Authentication Using Graphical password: Effects of Increased Security on Usability

William M. Martin

March 20, 2018

Abstract

Graphical User Authentication (GUA) is a method that uses images as passwords, instead of alphanumeric characters. We propose PassDecoy, a shoulder surfing resilient GUA system, and through previous research has been shown to have a higher degree of security when compared to PassPoints (Chaisson et al. [3]). However, previous work has exposed a password problem, where the effect of increased security often comes at the expense of decreased usability. We conducted a user study in an effort to evaluate the usability of PassDecoy, and its ability to be both secure and usable. The results of the user test yielded mixed results, where many results showed insufficient evidence to demonstrate an effect on usability. However, there was some observed decrease in usability, specifically in login time and a users perception of their ability to memorize passwords. These are key measures of password usability, and futrue work is required for PassDecoy to be implemented futher.

Keywords: Graphical User Authenticatiom, Shoulder Surfing Attacks, Hybrid Imagery, Usable Security

Contents

List of Figures

1 Introduction

Human Computer Interface Security (HCIsec) is the study of the interaction between humans and computers, specifically its pertinence towards information security. Its purpose is to improve the usability of security features, such as password authentication in end user systems. One of the primary usability characteristics evaluated in HCIsec is the efficiency of password authentication. However, increased efficiency and a higher degree of usability is often found to be in tension with a system's ability to prevent attacks. This tension has been described as the password problem [10], suggesting that a secure password is inherently less usable. The problem exists because passwords are expected to comply with two conflicting requirements:

1. Passwords are easy to remember, and the authentication protocol is completed quickly and easily by the human user.
2. Passwords are secure, i.e. the password looks random, is hard to guess, is changed frequently, and is different on different accounts of the same user.

Fulfillment of both these requirements is difficult for users, and this problem is well known in the security community. Previous research has shown that users tend to choose and handle alphanumeric passwords very insecurely [6]. Alphanumeric passwords have been used extensively as the prominent method of authentication for decades. This authentication scheme uses a string of numbers, symbols and letters, which is resistant to many forms of attack if the user maintains a secure password. However, a secure alphanumeric password is hard to memorize and successfully recall. This has generally resulted in users choosing passwords that are short or from a dictionary, rather than a random alphanumeric string. Chester Wisniewski, a security researcher within a large company, implemented an attack on his company's network to show the importance of keeping a strong password. He was able to retrieve about 60 percent of the employees' passwords within 30 seconds. His team used a brute force attack, attempting 3 million password keys per second (kps) [6]. It is important to recognize that malicious systems constantly improve. Currently, these same types of attacks have been shown to attempt over 13 million kps, and this number could grow to more than 17 million kps by 2020 [6]. This example shows how systems with malicious intent improve, and in response so must information security. Nonetheless alphanumeric passwords continue to be the main form of authentication, despite its insecurity due to the difficulty of maintaining a strong password.

Graphical User Authentication (GUA) schemes have been developed to address the shortcomings of other authentication methods. The idea of Graphical User Authentication, originally described by Greg

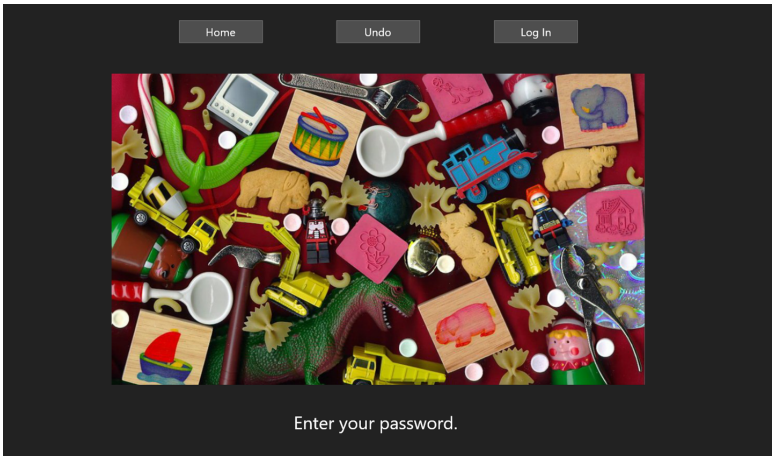

Figure 1: An example of a Graphical Password.

Blonder in 1996 [1], is to have the user select a few chosen regions from an image as their password input. A user would select an image to be used in their authentication protocol, such as the image in Figure 1. The user would then register their password, by selecting a number of click points. For example, a user may select one of the white candies, cement truck, and wrench in specific order. This image will then be presented again in a graphical user interface, where the user will recall these points in specific order. This results in an alternative to alphanumeric passwords, where a user is tasked with remembering parts of images, rather than characters.

The password problem arises primarily from fundamental limitations of human long term memory [6]. Once a password has been chosen and learned the user must recall their password to login. However, people regularly forget their passwords, because alphanumeric schemes achieve security by requiring the user to memorize a convoluted string of characters. The motivation for GUA initially comes from the fact that people have greater ability to memorize images within long term memory. As a result, users can use a more complex password without forgetting it over time. While memory remains the initial motivation for GUA, additional advantages have given GUA the merit to be implemented and studied further.

Research has shown that GUA is a feasible substitute for alphanumeric passwords due to many usability advantages [6]. One of these advantages is a GUA system's suitability for mobile devices, which are becoming a significant portion of technology use today. A graphical password can be presented across the entire screen, while alphanumeric passwords require a keyboard where the user must switch between letters, symbols and numbers. The graphical approach towards user authentication also has efficient login and registration times, which are key measures of usability. The elapsed time for a user to create an account, and authenticate themselves while using the account must be minimized for the system to be efficient. In comparison to alphanumeric authentication, GUA has acceptable error rates, meaning GUA is easy to use.

Additionally GUA has a strong public perception, showing that the users are satisfied while completing tasks within a GUA system. Lastly, the reuse of one password across many accounts and the use of pre-configured (default) passwords are significantly reduced when using GUA [4]. Usability science has shown that graphical passwords can be a superior authentication method, so why haven't they been implemented further? This is because the password problem has two parts, usability and security. For GUA to be applied in the real world it must possess the same degree of security when compared to other password methods. Therefore, the question arises, is GUA as secure as other methods? And can a GUA system achieve a higher degree of security without lowering usability?

2 Background and Related Work

Alphanumeric passwords are the most popular user authentication method, despite security and usability problems. Alternate methods such as biometric authentication also has drawbacks [2]. Graphical User Authentication offers another alternative, which has been shown to be more easily used and secure in comparison to previously described methods [6].

