore, to have a secure password system, all images in our picture

[19]Note that the proposed Déjà Vu system described above actually requires more images than this.

[20]My system uses a grid of 80 pictures and passwords of length 8. It is explained in detail in Chapter 3.

set are equally likely to be part of the user's password. To have a usable password system, all images should be memorable.

Definitions

Difference Let $D(p_i, p_j) \in [0, 1]$ be a measure of the difference between two pictures, p_i and p_j, where $D = 0$ indicates that p_i and p_j are the same image and $D = 1$ indicates that the two pictures are maximally different. D may be computed based on schematic similarity, conceptual similarity, color, verbal label, and other dimensions as applicable to memorability.

Uniform distribution Further, let us assume D is defined such that given all possible pairs of images p_i and p_j, $D(p_i, p_j)$ is uniformly distributed over the interval $[0, 1]$. The result of this is that two randomly selected images have an equal probability of being similar or different.

Distinctiveness Two images p_i and p_j are said to be "distinctive" with respect to one another if

$$D(p_i, p_j) > S$$

where S is some **threshold of distinctiveness** under which p_i and p_j are confusable. For a picture password system, we want $D(p_i, p_j)$ to be as large as possible for all i, j. This would indicate that the images are all different from one another and should yield the best memory performance [Snodgrass and McCullough 1986; Nelson 1977].

Probability of Distinctiveness Therefore, if p_i and p_j are randomly selected images, the probability that they are "distinctive" with respect to each other is

$$P(D(p_i, p_j) > S) = (1 - S)$$

Image Selection

Now, let us examine the problem of creating a picture collection, C, for a picture-password system by taking one picture, p_i, at a time.

1) Add picture p_0 to the picture set C.

2) Add p_1 to C if p_1 is distinctive with respect to the current items in the picture set. Since $C = \{p_0\}$, we require $D(p_0, p_1) > S$. This has probability $(1 - S)$.

3) Add p_2 to C if p_2 is distinctive with respect to the current items in the picture set. Since $C = \{p_0, p_1\}$, we require $(D(p_0, p_2) > S) \wedge (D(p_1, p_2) > S)$. This has probability $(1 - S)^2$.

...

The probability that an image p_n can be added to C is $(1 - S)^n$. This is an exponentially decreasing function. Though images are usually hand-selected rather than arbitrarily chosen, adding a picture to the picture set still carries with it the difficulty illustrated above. Therefore, increasing the size of C makes creating a set of distinctive pictures exponentially more difficult. For large picture sets, such as the multi-screen password system described above, this leads inevitably to user errors as the user confuses distractor images with those of their own password due to lack of distinctiveness.

Implications

Based on this analysis, it seems that large picture sets are unsuitable for use in graphical password systems. In the proposed Déjà Vu system cited above (section 2.5.2), the user has a portfolio of, say, 20 images. The authors claim the system could operate with 10,000 images [Dhamija and Perrig 2000]. Since the presentation is randomized to avoid the

problems of repeated exposure and intersection attacks, users cannot depend on images being in the same place each time. When the user is asked to choose images from his/her portfolio to authenticate, the distinctiveness dilemma indicates that users will confuse their portfolio images with distractor images that are similar schematically, conceptually, or verbally. Even the 80 images used in my password system may be too many for acceptable performance over the long term.

This problem compounds when multiple passwords are learned. The distractor images would then need to be distinctive from any of the images in any of the picture passwords the user has learned.

Of course, the value of S has not been specified and it is conceivable, given the sheer variety of pictures possible, that acceptable values of S may be extremely small. This means that it would be extremely unlikely for two pictures to be so similar as to be confusable, since the space of possible pictures is so large. Certainly S can be expected to decrease as pictures become more detailed or display resolutions increase. Nevertheless, as $|C|$[21] becomes large it should be obvious that the distinctiveness dilemma will, at the very least, have some negative effect on the performance of picture-password systems.

However, in at least one case, a confirming result was found by Weinshall and Kirkpatrick, where users learned 200 images (out of a total image set of 20,000) and used them as a one-time pad over several authentications [Weinshall and Kirkpatrick 2004]. In the initial experiment, user performance deteriorated after a few months when users became confused by similarities with distractor images. Only when the portfolio images were hand selected and the distractor images tweaked to be sufficiently different did user performance increase to a high level. It may then be suggested that distractor image algorithms could be designed so that user password images are not confusable with them, but this still does not deal with the problem of images from a user's other passwords being confused with a distractor image. It may also open the door to as yet undeveloped

[21] $|C|$ is the size of the image set for a given picture-password system.

distinctiveness attacks, where the authentication screen(s) are analyzed for distinctiveness between images and guesses made based on probability.

However, picture-password systems with a small number of images, such as the PIN-type systems explored by De Angeli, Moncur, and others [De Angeli et al. 2005; Moncur and Leplâtre 2007] are relatively unaffected by this problem. It should be easy to create several small sets of pictures for use in multiple PIN-type password systems for which user performance is extremely high.

Chapter 3

EXPERIMENT METHODOLOGY AND DESIGN

3.1 Research Questions

I wished to tackle several research questions related to password usability. Unfortunately, due to time constraints, only a single study could be performed[1]. The study attempted to answer the following questions regarding picture-based and character-based passwords:

- **Memorability**: Are picture passwords more memorable than character passwords?
- **Entry Time**: Do picture passwords take longer to enter than character passwords?
- **User Satisfaction**: Will users prefer the picture-password system to character passwords?
- **Shoulder Surfing**: How easily can passwords be stolen by an onlooker?

This prompted the use of a between-subjects design involving two groups designated Group I and Group II. Group I was the character password group and Group II was assigned picture passwords.

The study also attempted to answer three questions pertaining specifically to the picture-password system:

- **Awareness**: Will picture password users notice a rearrangement of their pictures after a one-week period?

[1] As will be seen in section 4.1, "Sample Size and Analysis Methods", page 54, this greatly affected the significance of my results.

- **Changed Grid**: Will picture password users be able to enter their password successfully when their pictures are rearranged?

- **Input Method**: If given free choice, will picture-password users prefer the keyboard or an on-screen mouse cursor for item selection?

Group I (n = 8)	Group II (n = 15)	
	IIc 8	IIh 7
8	15	
	IIk 7	IIm 8

Table 3.1: Treatment Groups

A table of groups with group sizes from my study is given in Table 3.1. In order to answer the first two questions about the picture-password system, a two-level within-subjects design was employed with Group II. After one week, Group IIc (changed) was first shown a rearrangement of their pictures and Group IIh (home[2]) was shown their normal arrangement of pictures[3].

Participants were likely to be inexperienced with picture-password systems and needed training with both mouse and keyboard. To prevent the order of input method from affecting user choice, an additional two-level within-subjects design was imposed on Group II. Group IIk was required to use the keyboard then mouse for their first and second inputs respectively, whereas Group IIm was required to first use the mouse then keyboard. Note that this is a separate research question from that of picture rearrangement and a 2x2 factorial design within Group II was not required. Nevertheless,

[2]The reason for the naming scheme here is given in section 3.6.2, "Authentication as Security Indicator", page 45.

[3]This is explained in more detail in section 3.6.2, "Testing of a Random Arrangement of Pictures", page 45.

participants were evenly distributed among all four pairs of the two treatment groups within Group II at time of enrollment.

3.2 Password System Design

I designed a picture-password system for use in my study and a character-password system was designed for use as a control. The systems are complete training and authentication systems, are almost entirely automated, and also collect logging information. Even though picture-based password systems usually have very good results compared to character-based systems, this may be due to differences in training or other inequalities described in the rest of this chapter.

Few existing picture-password systems have tested high-strength passwords. In my study, both character and picture passwords were assigned to participants by randomly selecting eight items from an 80 item set. There were no repeated items in passwords for either group, since this would hamper password memorability [Wickelgren 1965]. The *complexity*[4] of passwords for both groups was 1.17×10^{15} with serial ordering imposed. In my analysis, I also consider the passwords in unordered form, where entering the correct items in any order is counted as a successful login. In unordered form, the complexity of passwords for both groups is 2.9×10^{10}. By avoiding user-selected passwords[5] the entropy for all password forms in my system has been increased to the maximum amount[6].

3.2.1 Multiple Encodings

My picture-based password system is unique in that the password can be described with pictures, characters, and spatial locations. Users are able to enter their password by clicking on pictures with the mouse, or entering the corresponding keys with the

[4] See section 1.2.2, "Complexity", page 3.

[5] See section 2.2, "User-Selected Passwords", page 11.

[6] Password items in my system are not repeated which reduces the complexity of our passwords slightly below maximum, though entropy is still maximized.

Figure 3.1: Picture-Based Authentication System

keyboard. Keys are displayed below each picture with a layout matching that of a standard "qwerty"-style keyboard. The top four rows consist of uppercase characters and characters requiring the Shift key, while the bottom four rows contain the lowercase characters. See Figure 3.1 for a typical authentication screen.

Based on Nelson's sensory-semantic model of memory [Nelson 1977], multiple levels of encoding for an item enhance its memorability. In the optimal case, a user would remember all of the pictures, characters, and spatial locations of their password. If the password is forgotten, it may be reconstructed by the user based on the multiple encodings. For example, if a user typically enters their password with the keyboard, and forgets the fifth item in their password, they may be able to recognize that item's picture and spatial location and determine the missing character. In psychology literature, the process of reconstructing items from partially recalled information is known as **redintegration**.

3.2.2 Application Design

The experiment was run via a Java Web Start application on a webserver maintained by the Department of Computer Science at BGSU[7]. It was designed for a 1024×768 display resolution and ran as a fullscreen application. For larger displays, the main display area was centered on the screen, and the rest of the display was filled with whitespace. The application was used for all stages of the experiment and participants could access the experiment website remotely to complete one of their tasks. Instructions for each stage were delivered by information screens in the application and the experimenter provided additional verbal instructions where necessary. In order to promote better use of security indicators, the application ran with the default Java security manager and users were instructed not to trust the application if it asked for unrestricted access to the system.

[7]Bowling Green State University

Logging

A variety of user information was logged for later analysis. First, system information such as operating system and display resolution was gathered. All typed keys were logged with timestamps using standard Java functions. The timestamps were recorded with the System.getCurrentTimeMillis() function which has a resolution of about 10 ms on most operating systems.

3.3 Experimental Stages

The experiment was conducted over a four-week period. Participants performed their tasks individually within a nine-day period[8]. Participants first came into the lab on *Day 1*, were tested on *Day 2*, and retested on *Day 9*.

Both character and picture-based participants underwent the **same stages** of training and testing on Day 1 and Day 2. Picture-based participants performed an additional task requiring shoulder-surfing resistant input on Day 9 after memorability data had been collected, and also completed an evaluation survey. The series of tasks performed by participants are given in Table 3.2.

Upon enrollment, participants were assigned to one of two groups. Participants 1, 4, 7, etc. were assigned to Group I, while participants 2, 3, 5, 6, etc. were assigned to Group II. Group I received a character-based password (taken from a set of 80 characters), while Group II received a picture-based password (taken from a set of 80 pictures).

Participants were assigned a random identification number that served as their username for the authentication system. In order to maintain confidentiality, all data was logged using this random number. The number was given to participants on a slip of paper and participants were instructed to keep the slip with them for the duration of the study. Participants were instructed not to write down their password at any time.

[8]Participants sometimes took longer than nine days due to scheduling issues.

Task Name	Inputs	Description
		Day 1 - In Lab
Stage 1	2	Practice input while password is shown
Stage 2	4	Learn password interactively
Stage 3	4	Enter password with no assistance
Stage 4		Empty screen / consolidation stage
Stage 5	4	Reenter password with no assistance
		Day 2 - Any Location
Day 2	1	Unsupervised entry performed via website
		Day 9 - Group I - In Lab
Day 9	2	Supervised entry
		Day 9 - Group II - In Lab
Day 9.1	1	Group IIc - Supervised entry performed on randomized grid Group IIh - Supervised entry performed on home grid
Day 9.2	1	Group IIc - Supervised entry performed on home grid Group IIh - Supervised entry performed on randomized grid (if applicable)
SSR Input	1	Shoulder-surfing resistant input task
Evaluation		Evaluation of the picture-password system using Likert-scale responses

Table 3.2: Experiment Tasks

For Group II participants, a seed value was randomly generated which defined the arrangement of pictures on the screen. For all but the final tasks on Day 9, Group II participants always saw the same arrangement of pictures, known as the *home grid"*, while learning their password. The arrangements of the home grids were unique[9].

The participant then proceeded through five stages of learning their password on Day 1. Participants were tested on Day 2, and retested one week later on Day 9. Though all effort was made to require participants to complete the Day 2 and Day 9 tasks at one and eight days after Day 1 respectively, this was often not possible due to scheduling issues.

3.4 Day 1 - Training

Previous studies in picture-password systems always involved a training system [Wiedenbeck et al. 2005; Dhamija and Perrig 2000]. There are two reasons for this:

1. Participants usually have not had experience with a picture-based password system before, and need guidance in its operation.

2. It is necessary to make sure participants have successfully learned their password by some objective criteria before testing retention over several days.

In previous studies, training was accomplished by requiring the user to perform some number of correct inputs without assistance before continuing. My experiment followed the same design, requiring eight complete correct inputs (without assistance) by the end of the participants' first day. However, designing a training system allows us to actively affect the way users learn passwords. It should be noted that participants were never given explicit instructions regarding memory techniques. Instead, they were free to use whatever mnemonic method they wished in remembering their password, outside of required training.

[9] A different seed value was generated for each user.

3.4.1 Day 1 - Stage 1

In this stage, the participant was shown their password at the top of the screen for Group I and on the left side of the screen for Group II due to screen space issues. Two complete correct inputs were required before continuing.

Figure 3.2: Stage 1 - Group II (Pictures)

The picture-based password system allows both keyboard and mouse to be used for input. Group II participants were required to use the mouse and keyboard exclusively for each of their two inputs, and the two conditions were balanced. Group IIm used the mouse for their first input, while Group IIk began with keyboard input. For the rest of the experiment, Group II users could use either mouse or keyboard to enter their password (see section 4.5 for the effect of this). Figure 3.2 shows Stage 1 for Group II participants, where the password could be entered either by clicking on the items in the grid which are shown on the left, or entering the string "EL6nyDvq". Figure 3.3 shows Stage 1 for Group I participants.

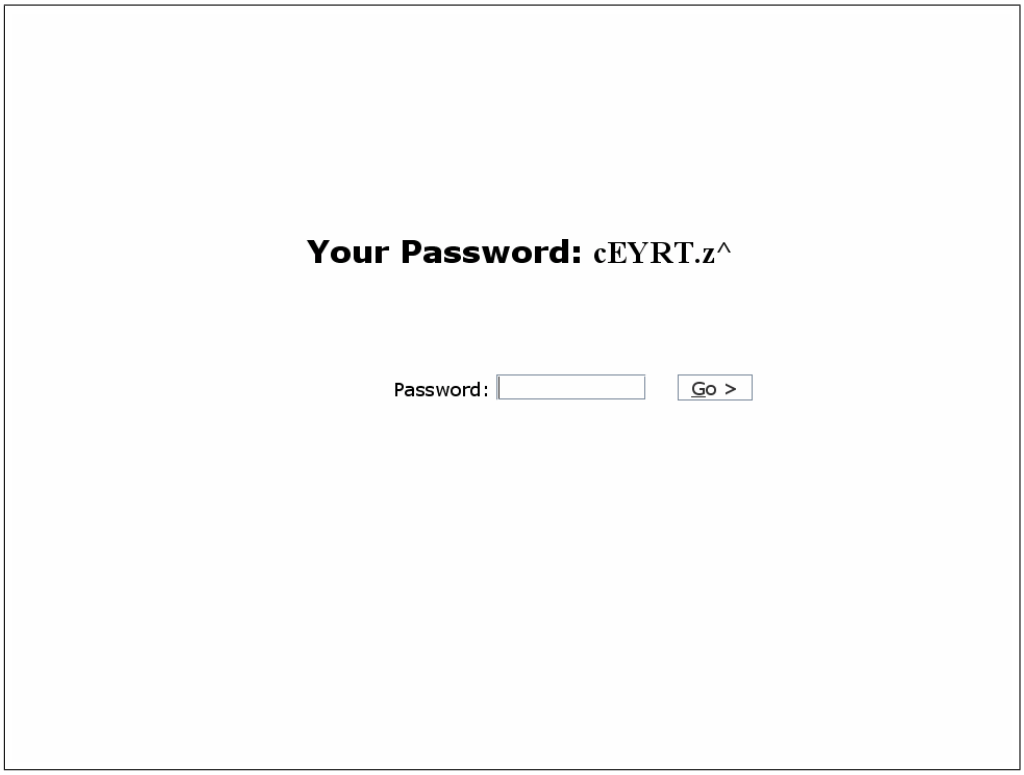


Figure 3.3: Stage 1 - Group I (Characters)

3.4.2 Day 1 - Stage 2 - Interactive Learning

Group II Comtemporary theories of serial learning agree that items are remembered via positional associations rather than through association with previous elements [Anderson and Matessa 1997; Johnson 1991]. In other words, in memory, items in lists are associated with their ordinal position in that list. According to Johnson's model of serial learning, association with other password elements may confound retrieval, if those elements become associated with other ordinal positions. Because of this, only one item in the user's password is displayed at a time in Stage 2. This is illustrated in Figure 3.4.