2.1 Security Attacks on GUA

Previous research shows that a graphical password is more secure than an alphanumeric password [14] [3]. This is due to a few factors, one being graphical passwords strong resilience towards most computer based security attacks. One example is a brute force attack, which attempts to guess passwords using enormous processing power and an unlimited number of attempts. Resilience towards this attack is correlated with a system's theoretical password space, which is all possible password combinations. If the theoretical password space is increased to a certain scale, brute force attacks can be resisted. On the other hand, if the theoretical password space is insufficient, brute force attacks can cause serious harm to a system. In alphanumeric passwords, the alphabet size is typically 64 to 96 characters, dependent on the system's requirements. The most secure alphanumeric schemes, require a symbol, a number, and a minimum length of 8 characters. This results in a theoretical password space where the lower-bound is $96^8 = 7.2x10^{15}$.

The equivalent to an alphabet size in GUA is the number of click-able points on the image. This number is calculated by taking the dimensions of the picture, divided by the tolerance (point size) of the GUA system. A typical GUA system will use a 1024 x 752 (pixel) image with the tolerance set to 20 x 20 (pixels). This results in 1,925 possible inputs, which is far greater than any alphanumeric password in use today [13]. If a user must select 5 click points to form a password, then the theoretical password space size is at least

$1925^5 = 2.6x10^{16}$. This is a size greater than the most secure alphanumeric systems, and it can be increased through system requirements. A brute force attack on this system must obtain the exact image of the user, so these attacks are much more difficult circulate. This is because the medium used in the authentication protocol is different for every user, while alphanumeric uses a standard text box across all users. Generally speaking, it is more difficult to use a brute force attack against graphical passwords than alphanumeric, because these attacks require substantial time and computing power.

A well-designed GUA scheme includes a large image library including all of the picture passwords a user may select. For a person to attack these sytems through the use of a dictionary, they must first obtain the users image, and create a dictionary for that image. This individualized approach to authentication has prevented these computer based dictionary attacks. A dictionary attack involves guessing passwords from an exhaustive list called a dictionary. This list consists of all passwords with a higher possibility of being remembered easily, sorted from most to least probable. While a brute force attack systematically attempts every element of the theoretical password space, a dictionary attack attempts the passwords most likely to succeed. A large theoretical password space does not guarantee security, but it makes the implementation of these attacks come at much higher cost. Additionally, a GUA scheme can be designed to reduce the predictability of password inputs. In comparison, it is very difficult to prevent a user from selecting alphanumeric characters that are already defined in a dictionary.

Another security attack is phishing, where a counterfeit interface is deployed in an effort to get users to willingly disclose their password. This attack has been shown to be most detrimental because they exploit the user, commonly known as the weakest link in a computer system [7]. Phishing is often carried out via e-mail and social network spoofing, and guides users to enter their password on a fake website which looks and feels like the legitimate one. A phishing attack's ability to steal passwords depends on its ability to mimic the legitimate interface. This relationship means phishing is less successful against GUA, because it is difficult to implement a counterfeit GUA system. This results from alphanumeric passwords being collected using one simple text box, while graphical passwords are collected using any image within a very large library. The counterfeit system would need to display the exact image assigned to a user for their social engineering scam to work.

A newly developed type of security attack is spy-ware. Unfortunately, GUA cannot protect against all spy-ware attacks, and users must continue to rely on protective anti-virus software. However, the design of GUA does rule out a few instances of spy-ware. One of the most commonly used pieces of software by hackers is a key-logger, which records the user's keyboard input [6]. GUA does not require the use of a keyboard, whereas systems that use fixed passwords and are entered via the keyboard are easily cracked by a key-logger attack.

The described robustness against most computer based security attacks has placed graphical passwords a step above alphanumeric passwords when looking at security. However there remains a significant breach in graphical password security in the form of shoulder surfing attacks. This attack model has proven to be a significant invalidation of Graphical User Authentication security, and in turn has impeded the application of GUA in real world systems [10].

2.1.1 Shoulder Surfing Attacks

Numerous studies have shown that shoulder surfing attacks (SSAs) on graphical passwords are a reality [6] [7] [13]. This attack method refers to someone using direct observation techniques to capture password inputs. For example, someone watches over a user's shoulder or records using an external device as the user enters a password. SSAs are effectively used in public places and can be most detrimental because these attacks obtain private information without the user's knowledge, where other attacks such as Phishing require the user to willingly divulge information. GUA is heavily reliant on a visual interface, which is the primary reason shoulder surfing has remained a severe security threat.

To date, many graphical password schemes have been evaluated and proposed in an effort to fix the SSA problem. However, many of these solutions have significant usability drawbacks, generally observed in the time and effort to login [14] [3]. There are additional studies that propose new approaches, claiming to have solved the SSA problem without ever evaluating usability measures [8]. In these situations, researchers are ignoring half of the password problem without usability metrics to augment a SSA resilient scheme. This half of the problem is important because the absence of a usability study, or an observed reduction in usability makes these schemes less suitable for real world application.

2.2 Categorization of Graphical User Authentication

More than twenty years have passed since the first Graphical User Authentication model was proposed. In this time, numerous studies have been undertaken and a wide variety of authentication schemes have resulted. According to authentication style, proposed GUA models can be broadly classified into three categories: Draw-metric, Cogno-metric and Loci-metric [6]. Each of these schemes will be described below, as some, by design are more suitable to solve the password and shoulder surfing attack problems. One of these schemes will be selected for its suitability and all future references to GUA will refer to this scheme.

2.2.1 Draw-Metric Schemes

Draw-metric GUA schemes are also known as recall-based graphical password [6]. In draw-metric

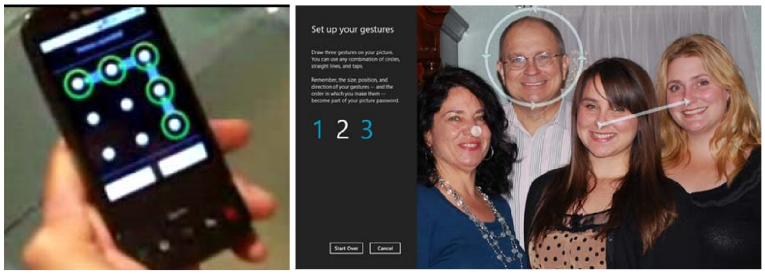

Figure 2: Draw-metric examples: Android and Windows 8 interfaces.

schemes a user reproduces a drawing that the user created during the registration phase. The drawing consists of one continuous stroke, or several strokes separated by pen-up actions.

This scheme has largely been ignored when attempting to solve the SSA problem [6]. This is because it is heavily reliant on the visual interface for confirmation, where the drawn password will always appear on the screen. Shoulder surfing attacks have been successful on draw-metric schemes, even on systems that do not display the password input. In these systems, a physical drawing action is still required and can be directly observed. Also, draw-metric schemes significantly benefit from the use of a touchscreen or stylus, which are not always available and make real world application difficult. Lastly, these scheme has been shown to have the smallest effective password space when compared to other schemes [6]. This is because users will often include symmetry and a small number of drawn strokes. While this scheme has been shown to be easy to remember and simple to operate, it has generally only been applied to systems which require a low security level. By design, draw-metric schemes are not conducive to a SSA resilient graphical password.