The black box[10] in Figure 3.4(a) surrounds the **first** item of the user's password. Blank spaces in the grid correspond to other items in the user's password. The yellow box surrounds the location of the **second** item of the user's password. This picture (the "kangaroo") will become visible if the user clicks the "lemon" image or types the

[10]The yellow and black boxes shown in Figure 3.4 are for illustrative purposes only. Boxes did not appear around password items on the participant's screen.

(a) First Item (b) Second Item

Boxes shown for illustrative purposes only.

Figure 3.4: Stage 2 - Pictures

capital letter "E". The result of this is shown in Figure 3.4(b). The "lemon" image has disappeared and the "kangaroo" is seen as the next image in the user's password. All input which does not match the next item in the user's password is ignored. Users are also able to use backspace to view previous items in their password.

Since the participant focused on their password items while inputting them with the on-screen mouse cursor and the keyboard in Stage 1, they should have little difficulty in finding their images in Stage 2 since the other images were never explicitly learned[11]. Once the first image is found, subsequent images can be found interactively by observing the grid and watching for changes.

Various cues to their password are also made visually available to users during this stage:

- The neighboring images to each password item should serve as cues to each item. Based on Johnson's model [Johnson 1991], not including other password items should prevent positional confusion.

[11] This corresponds to a *recognition* task as described in section 2.5.1, "Recognition vs. Recall", page 17.

- Participants must actively watch the grid while learning their password in order to find their password items. This reinforces spatial relationships between password items.

- In addition to providing visual feedback when items are selected, the asterisks in the password field also associate password items with an ordinal position.

Group I Character-password participants were shown a similar screen, with on-screen characters laid out in a standard keyboard arrangement (see Figure 3.5). The operation of this task was the same as that for Group II participants, with only one item of the user's password shown in the grid at one time. As above, this provides an interactive learning method and is intended to reinforce spatial relationships (on the keyboard) between password items.

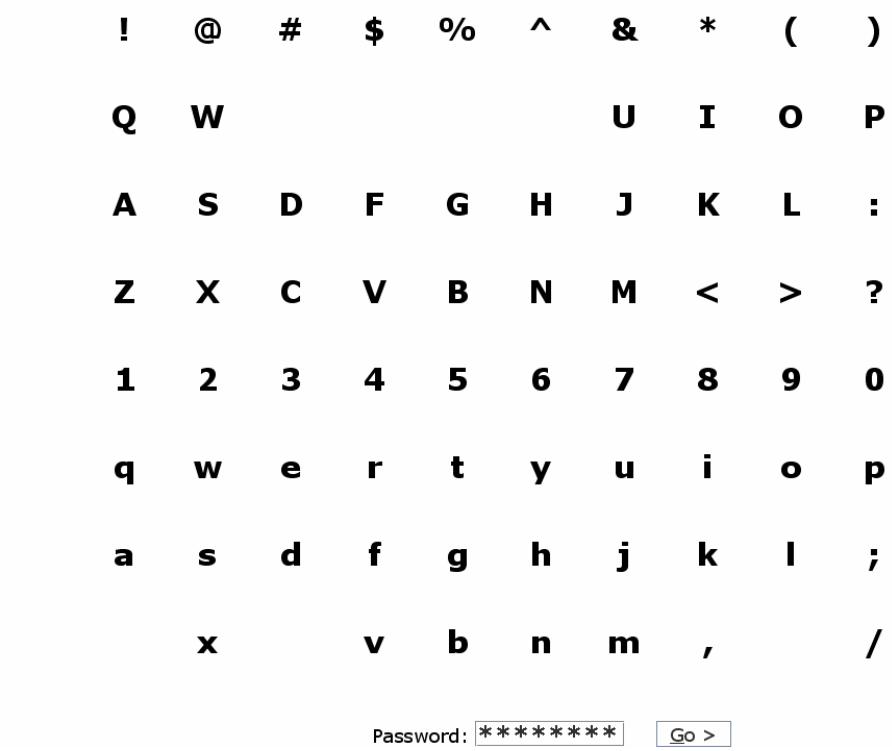
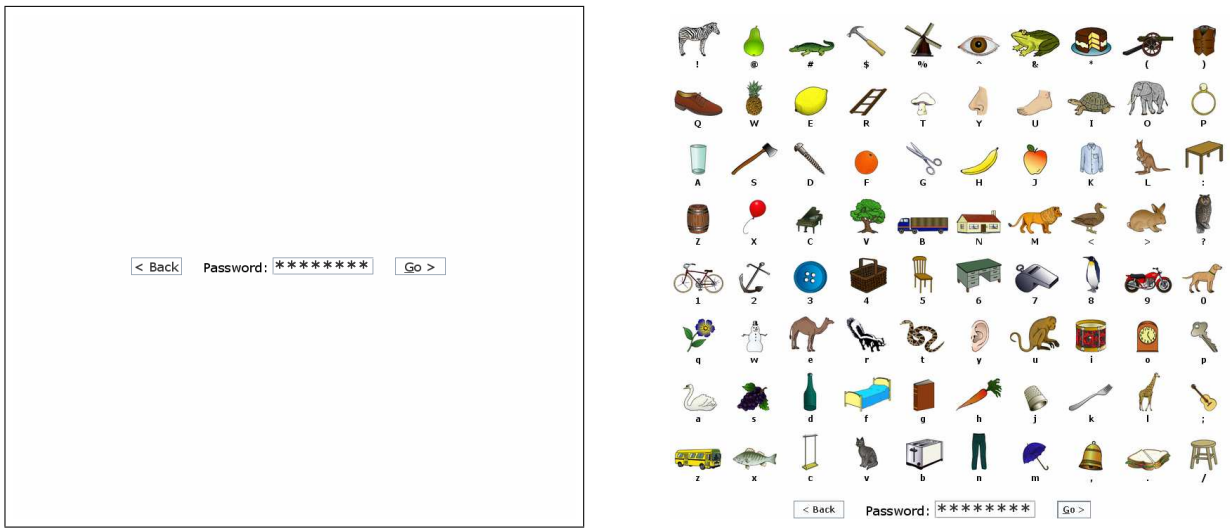


Figure 3.5: Stage 2 - Characters

3.4.3 Day 1 - Stage 3

After the four successful inputs of Stage 2, the participant must enter their password without any hints. Group II participants again see their home grid, as shown in Figure 3.6(b). Four correct inputs are required before continuing to the next task. For both groups, if the user enters five incorrect inputs, or presses the "Back" button, they are returned to Stage 1 but retain their current password. Figure 3.6 shows the authentication screens used for Stage 3. These are typical password screens which are also used in Stage 5, and for tasks on Days 2 and 9.



(a) Group I (Characters) (b) Group II (Pictures)

Figure 3.6: Typical Authentication

3.4.4 Day 1 - Stage 4 - Consolidation

In previous studies of picture passwords, it was noted that users spend more time learning a picture password than a comparable alphanumeric password [Dhamija and Perrig 2000]. This may impact memorability of the password. In a survey article by J.T. Wixted [Wixted 2004], the causes of forgetting were examined in depth. Recent neuroscience studies show that **retroactive interference** is the primary factor in forgetting. Recently formed memories are fragile, and the process by which these memories are strengthened

against forgetting is known as **consolidation**.

Retroactive interference occurs whenever new memories are created. The creation of new memories interferes with the consolidation of recently formed memories, possibly due to limited resources in the brain. It is especially important to prevent retroactive interference immediately after learning. Even though previous picture-based password studies equalized the number of correct input trials completed by participants, no attempt was made to equalize total time spent learning the password. Therefore, it is possible that character-based password users have lower performance for remembering passwords simply because they spend less time learning them and encounter retroactive interference from other sources sooner after learning than picture password users.

In Stage 4, learning times are equalized among participants through use of a non-stimulus. Based on pilot testing, a set time of ten minutes was chosen as an upper bound on the length of time picture-password users would spend learning their password.

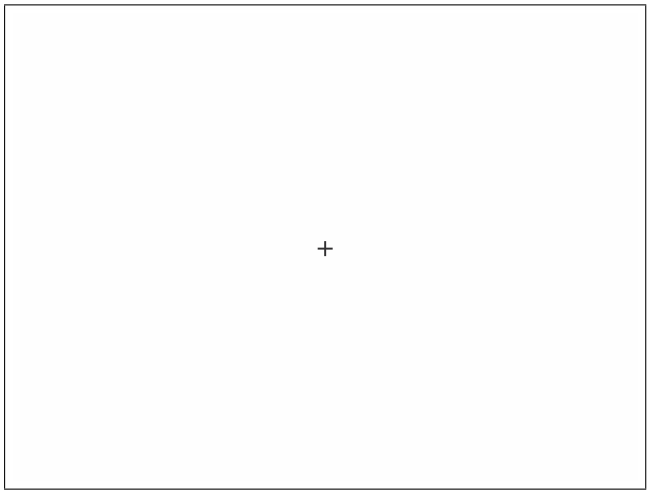


Figure 3.7: Stage 4 - Consolidation

The participant was shown a nearly empty screen with a cross in the center to focus on, as shown in Figure 3.7. Participants viewed this screen until their total time since first seeing their password was ten minutes. Additionally, participants were verbally instructed not to look around the room. For participants that had already spent more than ten minutes with their password, the screen was shown for only ten seconds. This

occurred for 3 out of 15 of our participants. Unfortunately, due to a bug in the program, two participants who clicked the "Back" button in Stage 3 had their times reset and spent much longer than ten seconds in this stage. This topic is revisited in section 4.6.

3.4.5 Day 1 - Stage 5

At the end of Stage 4, the participant was asked for four inputs of their password. Completion of Stage 5 concluded the participant's session on Day 1. The option of returning to Stage 1 was available (by clicking the "Back" button) if the password could not be recalled at this point. None of our participants needed to do so.

3.5 Day 2

Since only one input was required of users on Day 2, participants were not brought into the lab. Instead, participants were sent an email with instructions for completing their task through our department website. Participants who were unable to correctly enter their password in order, within five tries, were not brought back for Day 9.

3.6 Day 9

Group I and Group II participants encountered very different tasks on Day 9 and both groups were videotaped while entering their passwords.

3.6.1 Day 9 - Group I

Character-password participants were asked for two correct inputs on their standard authentication screen (see Figure 3.6(a) on page 42). A video camera was positioned to the right side of the keyboard and zoomed in so that only the "qwerty" section of the keyboard was visible (the arrow keys and number pad were not relevant to password entry.) Though all entries were videotaped, only the second input was intended for

shoulder-surfing analysis. However, since this task was extremely difficult, all inputs were reviewed. See section 4.10 on page 72 for results.

3.6.2 Day 9 - Group II

Group II participants were asked to complete three tasks with the authentication system on Day 9. They were videotaped while performing these tasks, however these tapes were not analyzed because the resolution of the videotape was too low to provide useful information. Instead, insecure behaviors were noted by the experimenter during the experiment. Group II participants also filled out a survey evaluating the picture-password system at the end of the session on Day 9.

Testing of a Random Arrangement of Pictures

As mentioned in section 2.1, the ineffectiveness of current security indicators informed the design of my authentication system. In my system, the arrangement of pictures acts as a security indicator. The arrangement should always be the same when accessed from a trusted client, such as when the user is at home, but will be rearranged if the client is untrusted. *Trust* could be implemented through use of an SSL cookie or other persistent client-side mechanism. Since users learn their password on a single arrangement of pictures (known as the *home* grid), changing the arrangement requires the user to perform a different task. Dhamija et al. [Dhamija et al. 2006] propose that users have "bounded attention", i.e. they ignore security indicators because the indicators are not necessary to the performance of their primary tasks. By making the security indicator part of the primary task of authentication, it should be more difficult for users to ignore.

If users ignore the arrangement of pictures, and enter the same set of keystrokes as required on their home grid, their authentication attempt will fail. This is a desirable result. If a phisher attempts a man-in-the-middle attack on the user's password, they will not have access to the user's home grid, since the phisher is an untrusted client. By

serving the user a changed grid and receiving input for the user's home grid, the phisher cannot reconstruct the user's password. Upon failing authentication, the user may then notice the changed grid, or examine the system for other security indicators. However, if the user is making a legitimate authentication attempt from an untrusted system (such as another computer) and is unable to reconstruct their password, usability of the authentication system is decreased.

In order to test the memorability of a password when the arrangement of pictures changes, Group II users were assigned to one of two groups (alternating between Group IIc and IIh). Group IIc were first shown a randomly generated arrangement of pictures (their *changed* or untrusted grid), while Group IIh were first shown the same set of pictures they trained on (their home grid). Both groups were asked to speak out if they "have any observations about the authentication system." This was meant to determine if participants could discern between their home grid and a changed grid without overt prompting from the investigator[12]. If participants did not speak while performing their tasks, they were asked for their observations after the tasks had been completed.

To reiterate, Group IIc saw the changed grid first, while Group IIh saw their home grid first. On entering their password, the following outcomes were possible:

- Group IIc participants, whether successful or unsuccessful, continued to their home grid for another authentication.
- Group IIh participants who successfully entered their password on their home grid were shown a changed grid and attempted another authentication.
- Group IIh participants who failed to enter their password on their home grid did not continue to the changed grid condition.

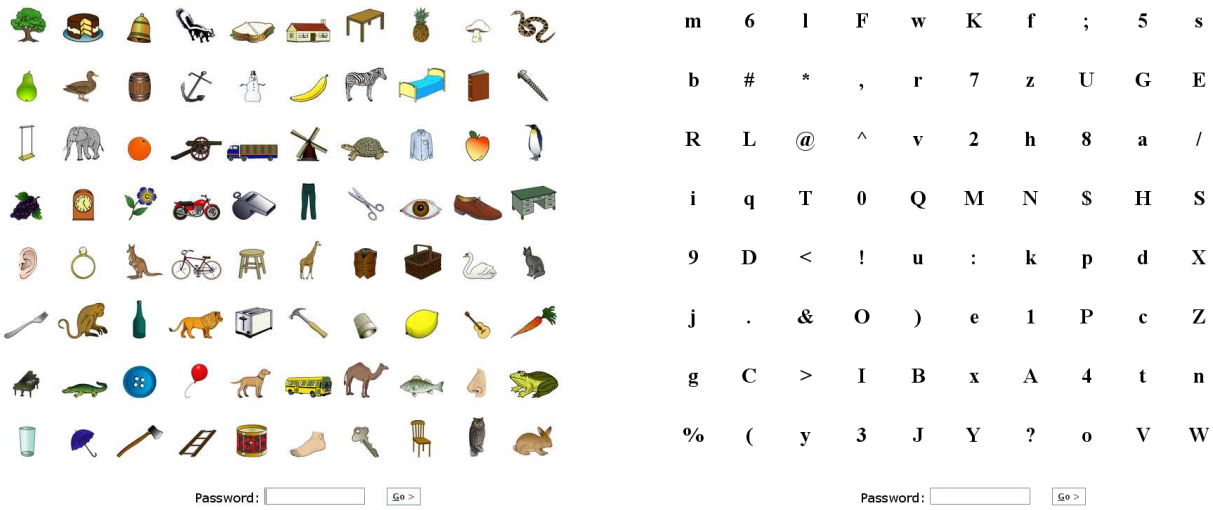
Participants who were unable to correctly enter their password were given a copy of their

[12]This turned out to be problematic and is discussed in section 4.9, "Effectiveness of Picture Arrangement as a Security Indicator", page 72.

password on a sheet of paper. They referred to this sheet while completing the shoulder-surfing resistant input task described below.

Shoulder-Surfing Resistant Input

Section 2.6.1 describes current approaches to shoulder-surfing. My approach is similar to the Spy-resistant Keyboard [Tan et al. 2005] in that a grid of letters changes for each item in the password. Screenshots are shown in Figure 3.8.



(a) Static Grid (b) Ctrl Toggled Dynamic Grid

Figure 3.8: Day 9 - Static Picture Grid and Dynamic Character Grid

Participants were shown a new, changed grid of pictures where letters were no longer displayed beneath each picture (see Figure 3.8(a)). Mouse entry was not enabled for this task, although the mouse pointer was still visible. Participants were required to hold down a toggle switch (Ctrl) to see the keys mapping to the pictures in their password (see Figure 3.8(b)). Participants then needed to press the key which appeared in the same location as their password item. Upon pressing any key, including backspace, the characters reshuffled while the picture grid remained unchanged. This pattern of entry continued until the participant's complete password had been entered.

The rationale behind this is similar to the Spy-resistant Keyboard [Tan et al. 2005] in that screen elements are randomized on each input. It is also similar to the PassFaces

input method [Tari et al. 2006] in that a shoulder-surfer must pay attention to both screen and keyboard at the same time. This makes it extremely difficult for an onlooker to determine which pictures were chosen, since the corresponding keys are randomized. As previously explained in section 2.6.1, the password may still be stolen through use of a recording device.

Evaluation

Finally, participants in Group II were asked to evaluate the picture-password system by filling out a survey. All of the questions used a 3-point or 5-point Likert scale. The survey gathered user perceptions about the picture-password system and also asked users to compare the picture-password system to character-password systems. Questions one through eleven were intended to gather user opinion, while question twelve also gauged user security understanding of an unfamiliar task. Based on research by Nicholls et al. [Nicholls et al. 2006], the Likert scale directions were balanced from left to right. The survey is reproduced in Appendix A.

3.7 Character Set

The character set consists of uppercase and lowercase versions of 40 keys on a standard "qwerty" keyboard. The 40 keys were taken from the 10x4 grid formed by the keys 1 through 0 in the top row and z through / in the bottom row. The full set is given in Figure 3.9.

The same set is used for composing passwords in the character-password system as well as the corresponding keys in the picture-password system. See Figure 3.1 on page 33 to see how keys are mapped to spatial locations in my picture-password system.

```
!   @   #   $   %   ^   &   *   (   )
Q   W   E   R   T   Y   U   I   O   P
A   S   D   F   G   H   J   K   L   :
Z   X   C   V   B   N   M   <   >   ?
1   2   3   4   5   6   7   8   9   0
q   w   e   r   t   y   u   i   o   p
a   s   d   f   g   h   j   k   l   ;
z   x   c   v   b   n   m   ,   .   /
```

Figure 3.9: Character Set for Passwords

3.8 Picture Set

A great deal of effort went into the selection of the picture set for my picture-password system. The selection process is explained below.

A revision of the Snodgrass and Vanderwart set [Snodgrass and Vanderwart 1980] was used as the source of pictures for the experiment. Rossion and Pourtois [Rossion and Pourtois 2004a] improved upon the Snodgrass and Vanderwart set by coloring in and adding greater detail to the objects. They have also made their version of the picture set available online [Rossion and Pourtois 2004b].

The source picture set contained 260 pictures. Since the design of my password system only required 80 images, I was able to select the 80 images from the set which best served the purpose of the experiment. Since the images were to be reduced in size, it was important to choose pictures which were easily recognized as the objects they attempted to represent. One reason for this was to remove polysemy[13]. Snodgrass proposes that one possible reason for the picture superiority effect is the fact that polysemy of words

[13]Polysemy in this context refers to pictures which might have multiple interpretations as to the objects they represent.

prevents access to the semantic concepts of a word. When polysemy is introduced into pictures, such as when picture fragments are used, the picture superiority effect is often reversed [Snodgrass and McCullough 1986; Kinjo and Snodgrass 2000]. By choosing pictures with high agreement scores, it is assumed that these pictures are not polysemous. Otherwise, a substantial proportion of respondents would have labeled them with different names.

Another reason for choosing pictures in this way is to find images which are very good at representing the objects they are meant to represent. Deregowski and Jahoda [Deregowski and Jahoda 1975] found that objects are remembered better than pictures which represent them, and theorized that this may be due to the fact that pictures incompletely represent concepts, while objects exemplify them. Therefore, I decided that choosing pictures from the Snodgrass and Vanderwart set which best represent objects could only have an enhancing effect on memory. It should be noted that McGeorge and Deregowski [Deregowski et al. 1999] advise strongly against assuming pictures will have the same memory effect as objects. However, I believe that in this case, where objects are not available as a stimulus, that it is reasonable to extend their advice to pictures which well represent objects[14].

In addition to choosing images which best represent objects, the picture set was filtered to enhance memory retrieval based on levels-of-processing theory [Nelson 1977] (see section 2.8). Thus, as much as possible, the picture set was chosen to maintain low conceptual, schematic, and verbal similarity among its items. Note that verbal similarity was only considered in English, and all participants were proficient in English to a level required for successful interaction at an English-speaking university.

[14]Weinshall and Kirkpatrick found the Snodgrass and Vanderwart set to perform poorly in their password system, which had previously been using photographs [Weinshall and Kirkpatrick 2004]. This may indicate a need for a new digital picture set using photographs that better represents objects than the images in the Snodgrass and Vanderwart set.

Labeling McGeorge and Deregowski found that the Stroop effect may disrupt performance for labeled stimuli if users focus on labels for memory. The Stroop effect refers to an interference in processing which occurs when objects are labeled incorrectly [Stroop 1935]. For example, if an orange is labeled with the word "apple" there will be a delay in cognitive processing of the item. McGeorge and Deregowski found that incorrectly labeled objects are poorly remembered if users are instructed to focus on the labels. If users are instructed to focus on the objects, and not the labels, performance is not impaired. However, they did not measure performance when users are given no instruction. In my system, the pictures are unlabeled because participants' native language might not have been English. Such users would have labeled objects using words from their native language, and if the labels were in English, this might have caused an impairment in memory performance due to the Stroop effect. Though dual-coding theory might indicate greater memory recall performance if pictures are labeled, Stenberg found no improvement in memory performance when participants were forced to label pictures aloud. This may indicate that users implicitly label pictures anyway, without need for a visual reminder [Stenberg et al. 1995].

Agreement In order to choose the best pictures from the set, data about the pictures was extracted from the original Snodgrass and Vanderwart study and two other studies which used this picture set [Snodgrass and Vanderwart 1980; Stenberg et al. 1995; Rossion and Pourtois 2004a]. In all three studies, data was collected about the agreement of names to pictures. Experiments were conducted in which participants were either asked to name the images, or rate their agreement level with a predetermined label. In addition, the experiments were conducted in English, Swedish, and French respectively. Snodgrass and Vanderwart and Rossion and Pourtois provide exact percentages of agreement, while Stenberg only lists 120 pictures for which agreement was 90% or better. For the 140 pictures for which Stenberg did not provide data, a safe value of 70% was chosen since

it was the average agreement score for those pictures in the Rossion and Pourtois data (Snodgrass and Vanderwart data had an average agreement score of 74% for that subset.) For each picture, the three percentages of agreement were added together. A threshhold of 265 cumulative score was arbitrarily chosen and 104 pictures remained. It should also be noted that Rossion and Pourtois' data was collected based on the revised, colored picture set, while the other two sources used the original line drawings. It is assumed that this would cause a constant increase in picture agreement score across all items. This fact, combined with the use of an assigned value from the Stenberg data, makes the numerical meaning of this threshold unclear, i.e. it should not be interpreted as an average value of 88.3% agreement.

A table of agreement scores obtained from the three data sources [Snodgrass and Vanderwart 1980; Stenberg et al. 1995; Rossion and Pourtois 2004a] is given in Appendix B.

Of the 104 pictures remaining, 24 were removed based on four criteria:

- Pictures which had poor detail or were difficult to recognize at a smaller size, such as the cigar and cap, were removed.
- Pictures which were schematically or conceptually very similar to other objects in the set, such as the fork and spoon; cup, bowl, and glass; had all but one image removed. The image with the highest agreement score among subsets of similar objects was not removed.
- Pictures whose labels were verbally similar in English, such as chair and chain, and screw and screwdriver had all but one image removed.
- Certain images, such as the ant and spider, were removed for purely aesthetic reasons.

Since 80 images remained, but the axe and hammer were still schematically similar, the hammer image was flipped horizontally so that it was oriented perpendicularly to the axe

image. Finally, some stray slightly off-white pixels were cleaned from the “fish”, “nose”, and “dog” images.

Size The pictures were then resized with constraints of 80-pixel width and 64-pixel height. Aspect ratios were maintained during resize, and the pictures had already been cropped in the original picture set. Pictures were not resized when displayed to participants. Instead, the picture-password system was designed for a 1024x768 resolution display.

3.9 Population

Participants were recruited from two sources. Eighteen participants were recruited from computer science classes, and five participants from the Technology Support Center (TSC) at BGSU were invited to participate. The eighteen participants from CS classes were awarded extra credit for participating in the study, while the five participants from the TSC received no compensation. 20 participants were male and 3 were female. Four participants (all from the TSC) were older professionals, while the remaining 19 participants were undergraduate or graduate students.

13 of 15 participants in Group II had never used a picture-based password system before. The other two participants stated that they had rarely before used a picture-password system.

Results of the study are discussed in the following Chapter.

Chapter 4

RESULTS AND ANALYSIS

Though my sample size was too small to produce statistical significance in most cases, picture passwords were always remembered better than character passwords, and required nearly the same amount of time to enter. When password inputs were analyzed without respect to serial order, pictures were significantly more memorable than characters over the full eight-day period. Several other results are also discussed in this chapter including: the effects of keyboard usage, shoulder-surfing topics, evaluation results, and the viability of picture arrangement as a security indicator. Another important result, the phenomenon of repeated incorrect logins, is covered in its own chapter, Chapter 5.

A summary of these results can be found at the end of this chapter, on page 74, where the original research questions[1] are restated with results.

4.1 Sample Size and Analysis Methods

Given the complexity of my experimental design, the study did not have enough statistical power to produce significant results in many situations. Unfortunately, time to perform the study was quite limited, and this prevented recruitment of a larger sample. Where necessary, a post-hoc power analysis was performed using the G*Power 3 [Faul et al. 2007] analysis tool. Also p_{rep}[2] values are given to aid in statistical interpretation. Statistical tests were performed using the R statistical package for Linux.

[1] See section 3.1, "Research Questions", page 30.

[2] The p_{rep} indicates the probability of replicating an experiment's effect [Killeen 2005].

About Fisher's Exact Test For correctness data, Fisher's exact test (two-tailed) was run on the 2x2 result matrices. This test computes exact p values even for unbalanced sample sizes and is a much better alternative to χ^2 for analysis of our data [Fleiss et al. 2003]. The *degrees of freedom* for this test is always 1, so it is not reported in my results.

About Welch's *t*-Test When comparing means between Groups I and II, Welch's t-test (two-tailed) was used to determine whether the difference in means was statistically significant. This test is necessary when comparing means between samples of unequal variance [Milliken and Johnson 2002]. Whenever used in this chapter, the t value, degrees of freedom, and p value are all reported. Note that the *degrees of freedom* for a Welch's test is a real number and usually not an integer.

4.2 Correctness

4.2.1 Ordered vs Unordered Recall

Memorability of a password is measured here based on the participant's ability to enter their password correctly within five tries. After the experiment concluded, input data for both character (Group I) and picture (Group II) participants was parsed to determine how successful participants **would have been** at an unordered input task[3]. It is important to remember that participants were never trained on an unordered input task and were not aware that an unordered input would have been accepted as "correct." The analysis of unordered inputs was not performed until after all participants had completed the study and all data had been collected.

[3]As explained in previous chapters, an unordered input involves entering all password items in **any** order. For example, if the user's password is "EwIg7$cw", "Igcw7Ew$" would be an acceptable unordered input. Similarly, for picture passwords, selecting the correct images in any order is acceptable.

4.2.2 Correctness - Day 2

All 15 Group II participants successfully entered their password in correct serial order on Day 2. Of Group I participants, 2 out of 8 were unable to correctly enter their password within five tries. However, 1 of the 2 participants made only a transposition error in their password and would have succeeded at an unordered input task. A summary of these results is given in Table 4.1 and Figure 4.1.

Ordered Input

	Correct	Incorrect
Group I	6	2
Group II	15	0

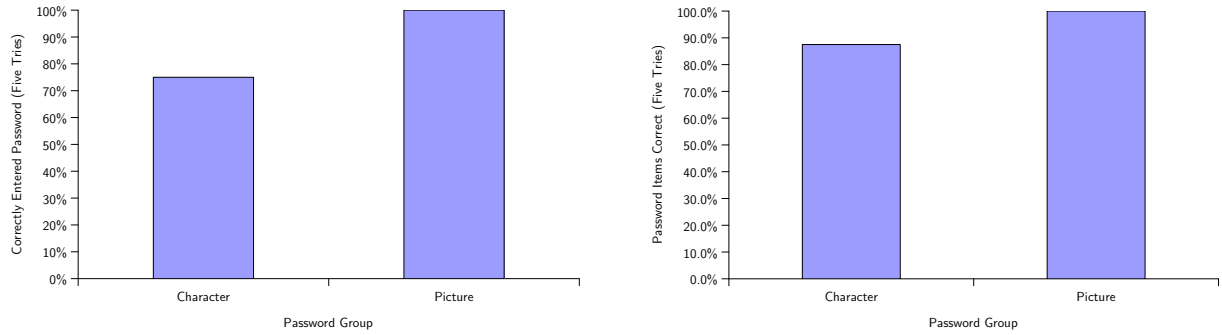
Fisher's:
$p = 0.1107$, $p_{rep} = 80.6\%$

Unordered Input

	Correct	Incorrect
Group I	7	1
Group II	15	0

Fisher's:
$p = 0.3478$, $p_{rep} = 60.9\%$

Table 4.1: Results for Correctness of Input on Day 2



(a) Ordered Input

(b) Unordered Input

Figure 4.1: Percentage of Participants who Correctly Entered their Password on Day 2

Fisher's exact tests were run on the 2×2 matrices of Table 4.1 and the results are shown below each table. Mean time between Day 1 and Day 2 for the two groups is shown in Figure 4.2. The error bars show the standard error of each mean. Character-password participants had a longer interval between tasks than picture-password participants but not significantly so. This was confirmed by Welch's t-test ($t = 0.7275$, df $= 15.729$, p $= 0.4776$). Mean time for both groups was 38.6 hours, which is much longer than the expected 24 hours between Day 1 and Day 2. This was partially due to webserver

downtime for a three day period that pushed back the Day 2 task for five participants (2 in Group I and 3 in Group II).

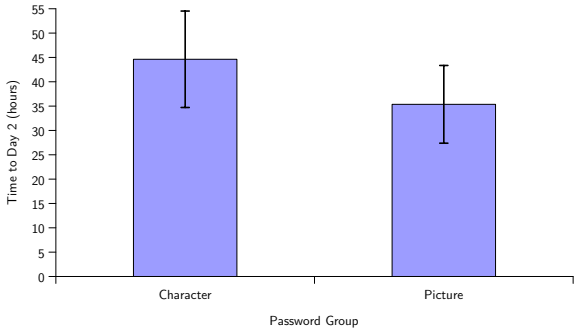


Figure 4.2: Mean Time in Hours between Day 1 and Day 2 with Standard Error Bars

Analysis After a short delay ($\approx$ 39 hours), picture passwords were more memorable than character passwords in both ordered and unordered form, though not significantly so. A post-hoc power analysis shows that the achevied power for these tests was extremely low: 32% for the ordered state, and 7% for the unordered state. This indicates that much larger sample sizes are needed to determine statistical significance. An *a priori* power analysis shows that sample sizes of 48 for the ordered state and 102 for the unordered state would have been sufficient. The p_{rep} for the ordered state suggests that, were the experiment to be repeated, there is an 80.6% probability that picture passwords would again outperform characters. For the unordered state, the p_{rep} of 60.9% is much lower, suggesting that character and picture-based passwords may be equally memorable after one day when accepted in unordered form.

The entry times for the Day 2 task are discussed in section 4.3 on page 61, and a discussion of error recovery is given in section 4.3.1 on page 63.

4.2.3 Correctness - Day 9

Comparison of Groups IIh and IIc

On Day 9, Group IIh was shown their home grid before performing the changed grid task, while Group IIc viewed the changed grid first. There was **no significant effect** of viewing the changed grid first on performance in the home grid condition. 5 out of 8 Group IIc participants successfully entered their password, in correct serial order, on their home grid on Day 9, while 5 out of 7 Group IIh participants were successful. These results are summarized in Table 4.2.

	Correct	Incorrect
Group IIh	5	2
Group IIc	5	3

Table 4.2: Comparison of Treatment Groups on Home Grid

Because of this, Group II is considered as a whole when compared with Group I for correctness.

Comparison of Groups I and II

Character and picture passwords were compared using the *home* grid condition for Group II. A discussion of the *changed* grid condition, in which the participant's pictures were rearranged, is presented later in this section. The two participants from Group I who failed to authenticate on Day 2 were not brought back for Day 9. This might have affected our samples and is discussed in section 4.4, "Missing Participants", page 63.