2.2.2 Cogno-Metric Schemes

Cogno-metric GUA schemes are also known as recognition-based graphical password [6]. They involve identifying whether a user has seen an image before. In this scheme, a user will create a password by selecting several images from an image library. These selections become the user's password. During authentication the user must select their entire set of images from a large group of random images. This is essentially selecting all of the images that the user recognizes, distinguishing them from the unrecognizable random images.

This scheme has also been largely ignored when attempting to solve the SSA problem. A recognition-based graphical password requires numerous images to be displayed, one of which represents the recognizable correct input. If the user only selects one image per input, it will be just as easy for an attacker to identify and select this same image. Many attempts have been made to disguise the correct image, or

Figure 3: Cogno-metric examples: Deja Vu and PassFaces.

distract the attacker with false positives. However, these additional efforts have been shown to confuse the user as well, and in turn reduce usability. This is another example of the password problem.

2.2.3 Loci-Metric Schemes

Loci-metric GUA schemes are also known as click-based graphical password schemes [6]. This is the original scheme proposed by Blonder, and is based on the popular mnemonic the Method of Loci. This is a technique in which a person visualizes some item they are trying to remember in different spacial locations. To do this, the person associates this item with a landmark, which helps them recall the items later. In a loci-metric scheme, the user is provided with an image so that they can select any location on the image as their password click point. For successful login, the user must select the right click points in the correct order.

Unlike the previous schemes, loci-metric schemes have been evaluated and proposed as solutions to the SSA problem. There are many reasons for this, one of which being loci-metric schemes' ability to avoid password hot spot regions. A hot spot can accure when the password image includes an element that draws too much user attention, which adversely effects security by reducing the effective password space. Another important factor is that this scheme does not require the password input to be highlighted on the screen. Instead of drawings, or an entire correct image being displayed, this scheme records a single click-point as the password input. This correct input is surrounded by hundreds of additional click-points, requiring an attacker to be more precise in their observation.

While all Graphical User Authentication schemes have the ability to be SSA resilient, previous research has shown that loci-metric is the only scheme capable of retaining usability [6] [5]. With this observation, we will limit the scope of this work to be within loci-metric GUA. The following section will include a detailed description of previous GUA models, to better understand the attempts at solving the usable vs. secure, and shoulder surfing issues.

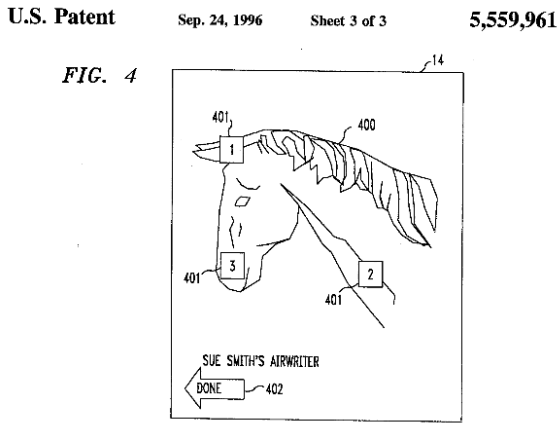

Figure 4: The original graphical password scheme.

2.3 Related Works in Loci-Metric GUA

The focus of these descriptions will be a high-level understanding of the model, followed by how each fell short of achieving SSA resilience while maintaining usability. In short, are these models both usable and secure?

2.3.1 Blonder

The first graphical password scheme was proposed by Blonder in 1996, and required the user to click on a predetermined area of a predetermined image [1]. Despite its concrete nature, the Blonder scheme still possessed advantages over alphanumeric passwords. As stated earlier, users find it easier to recall images, particularly an image with a personal significance. However, this system did not make substantial effort to protect against known security attacks. The predefined regions need to be easily recognizable by the user, and the password space is limited as there are a small number of these regions. The only way to increase security within the Blonder system is to increase the number of click-points, which increases monotony and adversely affects usability. In summary, this system did not make any attempt at solving the SSA problem. When tested, the results show that this system achieves less resistance to SSA when compared to alphanumeric authentication.

2.3.2 PassPoints

PassPoints, proposed by Wiedenbeck et al. expanded Blonders system [14]. This approach removes all predefined aspects of graphical passwords, allowing for the user to select a personal image. Furthermore, a user may select any region of the image as their input. These changes significantly increase the password space, and any attempt to crack a users password must be done on a personal basis. However, this system

resulted in a significant increase in login time, which was accredited to the low tolerance i.e. small click points. Due to its design, PassPoints is vulnerable to SSA, because the attacker can see the click points directly during authentication. Again, the only way to increase security in this system is to increase the number of click points, resulting in even longer creation and login times. In summary, this system was more robust against most security attacks due to the increased password space, but no measures were conducted to prevent the SSA problem.

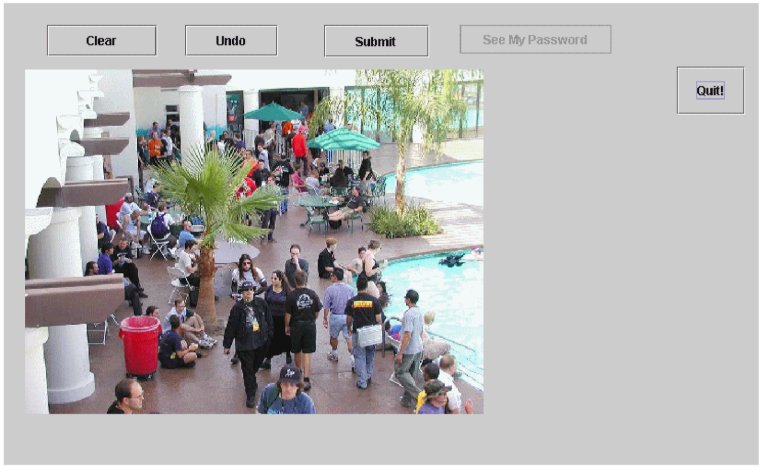

Figure 5: The PassPoint Interface.

2.3.3 Cued Click Points

Cued Click Points (CCP), proposed two years later by Chiasson et al. continued to expand graphical passwords' robustness [3]. In this system, the user selects only one point per image for a sequence of images. The following image is displayed based on the location of the last click point, so each image after the first is a deterministic function of the current image and the coordinates of the user-entered click point. If the user clicks an invalid click point the following image will not be their password image. This functionality is meaningless to an attacker, yet provides significant feedback to the user on their authentication progress. This change in design had positive effects on usability and removed a major concern in graphical passwords. In previous systems one image was used, and users tended to choose click points in a pattern i.e. a circle or straight line across the image. The change to single click points across a sequence of images removed patterns and lowered the effect of hot spots. While this is arguably the greatest development in graphical passwords, the researchers claim it is worse than PassPoints at preventing SSA attacks. They claim that Cued Click Points is susceptible to (SSA) attacks, and indeed in its present form the change in images may be easier to see from further away than mouse pointer movements in PassPoints [3]. In sum-

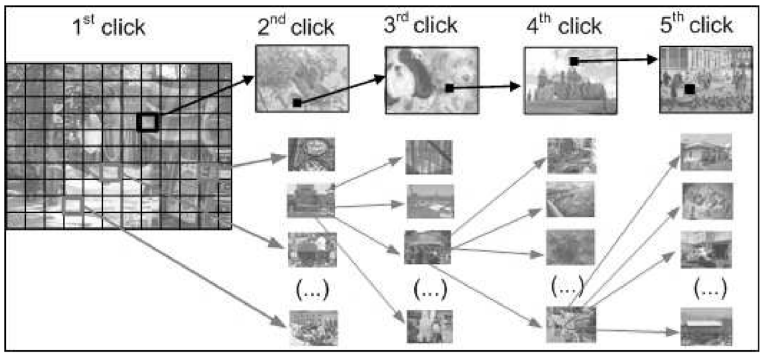

Figure 6: Cued Click Points: Adding the cue allows for shoulder surfing attacks.

mary, this system expanded GUA security a considerable amount, and in turn increased the possibility for GUA to be applied in the real world. They increased the password space, removed hot spots, patterns and added user feedback. However, this system is worse at preventing SSA, which still remains the greatest concern in GUA.

2.3.4 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) image blending, was proposed for use in graphical passwords by Miyachi et al. in 2010 [8]. The proposed method uses characteristics of human vision to achieve resilience towards SSA. By blending low frequency components of a decoy image with high frequency components of a password image this method prevents direct observation of authentication. This is because it is easy for the legitimate user to see the password image when they are close to the screen, but a few feet back this password image is not distinguishable from the decoy image. This means an attacker must be extremely close to the user for direct observation, and recording the authentication session is useless. This is the first loci-metric scheme that achieves effective SSA resilience. However, Miyachi et al. released only four pages of findings, as the project entailed a low fidelity test evaluating memorability and robustness against SSA. The study did not evaluate usability to the same degree as the previous related work. In summary, the proposed system achieved SSA robustness and proved the well known characteristic of memorable graphical passwords. However, for this method to applied in the real world an informative user study must be implemented, and usability must be upheld.

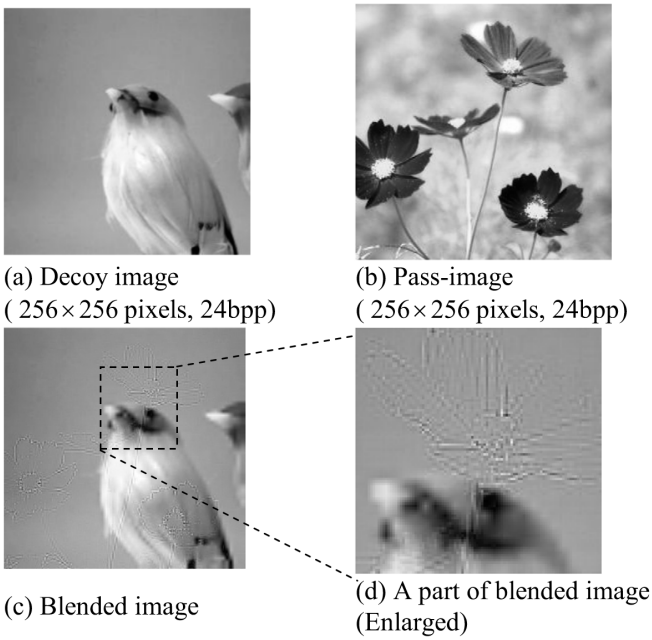

Figure 7: Blending images through discrete wavelet transform.

3 Design and Implementation

3.1 Motivations

Related works were evaluated, with the goal of finding a GUA system that is both usable and secure against the most common security attacks. Through this evaluation, PassPoints was identified as the best empirical test, and most related to the password and SSA problems described previously. PassPoints is a loci-metric scheme, which has been identified as the scheme with the greatest ability to achieve shoulder surfing resilience. Wiedenbeck et al. were some of the only researchers that directly compared their proposal to alphanumeric password authentication. This comparison evaluated both usability and security, claiming that PassPoints is more secure and that it was able to uphold many usability metrics in an empirical test [14]. The design of PassPoints, and its in depth evaluation make it the best system to conduct further research on.

By implementing an instance of PassPoints, the current researcher can make inferences as to how a newly proposed system would compare to alphanumeric password authentication. Key design decisions were taken from PassPoints, including the image size, number of click points, and the layout of the GUA interface[11]. Therefore, the proposal of PassPoints by Wiedenbeck et al. motivated the implementation of PassMatrix, a system proposed in the current study. This system does not make a direct attempt to solve the SSA problem, just as the authors of PassPoints. The only difference between these two systems is their

look and feel, and the click point tolerance being increased to 100x100. This change was made so that the participants in the current study are not confused by a grid of small cells, overlaid on their password images.

The authors of PassPoints also conducted a longitudinal evaluation of GUA memorability [14]. A well designed longitudinal study was not applicable given the constraints of the current research, so the ability to make inferences towards the memorability of loci-metric GUA is very important. PassPoints also motivated much of the experimental design, specifically the usability metrics recorded. By implementing a system based on the design and evaluation of PassPoints, many things were quickly achieved, allowing for further investigation of the described problems. First, PassMatrix could be compared to alphanumeric authentication, which is important because this form of authentication is used most in the real world. Second, the strong memorability characteristics of GUA could be observed directly, without the cost of a longitudinal study. Finally, PassPoints was one of the first GUA systems to be proposed, and ended up setting a base level of security that future GUA proposals compared themselves too.

One of these proposals that followed PassPoints was Discrete Wavelet Transform. While this study did not evaluate usability, Miyachi et al. evaluated security and directly compared their robustness against SSA to that of PassPoints. Testing for SSA robustness was done by pairing participants together, where one acted as the attacker and the other a legitimate user. This test proved that using high-frequency components of a password image can prevent an attacker from directly observing a login attempt. By implementing PassDecoy, a system motivated by the findings of DWT, the current researchers can make inferences as to how a newly proposed system would be able to prevent a SSA. The current study makes inferences on what past studies have proved, specifically that GUA is more secure and more easily memorized when compared to alphanumeric authentication. Furthermore, that using high frequency image blending can prevent the direct observation of GUA. These inferences allow the current study to investigate what past research did not, the effect of increased SSA security on graphical user authentication usability.

3.2 Hybrid Imagery

Hybrid images are based of the multi-scale processing of images by the human visual system. This technique produces a static image with two interpretations, which change as a function of viewing distance [9]. These blended images are used in a GUA interface in which the image appears to change in relation to the viewing distance from the image. Perhaps the most well known instance of a hybrid image is the Mona Lisa by Leonardo da Vinci, where the subject appears to smile / frown dependent on the viewing distance.