Only 10 of 15 Group II participants successfully entered their password, in correct serial order, on Day 9 on their home grid. Of Group I participants, 3 out of 6 were unable to correctly enter their password, in correct serial order, within five tries. However, only 1 of the 3 Group I participants would have succeeded at an unordered input task, while the other 2 participants entered characters which were not in their password. Among Group

II participants, all 15 participants chose only the items in their password from their home grid, successfully entering their passwords in unordered form. A summary of these results is given in Table 4.3 and Figure 4.3. More surprisingly, 14 of the 15 Group II participants chose the correct password items on their first attempt, with the remaining participant requiring only two attempts.

Ordered Input

	Correct	Incorrect
Group I	3	3
Group II	10	5

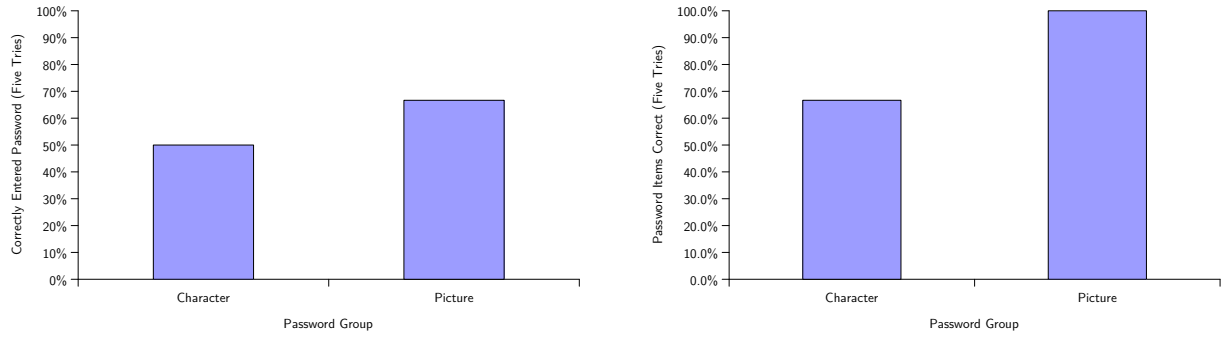
Fisher's:
$p = 0.631$, $p_{rep} = 40.7\%$

Unordered Input

	Correct	Incorrect
Group I	4	2
Group II	15	0

Fisher's:
$p = 0.071$, $p_{rep} = 85.0\%$

Table 4.3: Results for Correctness of Input on Day 9



(a) Ordered Input

(b) Unordered Input

Figure 4.3: Percentage of Participants who Correctly Entered their Password on Day 9

Fisher's exact tests were again run on the 2×2 matrices of Table 4.3 and the results are shown below each table. Mean time between Day 2 and Day 9 for the two groups is shown in Figure 4.4, with error bars showing the standard error of each mean. Here, picture-password participants had a longer interval between tasks than character-password participants, but this difference was not significant (t = -0.9882, df = 6.245, p = 0.3598). Mean time for both groups was 166.1 hours, which is very close to the expected 7 days between Day 2 and Day 9.

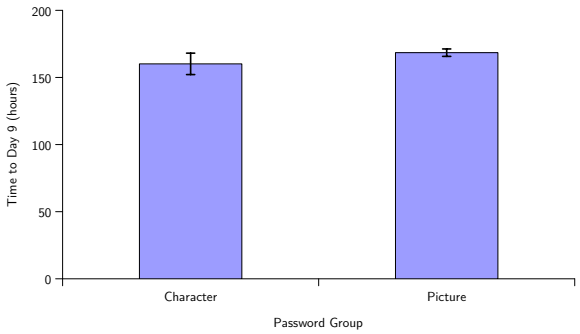


Figure 4.4: Mean Time in Hours between Day 2 and Day 9 with Standard Error Bars

Analysis After a long delay ($\approx$ 1 week), picture passwords were more memorable in both ordered and unordered forms, though again not significantly so. A post-hoc power analysis shows that the achevied power for these tests was only 7% for the ordered state, and 32% for the unordered state. This indicates that much larger sample sizes are needed to determine statistical significance (specifically, 347 for the ordered state and 46 for the unordered state). The p_{rep} for the unordered state suggests that, were the experiment to be repeated, there is an 85% probability that picture passwords would again outperform characters. This could be considered marginally significant ($p \approx 0.07$). For the ordered state, the p_{rep} of 40.7% is much lower, suggesting that character and picture-based passwords may be equally memorable in ordered form after one week.

Changed Grid

As explained in section 3.6.2, Group IIh participants who were unable to enter their password on their home grid did not continue to the changed grid condition. Therefore, the 5 out of 15 participants who were unsuccessful in entering their password on their home grid are excluded from the following analysis. Three of these participants were from Group IIc, and two were from Group IIh.

Of the 10 Group II participants who were able to successfully enter their password on their home grid, only 7 were able to enter their password on the changed grid. An additional participant would have succeeded at an unordered input. These results are

shown in Table 4.4.

	Correct	Incorrect
Ordered	7	3
Unordered	8	2

Table 4.4: Correctness of Input on the Changed Grid

Interestingly, 2 of the 3 participants who failed on the changed grid condition were "heavy" keyboard users. This is discussed in section 4.5.

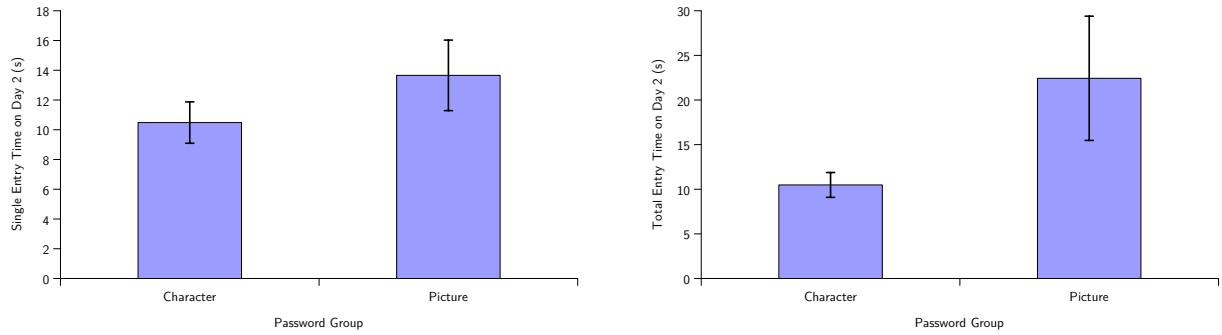
4.3 Comparison of Entry Times

A comparison of entry times was made based on data from Day 2 since this task was the most alike to a typical authentication scenario, where participants experienced a short delay ($\approx 39 hours$) since last entering their password. Entry times were calculated as the difference in time between the first item entry and the last. Even though times were recorded in milliseconds in Java, their values were rounded to the nearest tenth of a second before analysis since actual precision is around 10 ms (see section 3.2.2).

Two measures are compared: *single* entry time, and *total* entry time. **Single** entry time is the time to enter a single, correct input. This includes corrections, like hitting backspace, but does not include incomplete or incorrect inputs (both of which blank out the password entry box.) **Total** entry time is the total time the user spent authenticating, including incorrect inputs.

Two Group I participants were removed from the data because they were unable to authenticate successfully. Two Group II participants were removed from the data for having unusually long entry times. One participant had a single entry time of 87 seconds, while another had a total entry time of 211 seconds. These two values were 8.3 and 7.5 standard deviations from the mean respectively. Since Day 2's task was performed remotely, it is reasonable to assume that these participants were either not concerned

with completing their task or otherwise preoccupied. The next largest entry time of 39.5 seconds was retained in the data set (2.9 σ).



(a) Single Entry (b) Total Entry

Figure 4.5: Mean Entry Times in Seconds with Standard Error Bars

Mean entry times are given in Figure 4.5 with standard error bars displayed for each mean. For single entry, the mean times were 10.5 s for characters and 13.7 s for pictures. For total entry, the mean times were 10.5 s and 22.4 s for characters and pictures respectively. Minimum single entry time was 6.6 s and 5.3 s for characters and pictures respectively.

Analysis Statistical analysis of this data is inconclusive due to the small sample size. Welch's t-tests were performed for single entry times (t = -1.1547, df = 17.958, p = 0.2633) and total entry times (t = -1.6834, df = 12.928, p = 0.1163) which suggest that mean entry times between groups were not significantly different. Even with Group II outliers reintroduced into the data, the mean entry times were still not significantly different by Welch's t-test, single (t = -1.4536, df = 15.732, p = 0.1657) or total (t = -2.0045, df = 14.261, p = 0.06438), though the mean times for Group II become substantially larger. A power analysis on this data was not available because G*Power does not support a power analysis of the Welch's t-test. Therefore, there may not be sufficient evidence to accept the null hypothesis that the means of both groups are equal. However, the fact that the fastest entry time for all participants was a picture

password suggests that variance in picture entry time is large enough that there may be no significant difference between groups.

4.3.1 Error Recovery

Another interesting result was found in the analysis of the entry time data. Notice that the single and total entry times for characters are equal. The Group I participants who successfully authenticated on Day 2 made no incomplete or incorrect submissions. They correctly entered their password on the first try. 3 out of 15 Group II participants made transposition errors in their input, but were able to recover and successfully authenticate. Of the two Group I participants who were unsuccessful, one made a transposition error but was unable to recover successfully. It is likely that, due to multiple encodings (section 3.2.1), users are better able to recover their password once an error has been made when using a picture-based password system than a character-based one. The implications of this are explored in the following section.

4.4 Missing Participants

An alternate analysis of Day 9 is presented here. By excluding the two participants who failed to authenticate on Day 2, the population of Group I has been altered to only include high-performing participants. As explained in the previous section (4.3.1), low-performing participants from Group II may have continued to Day 9 because the design of the picture-password system enabled them to redintegrate[4] their password. Even though this aided participants on Day 2, by Day 9 too much of the password had been forgotten because a single trial was not enough to strengthen the password in memory. If the group composition was made unequal by eliminating participants on Day 2, it is better to compare aggregate performance of the groups over the full 8-day period.

In the ordered case we can simply carry over the two participants as incorrect

[4]Redintegration is the reconstruction of a partially remembered item through multiple cues.

scores on Day 9. Failure at a memory task on Day 2 implies failure on Day 9 because participants were not reminded of their password. However, for the unordered case it is trickier since one of the two participants succeeded at an unordered input on Day 2. This participant is treated favorably and is carried over as a correct score, while the other participant is carried over as an incorrect score. The aggregate data is presented in Table 4.5 and Figure 4.6 (compare this to Table 4.3 on page 59.)

Ordered Input

	Correct	Incorrect
Group I	3	5
Group II	10	5

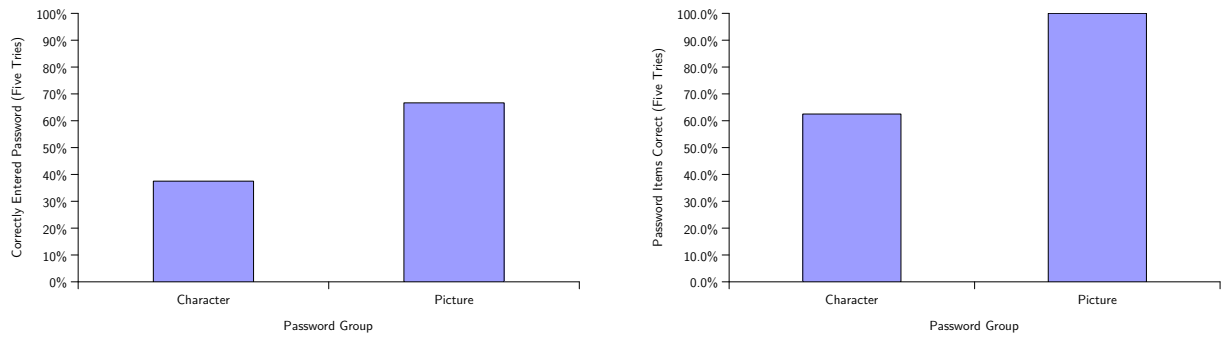
Fisher's:
$p = 0.2213$, $p_{rep} = 70.6\%$

Unordered Input

	Correct	Incorrect
Group I	5	3
Group II	15	0

Fisher's:
$p = 0.03162$, $p_{rep} = 90.5\%$

Table 4.5: Results for Password Memorability over 8-Day Period



(a) Ordered Input*

(b) Unordered Input*

*Data for missing participants included (see text for details)

Figure 4.6: Percentage of Participants who Remembered their Password by Day 9

Analysis In the aggregate, picture passwords appear to be significantly ($p < 0.05$, Fisher's exact test) more memorable than character passwords when accepted in unordered form. However, performance for both groups was not significantly different when ordering was required. Possible implications of this are discussed at the conclusion of this chapter (4.11).

4.5 Keyboard Usage in the Picture-Password System

Effect of Stage 1

During Day 1, Stage 1, Group IIk participants were required to use the keyboard exclusively on their first input, and Group IIm participants were required to use the mouse. Use of keyboard first, or mouse first, appears to have had **no effect** on whether or not participants used the keyboard on subsequent inputs. 4 of 15 participants chose to use the keyboard beyond Stage I. Of these four, two were from Group IIm, and two were from Group IIk.

Effect on Correctness

Of the four Group II participants who used the keyboard beyond Stage 1, three had keyboard usage above 50%, while the remaining participant stopped using the keyboard after Stage 3. This participant was able to correctly enter their password on both their home and changed grids. Of the three "heavy" keyboard users, two were able to enter their password in correct order on Day 9. Since this ratio (2:1) is the same as that for non-keyboard users, there appears to have been **no effect** of heavy keyboard use on password memorability.

This may appear to contradict the "multiple encodings" hypothesis (see section 3.2.1). However, the one participant who stopped using the keyboard after Stage 3 was successful on both the home and changed grids. All three "heavy" keyboard users were unable to enter their password on the changed grid[5], suggesting that they paid more attention to learning their characters than their pictures.

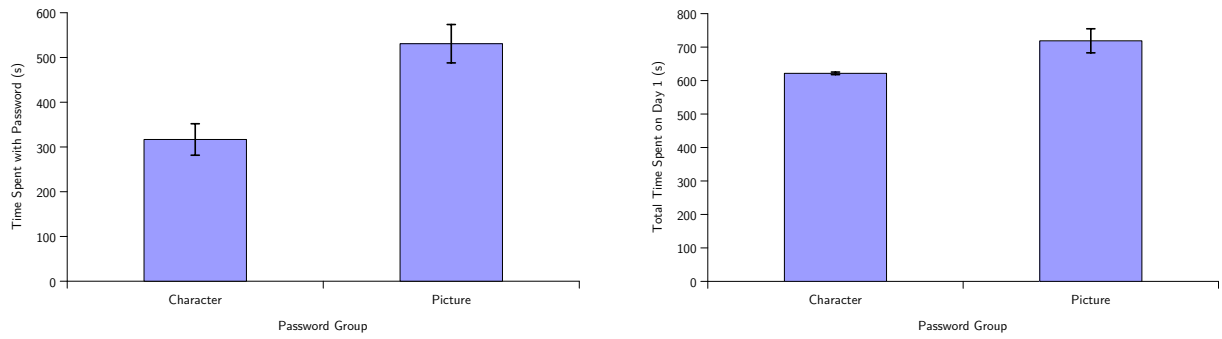
[5]As described above, one of these users was also unable to enter their password on their home grid.

4.6 Password Learning Times

Learning times of participants are given in Table 4.6 and Figure 4.7. Total times were calculated from when the password was first seen to when the last successful input was entered at the end of Day 1.

	Group I	Group II
Mean Total Time Spent on Day 1	622	719
Mean Time Spent on Stage 4 (Consolidation)	304	188
Mean Time Spent with Password on Day 1[6]	317	531
Number of Participants who Exceeded Preset Learning Time (600 s)[7]	0	4

Table 4.6: Mean Learning Times by Group (in seconds)



(a) Time Spent with Password on Day 1

(b) Total Time Spent on Day 1

Figure 4.7: Mean Learning Times in Seconds with Standard Error Bars

As described in section 3.4.4 on page 42, learning times between picture-password and character-password participants were equalized in Stage 4 by asking participants to view an empty screen until a preset threshold time of 10 minutes had been reached.

As shown in Figure 4.7(b), the difference between mean total learning times of character and picture-based passwords was significant. This was confirmed with a Welch's t-test (t = -2.6885, df = 14.318, $p < 0.02$). Even with the equalization effect of Stage 4,

[6]Difference between Total Time and Stage 4 Time.

[7]This does **not** include participants who exceeded the preset learning time due to program logic error.

Group II users spent significantly more time learning their passwords. This might have been a factor in the superior memorability of picture passwords reported in my study.

Effect of Program Logic Error Two of the Group II participants used the back button to review their password at Stage 3, which retrained them on their password. Due to an error in the experiment software, these participants spent longer in Stage 4 than they should have, which increased their total time on day 1 to 964 and 975 seconds. However, one Group II participant did not use the back button, and spent 987 seconds total learning their password. During pilot testing, two participants required over 1000 seconds of learning time. Therefore, the two participants' times, though artificially long, are not outside the normal range of learning times for the picture password system. Even when these two participants' times are removed from the analysis, Group II still took significantly longer to learn their password.

Need for Stage 4

Figure 4.7(a) shows the total learning times without the additional time required by Stage 4. The difference in learning times was extremely significant, as confirmed by a Welch's t-test (t = -3.8595, df = 20.565, p $<$ 0.001). Picture-password participants spent 67.5% longer learning their passwords than character-password participants.

As explained in section 3.4.4, the intention of Stage 4 was to equalize learning times between groups. One problem here is the small variance in total times of Group I ($\sigma \approx 11$ seconds) due to much faster learning. The maximum time spent by a Group I participant, not including Stage 4, was 478 seconds. All Group I users ended Stage 4 at the 600 second mark, at which point they entered their password at Stage 5 and were finished. However, 5 of the 15 Group II participants required more than 600 seconds to reach Stage 4, with a corresponding increase in mean time. In order to make the difference in learning times insignificant, the mean time of Group I would need to be increased by about 80 seconds.

If, as shown in Figure 4.7(b), learning times between character and picture-based passwords were significantly different, did Stage 4 have no effect? I don't believe so. As stated above, two Group II participants needed to use the back button to review their password at Stage 3, while **no** Group I participants needed to do so. This seems to be a violation of the picture superiority effect. However, as studied by Baddeley, the "size" of short term memory is approximately two seconds [Anderson and Matessa 1997]. While an 8-character password may be recited in under two seconds, perhaps an 8-word password is too long to maintain in short-term memory. Without the long break provided by Stage 4, Group I participants might have kept their password in short-term memory for the duration of Day 1's tasks, and hampered their long-term performance.

	Correlation (r_{pb}) With:	
	Time Spent with Password	Consolidation Time
Group I (n = 8)	-0.30	0.37
Group II (n = 15)	0.30	-0.36

Table 4.7: Correlation between Time Spent with Password and Correctness (Day 9)

Correlation with Correctness Point-biserial coefficients[8] were calculated for Group I and Group II to determine the strength and direction of correlation between the time participants spent with their password (without Stage 4) and consolidation time (Stage 4 time only) with correctness over the 8-day period[9]. The results are shown in Table 4.7. Both Group I and Group II show moderate correlations, but in opposite directions.

For Group I (characters), the correlation for time spent with password is moderately negative (r_{pb} = -0.30). This indicates that those participants who learned their passwords more quickly also remembered them better. For Group II (pictures), the opposite effect is seen. The correlation for time spent with password is moderately positive

[8]The point-biserial coefficient (r_{pb}) is used for measuring correlation between quantitative data and a dichotomous variable.

[9]Correctness in **ordered** form for all 23 participants was used. The data can be found in section 4.4, "Missing Participants", on page 64.

($r_{pb} = 0.30$), indicating that participants who spent longer with their passwords learned them better.

However, the strongest correlation is seen between Group I's consolidation time and their correctness ($r_{pb} = 0.37$). This seems to suggest that memory for character passwords was strengthened by participants' Stage 4 time, supporting the hypothesis given above that Stage 4 benefitted character-password participants. However, it is dangerous to make such inferences with insufficient evidence, and this hypothesis needs to be tested with further research.

4.7 Evaluation Results

As described in section 3.6.2, 15 Group II participants filled out a survey upon completion of their computer-based tasks. The following results were found:

- Most participants enjoyed using the mouse for authentication. The mean, median, and mode response to question two was "I was **satisfied** when using the mouse to enter my password." The mean and median response to question three was "I was **neither satisfied nor unsatisfied** when using the keyboard to enter my password."

- Participants felt that entering passwords on their home grid was **easy** and time spent was **short**. These were the mean, median, and mode responses to questions six and eight.

- Participants found entering their passwords on a changed grid **difficult** and time spent was **too long**. These were the mean, median, and mode responses to questions seven and nine.

- Participants felt that time spent performing the shoulder-surfing resistant input task was **too long**. This was the mean, median, and mode response to question eleven.

- Most participants felt the shoulder-surfing resistant input was **less secure** than a character password. This was the median and mode response. The mean response was **equally secure**.

- Participants generally preferred using the picture password system to a character password system, but there was high variance in the responses. The mean response to question four was "I found the picture password system to be **more satisfying** than a character-based password system.", and the median and mode response was **a lot more satisfying**. However, for question five the mean response was "I found the picture password system to be **neither more nor less efficient** than a character-based password system.", though the median and mode response was **more efficient**. Three out of 15 participants responded **a lot less efficient**, and all three participants were keyboard users.

Analysis Though participants agreed that entering their password was easy on their home grid, and the time spent was short, comparisons to a character-password system were not that favorable. A possible reason for this is given below:

On the changed grid, users must first find their password items on screen and then select them. Keyboard users face an additional hurdle, because they must then type the corresponding key to their password item. This task is simply easier to accomplish when using the mouse, because it does not require the extra step of finding and typing a corresponding key. Consequently, keyboard users negatively evaluated the picture password system, with 3 out of 4 responding that it was a lot less efficient than a character-password system.

Participants were not given a complete understanding of the picture password system and its context. After completing the survey, 2 of the 15 participants verbally expressed to me their disappointment with the picture-password system because of the changed grid task. If some participants thought the changed grid task was the standard

mode of operation of the picture-password system, it would explain the variance in responses when it was compared to a character-password system.

A discussion of the evaluation of the shoulder-surfing resistant input task is deferred to the next section.

4.8 Shoulder-Surfing Resistant Input (SSR)

All Group II participants completed a shoulder-surfing resistant input as described in section 3.6.2. The main purpose of this task was to observe an SSR[10] authentication system in operation, and allow users to evaluate it. Two important results are given below:

1. In evaluating the SSR input task, 9 out of 15 participants (60%) responded that it was **less secure** or **equally secure** to a character-password system. Unlike the other survey questions, I believe this question has a correct answer. Since participants were forced to use the keyboard, the task is resistant to shoulder-surfing. It is more secure than a character-password system.

2. Further, 6 out of 15 participants used the mouse while performing the SSR input task. Though on-screen cursor selection of pictures was disabled, the mouse pointer was still active. These participants used the on-screen mouse cursor to keep track of their password items while pressing Ctrl to see the corresponding keys. Of these 6 participants, 5 gave the **less secure** or **equally secure** response mentioned above. One additional participant did not use the mouse, but used their finger to point at the screen. However, the finger was far enough away from the screen that no information about the participant's password could be stolen.

Both of these items confirm the notion that many users have a poor understanding of security. This is not a new result and was thoroughly discussed in section 2.1. However, while item #1 might be expected or correctable only with substantial education about

[10]Shoulder-Surfing Resistant - This abbreviation will be used throughout this section.

security, item #2 may be correctable by better design. This is covered in Chapter 6, "Future Work", section 6.2.2 on page 95.

4.9 Effectiveness of Picture Arrangement as a Security Indicator

Testing the effectiveness of picture arrangement as a security indicator was highly problematic. Group II participants were instructed to "let me [the experimenter] know if you have any observations about the authentication system." The purpose of the instruction was to determine if participants would notice, after a one week delay, that their picture arrangement had changed. Unfortunately, it was difficult, if not impossible, to deliver the instruction without biasing participants. This is a similar problem to those discussed in the experiments of section 2.1, in which experiment participants become unusually paranoid about security when overtly prompted.

If a participant did not initially speak upon seeing the changed grid, they were prompted for their observations after completing the task. All Group II participants correctly identified the changed grid either during the task or afterward. However, only 8 out of 15 participants expressed this verbally during the task. Of these, 4 participants also offered other observations such as "The letters are switched around" and "This window wasn't there before" (in reference to the Internet Explorer Downloads window), which were false. Participants who did not speak out about the changed grid also did not speak up with any other observations. It is possible that these participants did not realize they were expected to verbalize their observations as soon as observed.

4.10 Shoulder Surfing of Character-based Passwords

For the three Group I participants who correctly entered their password on Day 9, the mean time for their first correct entry was 7.1 seconds (one prior incorrect entry by one participant was ignored.) On second correct entry, the mean time reduced to 5.3 seconds.

On review of the videotape, I was unable to decipher the participants' passwords even after five viewings of all inputs. Due to speed of entry, random nature of the assigned passwords, and obscured view of the keyboard by the participants' hands, I found strong passwords to be very difficult to steal by shoulder-surfing. This contradicts a previous finding [Tari et al. 2006].

In the study done by Tari et al, randomly generated passwords were used but were not case-sensitive. In watching the tape, I found that paying attention to the current shift state while also watching the participant's keystrokes was extremely challenging. Users can change shift state through very small movements while also moving towards other keys.

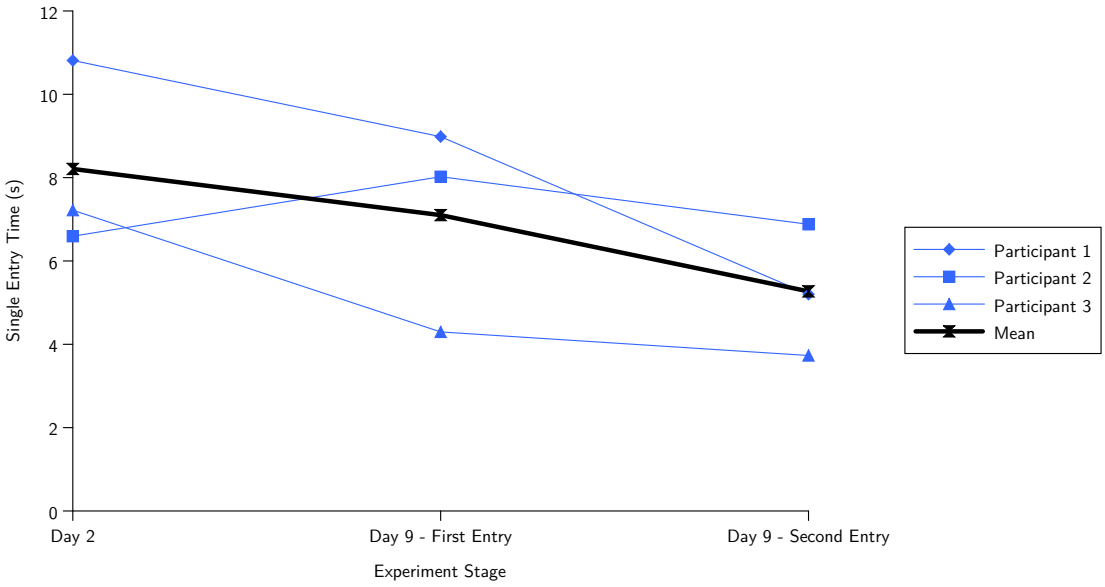


Figure 4.8: Mean Time in seconds of Password Entry for Group I Participants

Further, the speed of entry was much faster than expected. Typing speed gradually improved from Day 2 to Day 9, as shown in Figure 4.8. All three participants were from an upper-level computer science class, so it is possible that their typing speed was atypically high due to greater experience with typing special characters. However, this is an assertion which would need to be supported by further study. Password entry speeds may not be atypical and simply a result of practice.

4.11 Conclusions

Research Questions

At the beginning of Chapter 3, seven research questions were stated. They are reproduced here in the same order, along with summary results.

Memorability: Are picture passwords more memorable than character passwords?

Across all conditions, picture passwords were more memorable than character passwords, though the difference was not significant in most cases. When input data was analyzed to determine how well participants would have performed at an unordered input task, picture passwords were significantly more memorable than character passwords over the entire eight-day period: 100% recall vs 62.5% respectively[11].

Entry Time: Do picture passwords take longer to enter than character passwords?

Picture passwords took longer to enter than character passwords, although the difference was not significant. For a single password input, an average entry time of 10.5 s for character passwords and 13.7 s for picture passwords was recorded. For total entry time, including incorrect inputs, the average entry time was 10.5 s and 22.4 s for characters and pictures respectively.

User Satisfaction: Will users prefer the picture-password system to character passwords?

Generally, participants favored the picture-password system. Concern about entering their passwords on a changed grid might have prevented participants from favoring the picture-

[11] See section 4.4, "Missing Participants", page 63 for an explanation of how data was extrapolated over the eight-day period.

password system more strongly.

Shoulder Surfing: How easily can passwords be stolen by an onlooker?

Character passwords cannot be easily stolen by an untrained[12] onlooker.

Awareness: Will picture password users notice a rearrangement of their pictures after a one-week period?

All picture-password participants correctly identified the rearrangement after seeing both their original and rearranged grids. Only 8 of 15 participants expressed this verbally, though the other 7 might not have been aware of the need for immediate feedback.

Changed Grid: Will picture password users be able to enter their password successfully when their pictures are rearranged?

7 of 10 participants were able to enter their passwords on a changed grid in correct serial order. An additional participant correctly entered their password in unordered form. 2 of the 3 participants who failed at the changed grid condition were heavy keyboard users, which might suggest a need to focus users more on their pictures than on keys or spatial locations.

Input Method: If given free choice, will picture-password users prefer to use the keyboard or the mouse for item selection?

Picture-password users prefer to use the mouse. Only 4 out of 15 participants chose to use the keyboard for item selection.

It was difficult to achieve statistical significance at my sample size, but it is apparent that the picture-password system performs well, especially when unordered input is

[12]Some researchers have speculated that a trained shoulder surfer would be able to steal a password much more effectively than an untrained investigator [Tari et al. 2006]. See section 6.3.2 on page 97 for more information.

accepted. 100% memorability was acheived for passwords with 2.9×10^{10} possible values giving ≈ 35 bits of entropy. Such a system could be implemented with hashing as described in section 2.7.2. However, it is unknown if these results can be replicated when participants are not required to retain serial information[13].

Unfortunately, both character and picture passwords had poor performance after one week in the ordered condition, with 2 character-password participants failing to authenticate after only one day. It may be that randomly generated passwords of this complexity are simply too difficult to remember. Unless memorability can be improved, perhaps by reducing password size, ordered input appears to be infeasible for this (or perhaps any) password system.

[13]This is discussed further in section 6.2.1, "Unordered Input", page 93.

Chapter 5

PROTOCOL TO IGNORE DUPLICATES OF INCORRECT PASSWORDS

5.1 Introduction

One thing I noticed while analyzing the results of the password study was that users often enter the same incorrect password several times. This occurred during **individual sessions**, and is not simply a result of aggregating inputs over multiple authentications. In my study, 14 of 23 participants made more than one incorrect input in a single session. Of the total number of incorrect inputs made by these users, an average of 29% were repeats from the **same** session. Three of these participants made three repeated incorrect inputs in a single session.

User 1	User 2	User 3
P1	P1	P1
P1	P1	P2
P1	P1	P1
P1	P2	P2
P2	P1	P2

P1 and P2 represent the two distinct guesses made by each user.

Table 5.1: Observed Repeated Inputs by Three Participants

For these participants, of the five tries available to each participant before being rejected by the authentication system, four consisted of the exact same incorrect string. In two cases, these inputs were non-consecutive. The pattern of inputs entered by these three

participants is shown in Table 5.1.

Unfortunately, this result was unexpected, and user rationale for this behavior was not studied. One possible rationale is discussed here:

When a password is incorrectly remembered, users might be quite sure of their password, and unaware of their incorrectness. Four of our participants expressed verbal disbelief that their password was being remembered incorrectly and one even asked the experimenter if their password had been changed. From this, and the pattern of inputs shown in Table 5.1, it can be assumed that users repeatedly enter the same incorrect password because they believe it to be correct. They might believe that a typo prevented acceptance of the input, or the system made a mistake and would accept the password if offered a second time.

Whatever the reasoning, after repeatedly inputting an incorrect string a few times, users will eventually move on to other guesses, sometimes hitting upon the correct password. This is not a bad thing, as users do not have perfect recall (or perfect typing skills) and allowing multiple guesses at entering a password increases password usability [Sasse et al. 2001]. At some point, the user will be *locked out* and unable to attempt more guesses.

5.1.1 Account Lockout

Lockout may refer to a complete inability to login to the account, or an excessive delay required by the server between authentication attempts. This occurs after a set number of failed logins have been recorded. For example, Microsoft recommends that accounts should be locked out after 10 failed login attempts for a duration of 30 minutes[1] [Microsoft TechNet]. The primary purpose of lockout is to repel attackers who are attempting to brute-force the password. In the Microsoft example, an attacker can make ten guesses at the password before being locked out for 30 minutes.

[1]This is the "medium security" recommendation.