For this experiment, hybrid images were created in Adobe Photoshop. Two sets of images with different

Figure 8: Moving from left to right, a low frequency filter is applied.

characteristics were selected for blending in PassDecoy. The first set used in hybrid imagery is a decoy set. This set includes simple images such as scenery, still life photography, and zoomed in pictures of objects. This set of images must be completely colored, as any black or white regions will result in the next set of images to be indistinguishable from the decoy image. A low frequency filter is applied to these images, as well as some blurring and brightening to optimize each individual image. This technique creates images that are blurry when viewing distance is small, but look normal at greater distances. This process is demonstrated in Figure 8.

The next set of images is the password set. This set includes complex images such as a fruit stand, or a surface covered in money. A requirement of this set is that prominent objects must be displayed across the entire image, so that easily observed hot spots do not result. This is an extremely important requirement, because an image containing hot spots results in an effective password space that is much smaller than the theoretical password space. A high frequency filter is applied to this image, as well as lowering to its saturation to optimize the desired effect. This creates colorless images, containing only the high frequency components of the image. When viewing distance is small, all of the elements in the image can be seen, but the color has been removed. When the viewing distance is large, all of the elements in the image are hidden and it is perceived as a blank gray image. This process is demonstrated in Figure 9.

The application of hybrid imagery has, so far, resulted in a set of decoy images, and a set of password

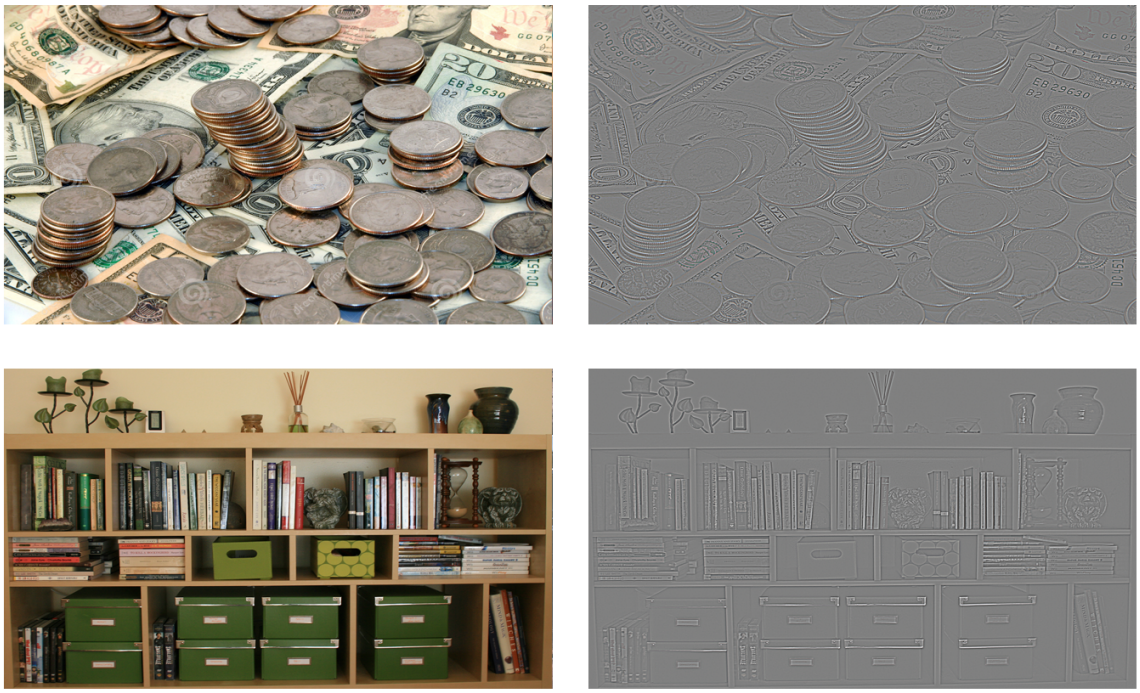

Figure 9: Moving from left to right, a high frequency filter is applied.

images. The password image is then overlaid on the decoy image, resulting in a single hybrid image. This image contains the high frequency elements of the password image blended with the low frequency elements of the decoy image. As previously mentioned, the hybrid image is a static image with two interpretations, which change as a function of viewing distance. The concept of optimization was mentioned in the creation of decoy and password images, and this concept will be described below.

The desired effect of the hybrid images is a static image, that is completely viewable from 1-2 feet, or the distance of the legitimate user [12]. However, as viewing distance increases to that of the attacker, say 5+ feet, the password elements must be hidden. To achieve this effect both the decoy and password images needed to be optimized. For the decoy images, all of the images did not start out at the same definition, so high definition images needed to be blurred more than low definition images. Additionally, very dark images needed to be brightened, because black or nearly black regions on the decoy will not allow for the password elements to be visible. The need for colored images without any black or white regions was described previously. Optimizing the password images was related to the severity of the high frequency filter. This severity was chosen so that the password elements were visible from a small distance, and hidden at any other distance. In a high definition image, the severity of the filter was relatively high, and all color was removed from the image. However, a low definition image required a lower severity of the high pass filter, so that the image was still visible at a small distance. This lowered severity would leave

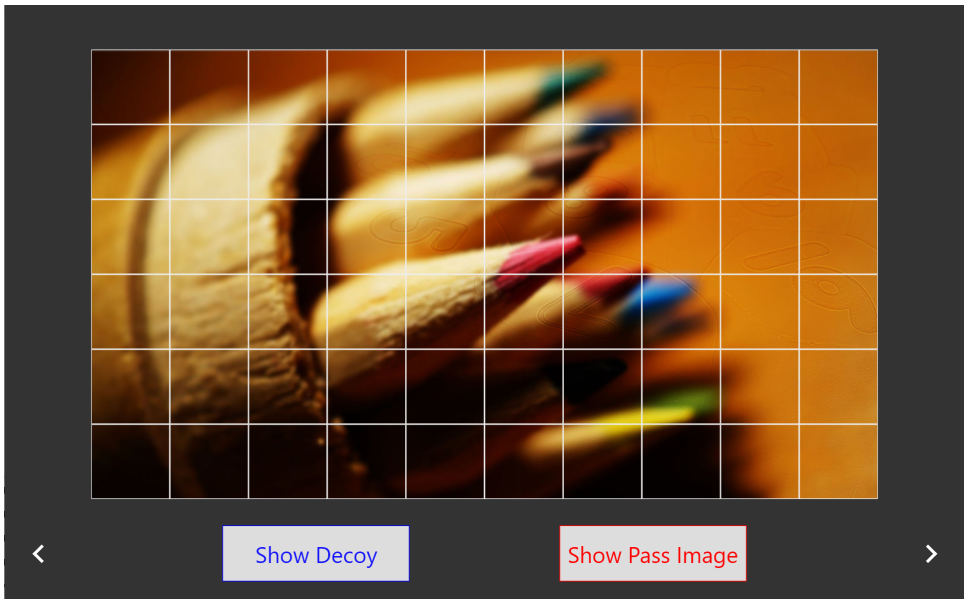

Figure 10: Hybrid GUA Prototype.

colored elements in the password image, which is where changing the saturation was used to remove any color. Removing the color of the password elements is necessary, as coloring the password elements would alert an attacker toward the presence of these elements, even when viewing distance is large and they cannot specifically make out any objects.