However, when users enter the same incorrect input repeatedly, they use up available guesses unnecessarily. This can cause the user's account to be locked out very quickly. In the examples shown in Table 5.1, each user was locked out after five tries, but only managed to attempt two guesses. Password usability would be improved if duplicate input in a single session was ignored, since users would then receive their full complement of guesses.

5.2 The Temporary Incorrect Input List (TIIL)

5.2.1 Simple Implementation

Therefore, I recommend that all password systems store hashes[2] of incorrect inputs temporarily and do not increment the incorrect tries counter when a matching input is re-entered. For security, this should be done silently, with no indication to the user (or a potential attacker) that a previously entered input was matched. Separate TIILs would be maintained for each user, and the incorrect input hashes would only be stored for the current authentication session. Since only those inputs matching previously incorrect inputs are discarded, the number of inputs that an attacker could make will normally not be affected. Resistance to brute-force attack, or simple password guessing, is nearly uncompromised. Usability is improved, since legitimate users can make more attempts to log in before getting locked out.

5.2.2 TIIL Attack

However, a slight vulnerability does exist in such a system, since the system is now storing more information about user input. An attacker who connects to the system while there are items in the TIIL might initiate a brute-force attack on the password. If the user has submitted an incorrect input to the system, and an attacker happens to guess the same

[2]See section 1.2.6, "Hashing", page 5.

input while attempting a brute-force attack, they will be allowed one extra authentication attempt before getting locked out. If the attacker is aware that they received an extra attempt, then they know that one of the passwords attempted in the attack corresponds to a user's incorrect input. This gives the attacker more information about the user's password, which may be used on subsequent attacks. It might even give them one of the user's passwords to a completely different system, if the user is cycling through their other passwords while attempting to authenticate. An example of this is given in Table 5.2. The attacker receives an extra guess ("Marshall") when an incorrect user input was matched ("Mars").

TIIL Contents		Attacker Input
Mars		Marriott
Venus		Mars
		Marseilles
	Expected Lockout⟶	Marshall

User Password: "Jupiter"

Table 5.2: TIIL Attack Scenario with Lockout at Five Tries

Further, the concept of a "session" is not built into most authentication systems. Users send authentication requests to a server, which simply accepts or denies them. This is problematic for TIILs because the user may attempt to log in and give up, at which point the user's TIIL may persist for a long time. This can be avoided by deleting the lists one minute or so after they are created. Unfortunately, an attacker may be able to ignore the list lifetime by using automated tools to monitor user authentication traffic and attempt several guesses in a short span of time. As will be seen later, imposing a separate lifetime on the TIIL is not strictly required for security.

5.2.3 Client-Side Implementation

The above and following discussions focus on the server side of authentication. Alternatively, the TIIL could be implemented as a feature of the client. This might be seen as

a way to avoid a TIIL attack. Currently, Microsoft Windows clients cache domain logon information in hashed form on clients [Microsoft Corporation]. This is used when the domain server is unreachable, such as when the client is offline or the domain server is down. A TIIL could be implemented using a similar mechanism. However, a client-side implementation is not without its problems. Utilities currently exist which can retrieve the cached domain information from Windows clients and input them to common cracking programs [Pilon 2007]. A client-side TIIL would be similarly vulnerable, and a TIIL attack may still be possible with physical access to the machine.

Much of the analysis which follows is applicable to a client-side TIIL. A server-side implementation is considered here because it is universally applicable. However, it is up to authentication system designers to produce an implementation which meets the security and usability needs of their users, whether this involves a client-side or server-side TIIL.

5.2.4 TIIL Policies

The risk of a TIIL attack can be greatly reduced by implementing a simple policy. The TIIL should be cleared when:

- The account is locked out.
- The user successfully authenticates.
- Any time the number of login attempts available to the user is reset.

This restricts the size of the TIIL to the number of login attempts defined by the system. Unless the size of the TIIL is known to the attacker, they will be unable to determine when an extra authentication attempt has been granted because they will never receive more login attempts than the system limit.

These policies also make implementation easier, since current authentication systems already code for such events. For example, when an account gets locked out, an authentication system might call a function to set a “lockout” flag in the account and

update a logfile. Clearing the TIIL can be added to existing functions like this one with little trouble.

Other policies for clearing the TIIL are optional, and may be implemented for greater security. These include:

- Close of authentication session (if supported by the authentication system)
- Start of new authentication session (detected by checking against last known IP/host ID)
- Expiration of list lifetime

5.2.5 Revised Implementation

To summarize, authentication systems would maintain separate TIILs per user, and clear them on account lockout, successful authentication, and any time their lockout counter is reset. If a user input matches an entry in their TIIL, it is ignored and does not count against their lockout count.

It is important to remember that we are not maintaining incorrect input lists for use over multiple authentications. The TIILs are intended to be temporary. The results given in section 5.1 (page 77) were collected from single authentication sessions where users were locked out after five incorrect inputs. A thorough analysis of the risk of this specific implementation follows. Though it may be useful to store a persistent list of incorrect inputs over multiple authentication sessions, this is a more complicated system than the one described here and may be a greater security risk.

5.3 Risk Analysis

Suppose an authentication system has a password space of size $|P|$, and users are given T tries to authenticate before being locked out. A user has made $|u|$ distinct incorrect inputs, and is not yet locked out or authenticated. Let u denote the set of strings stored

in the TIIL. Note that $|u| \leq (T-1)$. This is because an attacker cannot login when the account is locked out, and once the user successfully authenticates the system clears the TIIL. Next, assume an attacker will make guesses until the user's account is locked out. The number of guesses, α, the attacker can make is then given by $\alpha = (T - |u|)$.

Throughout this analysis, we assume the attacker knows the size of the TIIL. A TIIL attack succeeds if the attacker receives an extra login attempt (revealing that one of their guesses matched an item in the TIIL) or matches the user's password on any of their logins.

For example, take a user whose password is "x". The user has $T = 5$ tries to authenticate before being locked out. An input of "x" will successfully authenticate the user, and anything else is considered an incorrect input. The user's guesses are "z", "c", and "s". The TIIL then contains {"z","c","s"} and $|u| = 3$. At this time, an attacker begins a TIIL attack. We assume the attacker can monitor the authentication traffic and knows the user has made 3 distinct guesses, but is otherwise unaware of the content of those guesses. The attacker gets $\alpha = (T - |u|) = 2$ guesses before the account is locked out.

TIIL Contents	Possible Attack Scenarios	
{z, c, s}	a, b	(1)
	c, d, e	(2)
	w, s, x	(3)

User Password: "x"

Table 5.3: TIIL Attack Scenarios

Three attack scenarios are shown in Table 5.3. Note that these are independent scenarios, as the TIIL is cleared when the T tries for the user are exhausted. In scenario (1), the attacker makes two random guesses which do not match the user's password or an item in the TIIL. In scenario (2), the attacker matched the item "c" in the TIIL and was granted another guess, for a total of three guesses. This is the only indication to the attacker that

an incorrect input has been matched, and the attacker will not know which of the first two guesses was the match. In scenario (3), the attacker is given an additional attempt and guesses the user's password. Both scenarios (2) and (3) are considered successful TIIL attacks. However, instead of launching a TIIL attack, the attacker could have launched a normal brute-force attack and had five guesses at the user's password. The following analysis compares the risk of a TIIL attack with that of a normal brute-force attack by comparing their probabilities of success.

5.3.1 Simplistic TIIL Attack

In order to work up to the full probability of success for a TIIL attack given in Equation 5.3 on page 87, I make several approximations. As the analysis continues, the approximations are gradually replaced by more correct expressions.

To begin with, we assume that the attacker can only guess one item from the TIIL, at best, and the probability of guessing anything else is one[3]. In other words, it is difficult to guess an item from the TIIL but easy to guess anything else. An attacker can make $(\alpha + 1)$ guesses before being locked out, if and only if one of the first α guesses was a string from the TIIL. There are $|u|$ items in the TIIL, so the probability of choosing an item is $\frac{|u|}{|P|}$. The probability of an attacker choosing an item from the TIIL within the first α guesses is then $\alpha \times \frac{|u|}{|P|}$, given our simplistic assumption that the probability of guessing anything else is 1. The worst case of this occurs when $|u| = \frac{T}{2}$ where the probability of an attacker guessing one of the user's inputs is $\frac{T^2}{4 \times |P|}$. We can see now that T is the defining factor in the risk of this system. We also see that, in the worst case, the attack occurs in the time between the $(\frac{T}{2})$th input and the $(\frac{T}{2} + 1)$th input. As will be shown later, these are both correct conclusions.

[3]These simplifications are purely for illustrative purposes and will produce incorrect or incomplete probability equations for now. Correct equations are provided in subsequent sections.

5.3.2 TIIL Attack Part 2

The next step in our analysis is to allow the attacker to guess more than one item from the TIIL and gain more than one extra attempt. Again take $|u|$ as the number of items in the TIIL, and α as the initial number of guesses available to the attacker before lockout. The full equation for the probability of an attacker guessing at least one of the user's incorrect inputs is:

$$\sum_{k=1}^{min(|u|,\alpha)} \left[\binom{\alpha + k - 1}{k} \mathbf{P}(|u|, k) \left(\frac{1}{|P|} \right)^k \right] \times \left(1 - \frac{|u|}{|P|} \right)^\alpha \tag{5.1}$$

The terms of this equation can be broken down and explained as follows:

k — The number of items from the TIIL that the attacker has guessed.

$\sum_{k=1}^{min(|u|,\alpha)}$ — The attacker can guess anywhere from one to $|u|$ of the user's incorrect inputs, within the α guesses they have available. At this point it should be clear that our worst case scenario for the simplistic case still holds for the advanced case. When $|u| = \alpha = \frac{T}{2}$, the summation is **maximized**.

$\binom{\alpha + k - 1}{k}$ — If the attacker guesses k items from the TIIL, the number of guesses they have available increases by k. The -1 term is necessary because the attacker's final guess before lockout cannot match an item from the TIIL, because then they would get an additional guess.

$\mathbf{P}(|u|, k)$ Counts the permutations of the k items from the TIIL which are guessed by the attacker.

$\left(\frac{1}{|P|}\right)^k$ The probability of guessing k specific items in the password space. Here, the specific items are the user's inputs.

$\left(1 - \frac{|u|}{|P|}\right)^\alpha$ This covers the probability for the attacker's guesses which do not match any items from the TIIL. No matter how many of the attacker's guesses are taken from the TIIL, the attacker will eventually make α guesses which cause a lockout. The combinations of this are given by the $\binom{\alpha+k-1}{k}$ term within the summation, but since this term does not depend on k it can be removed from the summation and considered independently.

5.3.3 TIIL Attack Part 3

However, Equation 5.1 is still incomplete. The additional guesses given to the attacker gives them a better chance of guessing the user's actual password. The $\left(1 - \frac{|u|}{|P|}\right)^\alpha$ term in Equation 5.1 should be replaced by:

$$\frac{1}{|P|} \sum_{m=0}^{\alpha-1} \left(1 - \frac{|u|+1}{|P|}\right)^m + \left(1 - \frac{|u|+1}{|P|}\right)^\alpha \tag{5.2}$$

The summation in Equation 5.2 sums the probability of guessing the user's password on the first available guess, missing the user's password *and* the incorrect input list on the first guess but guessing the user's password on the second guess, and so on. The $\left(1 - \frac{|u|+1}{|P|}\right)^\alpha$ term of Equation 5.2 gives the probability of missing both the user's password and the incorrect input list on all α tries.

When Equation 5.2 is substituted into Equation 5.1, the final probability of success of a TIIL attack is given below:

$$\sum_{k=1}^{min(|u|,\alpha)} \left[\binom{\alpha+k-1}{k} \mathbf{P}(|u|,k) \left(\frac{1}{|P|}\right)^k \right] \times \left[\frac{1}{|P|} \sum_{m=0}^{\alpha-1} \left(1 - \frac{|u|+1}{|P|}\right)^m + \left(1 - \frac{|u|+1}{|P|}\right)^\alpha \right] \tag{5.3}$$

This is the combined probability of an attacker either guessing items from the TIIL, or being authenticated by guessing the user's actual password.

5.3.4 TIIL Attack Approximations

Luckily, Equation 5.3 can be simplified by making two assumptions. As shown in Table 5.5, these assumptions are quite safe. The first assumption is that we only consider the worst case scenario where $|u| = \alpha = \frac{T}{2}$. This occurs when T is even, and the attack is launched in the time between the $(\frac{T}{2})$th input and the $(\frac{T}{2}+1)$th input. The second assumption is that the password space $|P|$ is large.

When $|P|$ is large, almost all the terms of Equation 5.3 become relatively insignificant. For example, because of the $\left(\frac{1}{|P|}\right)^k$ term, only the $k = 1$ term in the summation need be considered since the higher order terms are too small. This can be explained intuitively: when $|P|$ is large, the probability of choosing any two specific passwords from the password space is nearly zero. The right hand term of Equation 5.3 (given in Equation 5.2) can also be simplified to 1 for the same reason. Randomly choosing both a password from the incorrect input list and the user's actual password is just too unlikely.

After simplifying Equation 5.3 with the worst case assumption $|u| = \alpha$, we get:

$$\sum_{k=1}^{|u|} \left[\binom{|u|+k-1}{k} \mathbf{P}(|u|,k) \left(\frac{1}{|P|}\right)^k \right] \times \left[\frac{1}{|P|} \sum_{m=0}^{|u|-1} \left(1 - \frac{|u|+1}{|P|}\right)^m + \left(1 - \frac{|u|+1}{|P|}\right)^{|u|} \right] \tag{5.4}$$

After taking only the $k = 1$ term and dropping the term from the right, the equation

becomes far simpler:

$$\left[\binom{|u|}{1}\mathbf{P}(|u|,1)\left(\frac{1}{|P|}\right)\right] = \frac{|u|^2}{|P|} \tag{5.5}$$

Thus the probability of success for a TIIL attack appears to be $O(|u|^2)$. How does this compare to a normal brute-force attack?

5.3.5 Brute-Force Attack

The probability of success for a brute-force attack can be simply given by the complement of the probability of failure on all T tries:

$$\left(1 - \left(1 - \frac{1}{|P|}\right)^T\right) \tag{5.6}$$

However, for the purpose of comparison with Equation 5.5 this will need to be restated. The probability of success for a brute force attack can also be given by:

$$\frac{1}{|P|} + \left(1 - \frac{1}{|P|}\right)\frac{1}{|P|} + \cdots + \left(1 - \frac{1}{|P|}\right)^{T-1}\frac{1}{|P|} \tag{5.7}$$

This sums the probability of success on the first guess, the probability of a miss on the first guess multiplied by the probability of success on the second guess, and so on for all T guesses. Equation 5.7 can then be rewritten as:

$$\frac{1}{|P|}\left(\left(1 - \frac{1}{|P|}\right)^0 + \left(1 - \frac{1}{|P|}\right)^1 + \cdots + \left(1 - \frac{1}{|P|}\right)^{T-1}\right) = \frac{1}{|P|}\sum_{m=0}^{T-1}\left(1 - \frac{1}{|P|}\right)^m \tag{5.8}$$

Each term within the sum given in Equation 5.8 is very close to 1 given the assumption that $|P|$ is large. Since there are T terms, this probability can be approximated by $\frac{T}{|P|}$. This gives us a probability of success for a normal brute-force attack that is $O(T)$.

5.3.6 Risk

We can define the *risk* of the TIIL system as the ratio of the probabilities in Equations 5.3 and 5.6. These are the probabilities of success of a TIIL attack and a normal brute-force attack, respectively. Since $T = 2|u|$ under our worst case assumption, the risk can be approximated by:

$$\frac{\frac{|u|^2}{|P|}}{\frac{2|u|}{|P|}} = \frac{|u|}{2} = \frac{T}{4} \tag{5.9}$$

Thus, given the assumptions made earlier, the risk of using TIILs is no greater than that faced by a standard authentication system if T is **four or less**. Note that these probabilities are independent. An attacker would choose either to launch a TIIL attack at the correct time, or to have all T tries available for a normal attack before lockout. In Table 5.5, I have calculated the probabilities using the full Equations 5.3 and 5.6 in order to test our approximations. Though only a maximum of six significant figures are shown, calculations were performed in OpenOffice.org Calc, which uses 64-bit precision (approximately 15 significant digits). As you can see, the approximations provide an accurate calculation of the risk incurred by the incorrect input list system.

$\|P\|$	T	$\|u\|$	α	$k = 1^\star$	$k = 2^\star$	Eqn 5.2	Probability of success for TIIL attack Eqn 5.3	Probability of success for normal attack Eqn 5.6	Risk
2^{12}	8	4	4	3.906×10^{-3}	7.153×10^{-6}	9.961×10^{-1}	3.898×10^{-3}	1.951×10^{-3}	1.99756
2^{18}	8	4	4	6.104×10^{-5}	1.746×10^{-9}	9.999×10^{-1}	6.103×10^{-5}	3.052×10^{-5}	1.99996
2^{18}	6	3	3	3.433×10^{-5}	5.239×10^{-10}	$\approx 1.000^{\star\star}$	3.433×10^{-5}	2.289×10^{-5}	1.49999
2^{18}	3	1	2	7.629×10^{-6}	-	$\approx 1.000^{\star\star}$	7.629×10^{-6}	1.144×10^{-5}	0.66666
2^{50}	8	4	4	1.421×10^{-14}	9.466×10^{-29}	$\approx 1.000^{\star\star}$	1.421×10^{-14}	7.105×10^{-15}	2.00000
2^{50}	5	2	3	5.329×10^{-15}	9.466×10^{-30}	$\approx 1.000^{\star\star}$	5.329×10^{-15}	4.441×10^{-15}	1.20000
2^{50}	4	2	2	3.553×10^{-15}	4.733×10^{-30}	$\approx 1.000^{\star\star}$	3.553×10^{-15}	3.553×10^{-15}	1.00000

*Higher order terms were calculated but are not shown

**Value was calculated as 1.000 with available precision.

Table 5.5: Additional Risk of Attack of the Incorrect Input List System

Keep in mind that while an attacker can attempt a brute-force attack on a

user's account at any time, the resources and time necessary to perform a TIIL attack are much greater. An attack system would need to monitor authentication attempts, discern incorrect inputs, and launch an attack immediately after $\frac{T}{2}$ incorrect inputs are detected. Even with such an attack system in place, the attack is unlikely to succeed. The probability of success is not very much greater than that of a common brute-force attack, and in some cases it is lower. As shown in Table 5.5, an organization which uses a "three strikes" lockout policy could implement TIILs with no additional risk. Therefore, I feel it is safe to recommend TIILs for use in current and future authentication systems.

Chapter 6

FUTURE WORK

My suggestions for future work can be broadly categorized into the areas of memorability, further testing of the picture-password system, and security issues.

6.1 Assessing memorability

6.1.1 Multiple Passwords

A major element of the "password problem[1]" is the cognitive burden placed on users by the need to remember multiple passwords. Unfortunately, due to practical considerations in experimental design, authentication systems are rarely studied in the context of multiple passwords. Simply asking users to remember multiple passwords is troublesome, because these passwords must then be remembered in addition to the multiple passwords the user must juggle in daily life. According to a study by Sasse, Brostoff, and Weirich [Sasse et al. 2001], users have an average of 16 passwords. Asking a user to memorize another 16 passwords in order to obtain data about an authentication system's memorability would probably produce irrelevant results.

A better approach would be to replace as many of the user's passwords as possible with those of the system to be tested. An application or browser plug-in could be developed which replaces user password input. Users would input their passwords into the experiment program, and would then be assigned and trained on a corresponding graphical

[1]See section 1.1, "Introduction", page 1.

password. Whenever a known website is accessed, the program launches an authentication dialog. Users authenticate with the experimental authentication system, and the program inserts the user's original password into the web site. Such technology is already used by many browsers and password management programs. Though it is likely that only the user's website passwords could be replaced, this should still be a substantial number of passwords. A study of this type would act both as a field test of graphical authentication systems as well as assess user performance with multiple passwords.

The Distinctiveness Dilemma[2] also indicates that picture-PIN password systems, where the length of a user's password and number of pictures is relatively small, may be the best choice for remembering multiple passwords. By using distinct, small sets of pictures for each password, multiple passwords should be more easily remembered than if the same grid is used in separate contexts.

6.1.2 Password Constraints

The results of my study showed that about 7% of incorrect inputs were less than the eight-item password length. Because participants knew the password length was eight, it is possible that they were able to reconstruct their password by guessing at the missing items. If password length was variable, and a password was partially forgotten, it would be more difficult for users to reconstruct their password. Users could forget the number of items in their password and have no way of retrieving the correct amount.

Constraining passwords to a specific length may be a simple and effective way to improve password usability while maintaining password complexity. Though variable length passwords are theoretically more difficult to guess, constraining passwords to a standard length might increase average password entropy, as users will be unable to pick short passwords. A comparison of variable length passwords to specific length passwords (of similar complexity) for memorability would be required.

[2]see section 2.8, "The Distinctiveness Dilemma", page 25

In addition to constraining password length, other constraints may be considered. For example, knowing that passwords cannot contain repeated items may also aid in reconstruction of partially remembered passwords. Again, even though this reduces the theoretical complexity of passwords, it may increase entropy by prohibiting whole dictionary words and/or forcing the use of mixed-case in user-selected passwords. Unfortunately, while it is possible to constrain password length with most current authentication systems, (non-consecutive) repeated items are not commonly checked.

6.1.3 Graphical Structures

Mandler and Ritchey found that scene schema (layout) can improve memorability for items in a scene [Mandler and Ritchey 1977]. This finding could be applied to graphical password systems. Simply by arranging the pictures into a recognizable structural arrangement, memorability for items should be improved since they can be more easily remembered spatially. For example, instead of arranging pictures in a standard grid, pictures could be arranged into an abstract skyline. The items are still randomly assigned, but the position of items (and therefore the items themselves) are more memorable because they map to meaningful spatial locations. Unfortunately, producing a structural arrangement requires a great deal of blank space and may only be applicable to picture-password systems using a small number of pictures or large displays.

6.2 Further Testing of the Picture-Password System

6.2.1 Unordered Input

The results of my study show that unordered recall can produce an extremely successful authentication system. However, since users were trained serially it is necessary to see whether this result is still maintained if users are not trained with a serial requirement.

In this case, the design of the training system may have an effect on the perfor-

mance of participants. Though password items may be entered in any order, users still choose the items in some particular order. Training can then be carried out with three approaches:

1. Impose no order during training. Users are allowed to select items as freely in training as in a typical authentication.

2. Allow users to choose an order during training. Users are asked to select the items in a particular order, and are trained on that ordering for the duration of training.

3. Generate an ordering for users and force them to use that ordering for the duration of training.

The intention here is to determine if memorability improves when users learn a particular order during training, even when passwords are accepted without respect to order.

Generating an ordering may be helpful for some users. When the set of password items for a user is randomly generated, the system may be able to find an ordering that enhances memorability and present it to the user. For example, choosing an ordering of items that alternates (as much as possible) between hands when entered with the keyboard, or an ordering of pictures that runs from left to right and top to bottom, or in a rough circle. Users could then accept this ordering, choose their own, or perhaps have the system generate another ordering based on a different algorithm. There are 8! orderings for a given password, so having the system generate orderings might be beneficial to users.

This does not reduce the strength of the passwords, because the underlying items are still randomly chosen. Our unordered passwords still have a complexity of $\binom{80}{8}$. Allowing user choice in ordering has no effect on this.

It would also be useful to find out whether or not users can better identify a changed arrangement of pictures when trained on a particular ordering of items. If so, it may still be possible to use the picture grid as a security indicator even though passwords are accepted in any order.

6.2.2 Shoulder-Surfing Resistant Input

In my study, I found that participants exhibited an insecure behavior while performing the shoulder-surfing resistant input task: using the mouse to point at their password items on the screen. In this task, participants were required to find a password item in a randomly generated picture grid. Once found, they were expected to press Ctrl and find the key which occupied the same location as the picture. It is evident from their behavior that keeping track of an unfamiliar location in a 10x8 grid is too difficult for users. Even if the mouse pointer were made invisible, the difficulty of the task may force users to use their finger to keep track of password items.

One approach is to adapt the technique of the Spy-resistant Keyboard [Tan et al. 2005] and allow users to use the mouse in a shoulder-surfing resistant way. There may also be ways of framing the grid with lines, borders, or row/column identifiers that make it easier to keep track of item locations.

6.2.3 Long-Term Effects

If users use the keyboard over the long term, will they still remember the picture items of their password or will they forget them over time? The majority of participants in the picture-password group for my study used the mouse instead of the keyboard. I predicted that, in the interest of speed, users would eventually settle into keyboard entry. A longer term study, with more frequent input tasks, is needed to determine whether or not this is true. If true, determining the effect this has on password memorability is also important.

6.2.4 Results Analysis

Timing data on user input was collected throughout the course of my study. This data could be applied to various cognitive frameworks like ACT-R theory [Anderson and Matessa 1997] and confirm, or deny, various hypotheses about how users remember passwords. For example, ACT-R theory predicts that users might group an eight-item

password into chunks of size 3-3-2. In this case, there would be a short pause in input between the third and fourth, and sixth and seventh, items. Implementation of a high resolution timer would aid in this type of analysis. Such data could be easily collected in the course of other studies.

6.2.5 Consolidation Time

The effect of Stage 4 as implemented in my experiment is difficult to interpret (see section 4.6, "Password Learning Times", on page 66). It would be worth investigating whether character passwords, of the complexity used in my experiment, are as memorable when there is no break in training.

6.2.6 Confirmatory Experiments

The small sample size used in my study often made statistical interpretation difficult. More participants would improve the robustness of my findings and allow for a better comparison of picture and character passwords.

6.3 Security Issues

6.3.1 Entropy

The true entropy of user passwords has never been tested. Unless users can be convinced to reveal their passwords for research purposes, this is unlikely to change with a research study. A better way to determine entropy would be to obtain and examine very old password files. Computing power today is sufficient to brute-force password files which have been encrypted with older protocols. Assuming that backups of very old password files can be obtained, examining the passwords would provide very useful information about password entropy. Yan [Yan et al. 2004] and others have performed such studies by examining current password files, but only the weak passwords could be brute-forced.

Though it is good to have information on the composition of passwords in the worst case, this does not provide any information above the average strength of passwords.

6.3.2 Shoulder Surfing

Tari et al. [Tari et al. 2006] assert that users enter randomly-generated passwords 40% slower than weaker, dictionary-variant passwords. They base this number on research done by Thomas et al. into typing performance metrics when typing randomly generated strings [Thomas et al. 2005]. However, this ignores the facts that users gain a great deal of practice in typing their passwords over time, and that authentication is only an intermediate goal towards more important primary tasks. In my study, three character-password participants entered their password correctly on Day 9. These participants were able to do so in only 5.3 seconds on their second entry, and their typing speeds gradually improved (see Figure 4.8 on page 73). Future studies should measure the typing speed of typical, yet practiced users on strong passwords.

There is also an unconfirmed belief that a trained individual can steal a user's password much more easily than an untrained individual. If such individuals do not make themselves publicly available for use in experiments, it should still be possible to train an experimenter in shoulder-surfing techniques. Such a study would expose optimal shoulder-surfing techniques and measure the amount of training necessary to become proficient. Since current shoulder-surfing resistant authentication methods are too cumbersome for use on a regular basis, a better study of shoulder-surfing vulnerability is needed.

6.3.3 Security Indicators

As discussed in section 2.1 on page 7, current studies reveal that users lack the education required to properly judge security indicators, and that current efforts by websites and browsers are ineffective at focusing user attention in a meaningful way. Studies should be performed to assess the amount of education required to allow users to properly judge

security indicators, and to determine the extent of insecure behaviors that users will engage in. Schechter et al. found that users will enter their password into an apparently insecure site [Schechter et al. 2007]. However, are users willing to enter more personal information such as "Mother's Maiden Name" and "City of Birth"?

Methodology for such experiments seems to be especially tricky, as users become overly sensitive to possible security problems [Anandpara et al. 2007]. This problem was identified in my study[3]. Methods for conducting studies of this type likely exist in other fields, and could be adapted to the current problem.

Research on user education is also important. Whether or not security indicators can be made useful to the general population, developing an effective education program about computer security might be useful in and of itself.

Ordered Input

In my picture-password system, the arrangement of pictures acts as a security indicator. This probably requires that input be ordered. If the authentication system allows users to choose their password items in any order, it is less likely that a change in the arrangement of pictures will be noticed. However, if users determine that keyboard input is faster, and always enter the same string of characters to authenticate, their authentication will fail on a changed grid. This occurred with both of the Group II participants in my study who had heavy keyboard usage and were able to enter their password correctly on their home grid. In this case, it may necessary to better focus user attention on pictures during training.

Picture passwords are a relatively new area of study, so the possibilities for future work are extensive and diverse. Based on the results presented in this thesis, the most

[3]See section 4.9, "Effectiveness of Picture Arrangement as a Security Indicator", page 72.

promising future work is in the area of unordered passwords. Research into security indicators and insecure user behaviors is also extremely important, as is the study of multiple user passwords.

Bibliography

Anandpara, V., Dingman, A., Jakobsson, M., Liu, D., and Roinestad, H. 2007. Phishing IQ Tests Measure Fear, Not Ability. *Usable Security (USEC'07). http://usablesecurity.org/papers/anandpara.pdf*.

Anderson, J. and Matessa, M. 1997. A production system theory of serial memory. *Psychological Review 104,* 4, 728–748.

Anderson, R. 2007a. Closing the phishing hole: fraud, risk and nonbanks. *Federal Reserve Bank of Kansas City, Conference on Nonbanks in the Payments System.*

Anderson, R. 2007b. Searching for evil. http://video.google.com/videoplay?docid=-1380463341028815296 (accessed October 2007).

Bailey, A. 2006. Analyzing 20,000 myspace passwords. http://www.cyber-knowledge.net/blog/2006/09/16/ analyzing-20000-myspace-passwords/ (accessed October 2007).

Brostoff, S. and Sasse, M. 2000. Are Passfaces more usable than passwords? A field trial investigation. *People and Computers XIV-Usability or Else! <http://www.cs.ucl.ac.uk/staff/a.sasse/hci2000.pdf>*, 405–424.

Burr, W. E., Dodson, D. F., and Polk, W. T. 2004. Special publication 800-63 - Electronic Authentication Guideline. Tech. rep., National Institute of Standards and Technology.

De Angeli, A., Coventry, L., Johnson, G., and Renaud, K. 2005. Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies 63,* 1-2, 128–152.

Deregowski, J. and Jahoda, G. 1975. Efficacy of Objects, Pictures and Words in a Simple Learning Task. *International Journal of Psychology 10,* 1, 19–25.

Deregowski, J., Parker, D., and McGeorge, P. 1999. What's in a Name, What's in a Place? The Role of Verbal Labels in Distinct Cognitive Tasks. *Current Psychology 18,* 1, 32–46.

Dhamija, R. and Perrig, A. 2000. Deja Vu: A User Study Using Images for Authentication. In *Proceedings of the 9th USENIX Security Symposium. <http://www.simson.net/ref/2000/usingImagesForAuthentication.pdf>*. 45–48.

Dhamija, R., Tygar, J. D., and Hearst, M. 2006. Why phishing works. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 581–590.

Faul, F., Erdfelder, E., Lang, A.-G., and Buchner, A. 2007. G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods 39*, 175–191.

Fleiss, J., Levin, B., and Paik, M. 2003. *Statistical methods for rates and proportions.* Wiley Series in Probability and Statistics.

Granger, S. 2001. Social Engineering Fundamentals, Part I: Hacker Tactics. Tech. rep. http://www.securityfocus.com/infocus/1527 (accessed October 2007).

Jackson, C., Simon, D., Tan, D., and Barth, A. 2007. An Evaluation of Extended Validation and Picture-in-Picture Phishing Attacks. *Proceedings of the Workshop on Usable Security (USEC'07)..*

Johnson, G. 1991. A distinctiveness model of serial learning. *Psychological Review 98,* 2, 204–217.

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

Killeen, P. R. 2005. An alternative to null-hypothesis significance tests. *Psychological science : a journal of the American Psychological Society / APS 16*, 345–53. PMID: 15869691.

Kinjo, H. and Snodgrass, J. 2000. Is there a picture superiority effect in perceptual implicit tasks? *European Journal of Cognitive Psychology 12,* 2, 145–164.

Malone, D. and Sullivan, W. 2004. Guesswork and Entropy. *IEEE Transactions on Information Theory 50,* 3, 525–526.

Man, S., Hong, D., and Mathews, M. 2003. A shoulder-surfing resistant graphical password scheme. *Proceedings of International conference on security and management I*, 101–111.

Mandler, J. and Ritchey, G. 1977. Long-term memory for pictures. *Journal of Experimental Psychology: Human Learning and Memory 3,* 4, 386–396.

Massey, J. 1994. Guessing and entropy. In *Proceedings of the IEEE International Symposium on Information Theory.*

Microsoft Corporation. 2007. Cached domain logon information. http://support.microsoft.com/kb/172931 (accessed October 2007).

Microsoft TechNet. 2007. Windows Server 2003 - Account Passwords and Policies. http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/bpactlck.mspx (accessed October 2007).

Milliken, G. and Johnson, D. 2002. *Analysis of Messy Data: Volume 1 - Designed Experiments.* Lifetime Learning Publications.