While optimizing both the decoy and password sets of images, the zoom on Photoshop was held at 85

3.3 PassMatrix and PassDecoy

The first steps taken in the implementation of shoulder surfing resilient GUA systems was a proof of concept. This was necessary because previous work was used to make inferences on the systems robustness against shoulder surfing. A proof of concept was achieved through a Hybrid GUA prototype. This was a low fidelity interface implemented using the same tools as PassMatrix and PassDecoy. It contained three password images, three decoy images, and the combined 3 hybrid images. The only functionality was the ability to show only the decoy, and only the password image before they were blended. This prototype showed that a static image could change as a function of viewing distance, and was used to legitimize the proposal of PassDecoy before attempting any higher cost implementations.

With a proof of concept achieved, implementing PassMatrix was the next task. The main tool used in the implementation was Adobe Experience Design. This is part of the Adobe creative suite, so the images created in Adobe Photoshop could be exported, and inserted directly into the interface without changing its dimensions. Every individual web page was designed using this tool, from the registration pages to the

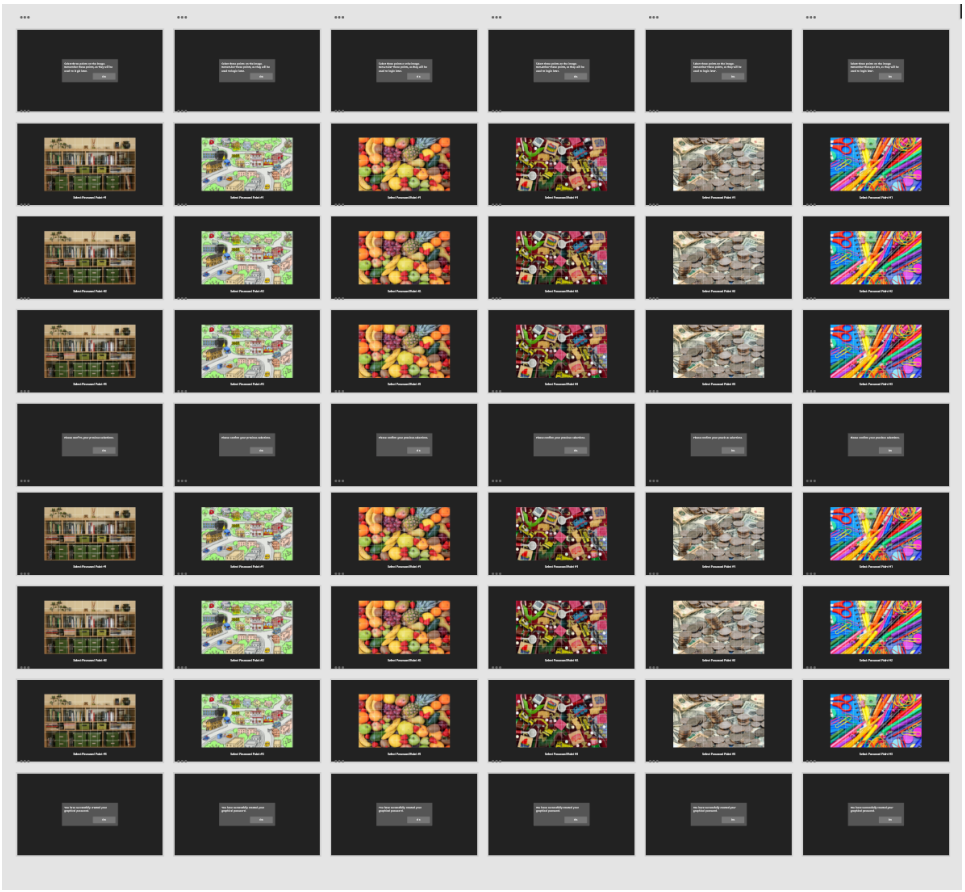

Figure 11: Each image that a user may select has its own user flow.

error messages. This resulted in prototypes that could be produced quickly, which was important because many iterations were attempted before landing on the final design, i.e. a design that worked just like the previously studied PassPoints. With the final design of PassMatrix set, its functionality needed to be implemented.

The authentication protocols functionality was achieved through a few different means. First, the users individually selected image was continuously displayed by creating 6 distinct user flows, dependent on which image they selected. This circumvented the need to record the image selection, and then present the correct image on the remainder of the web pages. The implementation of 6 distinct user flows can be seen in the figure below.

To add the successful versus failed login functionality the Bootstrap 4.0 framework included in Adobe was used. Bootstrap is an open source toolkit for developing with HTML, CSS and JS. The password sequence was recorded using Bootstraps form tags. This framework allowed for the users click points to be recorded as a cell number, and this sequence of cell numbers could be stored in a PHP file to be referenced later. During a login attempt, another sequence would be recorded. If these two sequences matched, then

the user was directed to a successful login message. If the users sequences did not match then they were presented with a failed login message. Confirming, or clicking OK on the messages returned the user to the top of their individual user flow, so that the image the selected was presented for the next login attempt. This was all of the functionality included in PassMatrix.

To begin the development of PassDecoy, the Adobe .xd file was duplicated and renamed to reflect the second system. This meant that all of the design decisions used in PassMatrix were also present in PassDecoy. The instance of the decoy image was isolated as the only change between the two systems, so that this specific change would be the only cause of variation in usability. The randomization of the decoy image was also achieved through some low fidelity implementation. Each user was tasked with inputing their three click point password three times. This meant that only nine images were needed to resemble a continuously random decoy. Nine hybrid images were created for each of the possible password images. The nine hybrid images all included a different decoy image, and these images were linked together to create the login protocol. The process of displaying a successful versus failed message was achieved in the same manor described previously. These steps resulted in a GUA interface, that included the functionality of a personally selected image, checking a current login sequence with a previously registered sequence, and a seemingly randomized decoy image. A personally selected image is an important feature to implement because a user selecting their own image is needed to uphold the strong memorability of GUA. Adobe Experience Design was also able to publish links to the developed systems. This added a lot to the overall system, because the participants interacted with the systems using the Chrome browser as they would have in the real world.

4 Methodology

4.1 Experimental Design

This experiment was conducted across two weeks and involved 20 members from a university community. Each participants was evaluated individually, in one ten minute session. The experiment involved a within-subjects design, where each participant interacted with both systems. To limit the effects of learning, the systems were randomly administered, resulting in 10 users interacting with PassMatrix first, and 10 users interacting with PassDecoy first. Regardless of the treatment order, the participants completed a registration phase prior to using either system. This phase was completed so that the participants could register their graphical password, but also so that they could practice using the interface prior to being evaluated on usability metrics described in the next section.

4.2 Usability Metrics

Usability is evaluated in three general categories: Effectiveness, Efficiency and Satisfaction. These metrics are like metrics used in previous studies, so that a broad comparison followed the user study [3]. Each category and the specific metrics are described below.

Category	Measure	Description
Effectiveness	# of Retries	User fails a login attempt.
	# of Errors	User adds to the elapsed time through an incorrect action.
Efficiency	Registration Time	Time elapsed for a successful registration.
	Login Time	Time elapsed for a successful login.
Satisfaction	Questionnaire	Seven questions to obtain user sentiment.

Table 1: Measures to be recorded in user study.

Effectiveness is evaluated using two metrics. The first is the average number of retries. This is defined as any time the user selects an incorrect click point and must re attempt authentication. The second metric is the number of errors. An error is defined as something that does not fail the login attempt, but adds to the elapsed time for authentication. Effectiveness highlights whether it is difficult or trivial to use each system.

The next usability focus is Efficiency and is evaluated using two metrics. The first is the time elapsed during registration. The second is the time elapsed during a successful login attempt. These metrics were recorded in seconds. Login and creation times have been evaluated in the majority of GUA user studies, because a system must be efficient to be applied in the real world.

The last usability focus is user satisfaction. This was evaluated using a brief questionnaire, to obtain the user's general sentiment on each system. Responses were recorded using 1-5 Likert Scale responses, 1 being strongly agree and 5 being strongly disagree. Whether the given statement was inputting my password was fast or inputting my password was slow there will be some effect of a leading question. To limit the effects of leading statements, the words strongly agree and strongly disagree were displayed next to the statements. This was attempt to show users they did not have to agree with the statement, and in turn limit the effects of any leading statements. The statements used to evaluate satisfaction are listed below, and their presentation can be seen in Figure 12.

1. It did not take me long to input my password 3 times.
2. Once I created my password, I was able to input it correctly
3. Registering my password was fast.

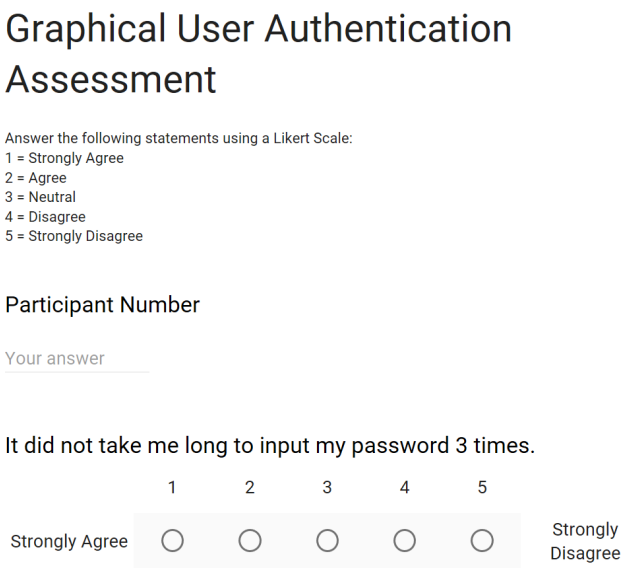

Figure 12: The Google Forms used to evaluate user satisfaction.

4. Inputting my password was easy.

5. My password images are easy to memorize.

4.3 Materials

This experiment was conducted in the CROCHET lab at Union College. The test was administered on a Windows 7 Enterprise 64 bit operating system, and a Dell U2211h 21.5 inch LCD monitor. A Logitech web cam was set up in the back of the room, and aimed at the Dell monitor. The Google Chrome browser was set to full screen and used to visit both the system and questionnaire web pages. Adobe Experience Design hosted the PassMatrix and PassDecoy web pages, while Google Forms hosted the questionnaire web page. The participants navigated these web pages through bookmarks on the Chrome browser.

Morae recording software provided by Union college was used to record the entire test session. This software was configured to begin the recording when Chrome was opened, and end the recording when Chrome was closed. This resulted in the participants opening the browser, completing the experiment, and exiting the browser without being affected by the recording software.

4.4 User Test Procedure

Testing was done individually in a quiet human computer interface lab. Participants were randomly assigned to use either the PassMatrix and PassDecoy condition first. Each individual participated in one

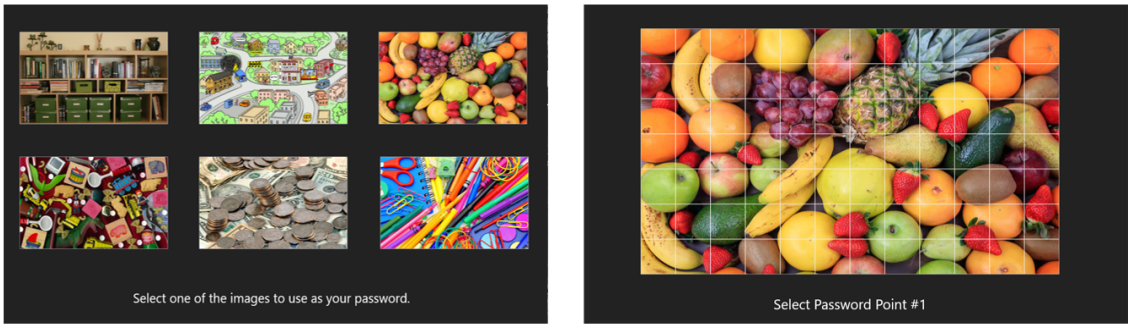

Figure 13: Examples of the Registration phase.

session, lasting about 10 minutes. The participants were introduced to the appropriate purposes and procedures of the experiment through a proctor reading parts of a test script. This test script can be seen in the appendix. Consent was received to record the screen, and the tests followed.

In the password registration phase the participant was instructed to register their password by following the instructions on screen. Users who selected login received an on screen instruction to register their password first, and this is an example of an error. Users who selected register were asked to select one of six pictures to use as their password input. Once an image was selected, the users registered a 3 click-point password sequence. A sliding transition was used in between click points, to provide the user with feedback that their click point was registered. After this transition, the user was asked to select their next click point. After creating a password sequence the users were asked to confirm their previous selections. If the two sequences matched, the user had successfully registered their password and was instructed to move onto the login phase. Within these next set of instructions involved the proctor reading the following instruction for all participants:

> I have just turned on a web cam in the back of the room. This web cam is aimed at your screen, just as a human would look at your screen when attempting to steal your password. You will now input the password you just registered. Keep in mind that someone could be on the other side of that web cam and may be attempting to steal your password. I would like you to login 3 times using your password.

The described webcam purpose was not to record any video. Instead, it was used so that the users believed their login attempt may be attacked, resulting in them protecting this password as they would in a more natural setting. While the user was interacting with the assigned system, the proctor was recording the number of failed login attempts and errors made by the users. Once a user had 3 successful login attempts they were verbally instructed to navigate to the questionnaire, to complete the satisfaction portion

of the test.