Moncur, W. and Leplâtre, G. 2007. Pictures at the ATM: exploring the usability of multiple graphical passwords. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM Press New York, NY, USA, 887–894.

Morris, R. and Thompson, K. 1979. Password security: A case history. *Communications of the ACM 22,* 11 (Nov.), 594–597.

Nelson, D. 1977. Learning to Order Pictures and Words: A Model of Sensory and Semantic Encoding. *Journal of Experimental Psychology: Human Learning and Memory 3,* 5, 485–497.

Nicholls, M. E. R., Orr, C. A., Okubo, M., and Loftus, A. 2006. Satisfaction guaranteed: The effect of spatial biases on responses to likert scales. *Psychological Science 17*, 1027–1028.

Nickerson, R. 1965. Short-term memory for complex meaningful visual configurations: A demonstration of capacity. *Canadian Journal of Psychology 19*, 155–160.

Paivio, A. and Csapo, K. 1971. Short-term sequential memory for pictures and words. *Psychonomic Science 24,* 2, 50–51.

Pilon, A. 2007. Securiteam - cachedump - recovering windows password cache entries. http://www.securiteam.com/tools/5JP0I2KFPA.html (accessed October 2007).

Preneel, B. 1993. Analysis and design of cryptographic hash functions. Ph.D. thesis, Katholieke Universiteit te Leuven.

Reinhold, A. 2007. The diceware passphrase home page. http://world.std.com/ reinhold/diceware.html (accessed Oct. 2007).

Rossion, B. and Pourtois, G. 2004a. Revisiting Snodgrass and Vanderwart's object pictorial set: The role of surface detail in basic-level object recognition. *Perception 33,* 2, 217–236.

Rossion, B. and Pourtois, G. 2004b. Snodgrass and Vanderwart Like Objects. http://alpha.cog.brown.edu:8200/stimuli/objects/svlo.zip/view (accessed Sept. 2007).

Sasse, M., Brostoff, S., and Weirich, D. 2001. Transforming the 'Weakest Link'-a Human/Computer Interaction Approach to Usable and Effective Security. *BT Technology Journal 19,* 3, 122–131.

Schechter, S., Dhamija, R., Ozment, A., and Fischer, I. 2007. The emperor's new security indicators-an evaluation of website authentication and the effect of role playing on usability studies. *Proceedings of the IEEE Symposium on Security and Privacy (S&P'07).*

Shannon, C. E. 1948. A mathematical theory of communication. *Bell Systems Technical Journal 27*, 379–423.

Snodgrass, J. and McCullough, B. 1986. The role of visual similarity in picture categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition 12,* 1, 147–154.

Snodgrass, J. and Vanderwart, M. 1980. A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity. *Journal of Experimental Psychology: Learning, Memory, and Cognition 6,* 2, 174–215.

Standing, L. 1973. Learning 10000 pictures. *The Quarterly Journal of Experimental Psychology 25,* 2, 207–222.

Stenberg, G., Radeborg, K., and Hedman, L. 1995. The picture superiority effect in a cross-modality recognition task. *Memory and Cognition 23,* 4, 425–441.

Stroop, J. R. 1935. Studies of interference in serial verbal reactions. Ph.D. thesis, George Peabody College for Teachers.

Tan, D., Keyani, P., and Czerwinski, M. 2005. Spy-resistant keyboard: more secure password entry on public touch screen displays. *Proceedings of the 19th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction*, 1–10.

Tari, F., Ozok, A., and Holden, S. 2006. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of the second symposium on Usable privacy and security*. ACM Press New York, NY, USA, 56–66.

Thomas, R., Karahasanovic, A., and Kennedy, G. 2005. An investigation into keystroke latency metrics as an indicator of programming performance. *Proceedings of the 7th Australian conference on Computing education 42*, 127–134.

Weinshall, D. and Kirkpatrick, S. 2004. Passwords you'll never forget, but can't recall. *Conference on Human Factors in Computing Systems*, 1399–1402.

Weldon, M. and Roediger, H. I. 1987. Altering retrieval demands reverses the picture superiority effect. *Memory & Cognition 15,* 4, 269–280.

Wickelgren, W. 1965. Short-term memory for repeated and non-repeated items. *The Quarterly Journal of Experimental Psychology 17*, 14–25.

Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., and Memon, N. 2005. Authentication using graphical passwords: Basic results. In *Human-Computer Interaction International 2005*.

Wiedenbeck, S., Waters, J., Sobrado, L., and Birget, J. 2006. Design and evaluation of a shoulder-surfing resistant graphical password scheme. *Proceedings of the working conference on Advanced visual interfaces*, 177–184.

Wixted, J. 2004. The psychology and neuroscience of forgetting. *Annual Review of Psychology 55*, 235–269.

Yan, J., Blackwell, A., Anderson, R., and Grant, A. 2004. Password memorability and security: empirical results. *IEEE Security & Privacy Magazine 2,* 5, 25–31.

Appendix A

Picture-Password System Evaluation Survey

Evaluation of the Picture-based password system

Please complete the following statements:

1. I have _____ used picture-based password systems before participating in this study.

1	2	3
never	rarely	often

2. I was _____ when using the mouse to enter my password.

1	2	3	4	5
very unsatisfied	unsatisfied	neither satisfied nor unsatisfied	satisfied	very satisfied

3. I was _____ when using the keyboard to enter my password.

1	2	3	4	5
very unsatisfied	unsatisfied	neither satisfied nor unsatisfied	satisfied	very satisfied

4. I found the picture password system to be _____ than a character-based password system.

1	2	3	4	5
a lot more satisfying	more satisfying	neither more nor less satisfying	less satisfying	a lot less satisfying

5. I found the picture password system to be _____ than a character-based password system.

1	2	3	4	5
a lot more efficient	more efficient	neither more nor less efficient	less efficient	a lot less efficient

When you were first given your password, the arrangement of pictures on the screen was always the same. Earlier today, you were shown a different arrangement of pictures.

6. On my original arrangement of pictures, I found it _____ to enter my password.

1	2	3	4	5
very easy	easy	neither easy nor difficult	difficult	very difficult

7. When my arrangement of pictures changed, I found it _____ to enter my password.

1	2	3	4	5
very easy	easy	neither easy nor difficult	difficult	very difficult

8. On the original arrangement of pictures, the amount of time I spent entering my password was _____.

1	2	3	4	5
very short	short	not too short, not too long	too long	far too long

9. When my arrangement of pictures changed, the amount of time I spent entering my password was _____.

1	2	3	4	5
very short	short	not too short, not too long	too long	far too long

In your final task today, a grid of letters changed as you entered your password.

10. I found it _____ to enter my password for this task.

1	2	3	4	5
very easy	easy	neither easy nor difficult	difficult	very difficult

11. The amount of time I spent entering my password was _____ for this task.

1	2	3	4	5
very short	short	not too short, not too long	too long	far too long

12. Compared to a standard character-based password system, I felt _____ while entering my password for this task.

1	2	3	4	5
a lot less secure	less secure	equally secure	more secure	a lot more secure

13. Additional observations or comments:

Appendix B

Picture Set Data

Picture set data was taken from:

- Snodgrass and Vanderwart's 1980 paper [Snodgrass and Vanderwart 1980]
- Stenberg, Radeborg, and Hedman's 1995 paper [Stenberg et al. 1995]
- Rossion and Pourtois' 2004 paper [Rossion and Pourtois 2004a]

An explanation of the design of the picture set is given in section 3.8, "Picture Set", page 49.

Table B.1: Agreement Scores for Pictures in the Snodgrass and Vanderwart Data Set

English name	Rossion and Pourtois Agreement (%)	Snodgrass and Vanderwart Agreement (%)	Assigned Stenberg Agreement Score	Cumulative Agreement Score	Inclusion in Final Picture Set
Accordion	100	88	70	258	
Airplane	100	60	90	250	
Alligator	95	88	90	273	Included
Anchor	100	93	90	283	Included

Ant	100	81	90	271	
Apple	98	98	90	286	Included
Arm	100	90	70	260	
Arrow	37	98	70	205	
Artichoke	71	52	70	193	
Ashtray	58	100	70	228	
Asparagus	100	69	70	239	
Axe	100	90	90	280	Included
Baby carriage	100	52	70	222	
Ball	89	93	90	272	
Balloon	84	100	90	274	Included
Banana	100	100	90	290	Included
Barn	100	69	70	239	
Barrel	100	100	70	270	Included
Baseball bat	100	52	70	222	
Basket	100	90	90	280	Included
Bear	100	88	70	258	
Bed	100	100	90	290	Included
Bee	100	60	70	230	
Beetle	100	50	70	220	
Bell	100	100	70	270	Included
Belt	74	98	70	242	
Bicycle	95	88	90	273	Included
Bird	100	88	70	258	
Blouse	85	43	70	198	
Book	100	100	90	290	Included
Boot	100	88	70	258	
Bottle	100	95	70	265	Included
Bow	100	74	70	244	
Bowl	100	95	70	265	
Box	94	88	70	252	

Bread	88	83	70	241	
Broom	100	100	70	270	
Brush	73	83	90	246	
Bus	100	100	90	290	Included
Butterfly	100	100	90	290	
Button	94	98	90	282	Included
Cake	100	83	90	273	Included
Camel	100	95	70	265	Included
Candle	33	100	70	203	
Cannon	95	90	90	275	Included
Cap	100	86	90	276	
Car	85	81	70	236	
Carrot	100	100	90	290	Included
Cat	100	100	90	290	Included
Caterpillar	90	79	70	239	
Celery	91	76	70	237	
Chain	95	98	90	283	
Chair	100	100	90	290	Included
Cherry	100	83	70	253	
Chicken	100	67	70	237	
Chisel	100	33	70	203	
Church	74	93	90	257	
Cigar	100	100	70	270	
Cigarette	89	98	70	257	
Clock	100	98	70	268	Included
Clothespin	100	81	70	251	
Cloud	100	95	70	265	
Clown	88	95	70	253	
Coat	95	79	70	244	
Comb	60	93	90	243	
Corn	89	81	70	240	

Couch	63	67	90	220	
Cow	100	93	90	283	
Crown	60	100	90	250	
Cup	85	93	90	268	
Deer	100	76	70	246	
Desk	100	95	70	265	Included
Dog	100	100	70	270	Included
Doll	100	71	70	241	
Donkey	77	86	90	253	
Door	53	98	70	221	
Doorknob	100	90	70	260	
Dress	38	100	70	208	
Dresser	80	36	90	206	
Drum	100	98	90	288	Included
Duck	100	95	70	265	Included
Eagle	89	76	70	235	
Ear	100	95	90	285	Included
Elephant	100	100	90	290	Included
Envelope	59	98	70	227	
Eye	100	98	90	288	Included
Fence	100	74	90	264	
Finger	100	71	70	241	
Fish	95	100	70	265	Included
Flag	84	95	70	249	
Flower	100	93	90	283	Included
Flute	74	88	70	232	
Fly	100	76	70	246	
Foot	95	95	90	280	Included
Football	89	100	70	259	
Football helmet	100	62	70	232	
Fork	100	100	90	290	Included

Fox	76	74	90	240	
French horn	100	57	70	227	
Frog	100	100	70	270	Included
Frying pan	100	60	70	230	
Garbage can	100	88	70	258	
Giraffe	100	95	90	285	Included
Glass	100	98	90	288	Included
Glasses	95	64	90	249	
Glove	94	98	70	262	
Goat	100	86	70	256	
Gorilla	59	76	70	205	
Grapes	100	90	90	280	Included
Grasshopper	90	71	70	231	
Guitar	90	98	90	278	Included
Gun	100	74	70	244	
Hair	100	90	70	260	
Hammer	100	100	90	290	Included
Hand	76	93	90	259	
Hanger	95	86	70	251	
Harp	100	93	70	263	
Hat	42	98	90	230	
Heart	50	100	70	220	
Helicopter	61	95	90	246	
Horse	89	100	90	279	
House	100	95	70	265	Included
Iron	83	95	90	268	
Ironing board	40	83	70	193	
Jacket	100	81	70	251	
Kangaroo	100	100	90	290	Included
Kettle	95	40	70	205	
Key	100	100	90	290	Included

Kite	67	100	90	257	
Knife	75	90	90	255	
Ladder	100	98	90	288	Included
Lamp	100	93	70	263	
Leaf	100	90	70	260	
Leg	100	81	70	251	
Lemon	94	100	90	284	Included
Leopard	100	76	70	246	
Lettuce	100	74	70	244	
Light bulb	100	86	70	256	
Light switch	100	67	70	237	
Lion	100	93	90	283	Included
Lips	100	93	70	263	
Lobster	100	90	70	260	
Lock	95	88	70	253	
Mitten	95	76	70	241	
Monkey	100	95	90	285	Included
Moon	85	62	70	217	
Motorcycle	100	95	90	285	Included
Mountain	44	90	70	204	
Mouse	100	79	70	249	
Mushroom	100	98	90	288	Included
Nail	68	98	70	236	
Nail file	100	67	70	237	
Necklace	94	60	90	244	
Needle	60	81	90	231	
Nose	80	98	90	268	Included
Nut	100	64	90	254	
Onion	95	95	90	280	
Orange	100	81	90	271	Included
Ostrich	85	86	70	241	

Owl	100	100	90	290	Included
Paintbrush	100	74	90	264	
Pants	100	88	90	278	Included
Peach	100	74	70	244	
Peacock	100	79	70	249	
Peanut	100	93	70	263	
Pear	89	100	90	279	Included
Pen	100	95	70	265	
Pencil	95	100	70	265	
Penguin	100	90	90	280	Included
Pepper	75	67	70	212	
Piano	100	81	90	271	Included
Pig	74	90	90	254	
Pineapple	100	100	90	290	Included
Pipe	53	98	90	241	
Pitcher	100	88	70	258	
Pliers	100	88	70	258	
Plug	100	88	70	258	
Pocketbook	100	57	70	227	
Pot	50	81	70	201	
Potato	100	90	70	260	
Pumpkin	94	98	70	262	
Rabbit	100	100	90	290	Included
Raccoon	85	79	70	234	
Record player	84	50	70	204	
Refrigerator	89	93	70	252	
Rhinoceros	39	83	70	192	
Ring	100	98	90	288	Included
Rocking chair	69	90	90	249	
Roller skate	100	52	90	242	
Rolling pin	100	71	70	241	

Rooster	65	76	90	231	
Ruler	80	98	90	268	
Sailboat	80	93	70	243	
Saltshaker	100	83	70	253	
Sandwich	100	100	70	270	Included
Saw	64	98	90	252	
Scissors	100	98	90	288	Included
Screw	100	93	90	283	Included
Screwdriver	100	98	90	288	
Sea horse	100	88	70	258	
Seal	60	88	90	238	
Sheep	59	67	70	196	
Shirt	100	100	90	290	Included
Shoe	100	95	90	285	Included
Skirt	100	98	90	288	
Skunk	100	98	90	288	Included
Sled	100	98	70	268	
Snail	100	86	90	276	
Snake	100	98	90	288	Included
Snowman	100	100	90	290	Included
Sock	89	100	70	259	
Spider	85	88	90	263	
Spinning wheel	100	50	70	220	
Spool of thread	69	55	70	194	
Spoon	100	98	70	268	
Squirrel	79	93	90	262	
Star	47	100	90	237	
Stool	100	98	90	288	Included
Stove	95	76	70	241	
Strawberry	100	90	70	260	
Suitcase	85	79	90	254	

Sun	79	100	70	249	
Swan	100	88	90	278	Included
Sweater	100	83	70	253	
Swing	100	95	70	265	Included
Table	100	95	90	285	Included
Telephone	83	86	90	259	
Television	95	52	90	237	
Tennis racket	100	86	70	256	
Thimble	100	83	90	273	Included
Thumb	95	98	70	263	
Tie	100	69	90	259	
Tiger	100	93	90	283	
Toaster	100	100	90	290	Included
Toe	95	55	70	220	
Tomato	100	88	70	258	
Toothbrush	94	98	90	282	
Top	80	86	70	236	
Traffic light	100	67	70	237	
Train	100	86	70	256	
Tree	100	100	90	290	Included
Truck	100	90	90	280	Included
Trumpet	89	79	90	258	
Turtle	90	95	90	275	Included
Umbrella	100	100	90	290	Included
Vase	0	95	70	165	
Vest	100	98	90	288	Included
Violin	95	86	70	251	
Wagon	85	79	90	254	
Watch	10	90	90	190	
Watering can	100	55	90	245	
Watermelon	100	86	70	256	

Well	94	90	70	254	
Wheel	94	93	70	257	
Whistle	100	100	90	290	Included
Windmill	100	98	70	268	Included
Window	68	95	70	233	
Wineglass	100	50	70	220	
Wrench	100	76	70	246	
Zebra	95	98	90	283	Included