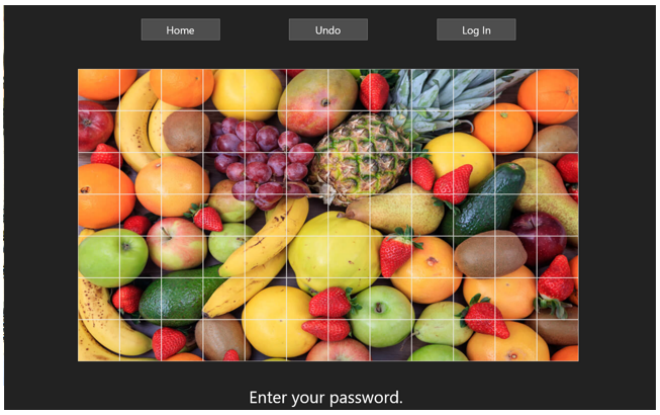

Figure 14: Examples of the Login phase.

The steps taken in the previous paragraph were done a second time for the second condition. The user was again read the statement describing the deception, and instructed to input their password 3 times. Successful and failed login attempts were communicated to the user through on screen messages. After interacting with the second system the users completed the questionnaire a second time. Login times were calculated after the session using the screen recordings, and Google forms compiled the satisfaction responses for each participant.

Following the administered tests, the participant was debriefed on the nature of the study. This debrief thanked them for their participation and explained the small amount of deception used. The scripted debrief can be seen in the appendix.

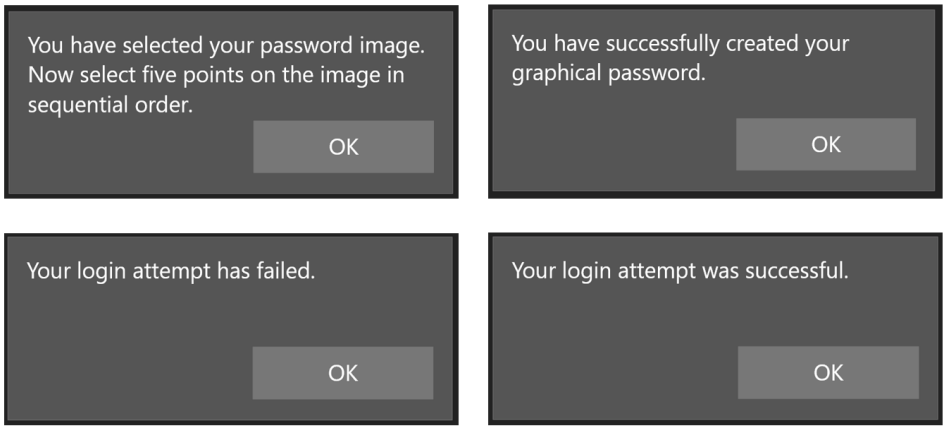

Figure 15: Examples of the possible messages a user recieved.

5 Results

The results are reported below by usability metric category: Effectiveness, Efficiency and Satisfaction. Any user comments recorded by the proctor are not reported here, but were taken into account during the discussion of the results. This experiment was a paired comparison, where the value of a metric for PassMatrix was paired with the value of the same metric for PassDecoy. These comparisons were done for the values of one subject. To eliminate any structural variation, where one participant recorded their satisfaction consistently higher than another participant, the difference in usability metric was evaluated; i.e. the result of PassMatrix the result of PassDecoy. This was applied to all of the metrics, and resulted in a consistent means to answer the question of whether adding the decoy reduced, maintained or improved usability. A negative difference signifies that PassDecoy is less usable, a difference of zero signifies no change between the systems, and a positive difference signifies that PassDecoy in more usable.

5.1 Effectiveness

On the number of user errors and number of failed login attempts, there is insufficient evidence to demonstrate that there is a difference between the two systems. A 65 percent majority of participants had no difference in the number of user errors (p = .716), and 80 percent of the participants had no difference in the number of failed login attempts (p = 1).

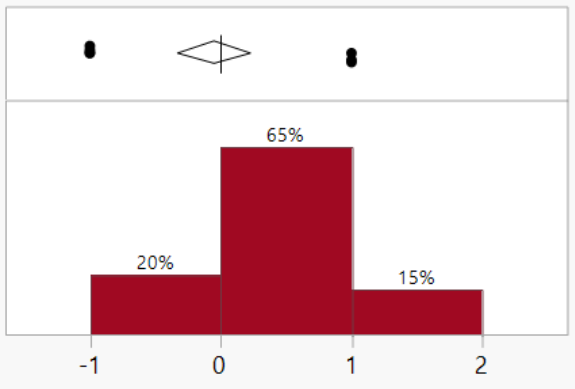

Figure 16: Number of User Error Difference.

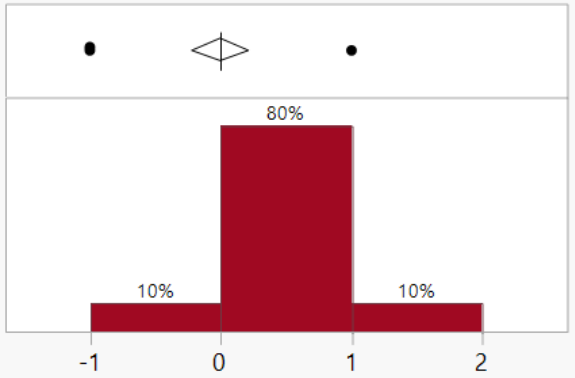

Figure 17: Number of Failed Login Attempt Difference.

5.2 Efficiency

On the elapsed time for a participant to login, there is sufficient evidence to demonstrate a difference between the two systems. An overwhelming 80 percent of the participants took longer to authenticate themselves when using PassDecoy (p = .004) and this result is statistically significant. With a confidence of 95 percent, it can be said that PassDecoy will take users an additional .25 to 1.13 seconds per login attempt. To demonstrate this result, the login time for PassMatrix was ordered, and plotted against the login time for PassDecoy. The variation between the two systems appears to be reduced as login time is increased.

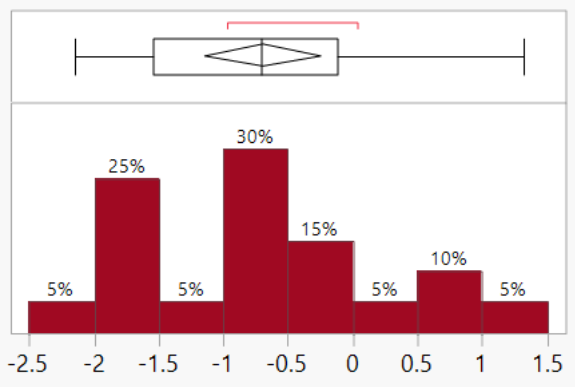

Figure 18: Difference in the Average Login Time.

5.3 Satisfaction

On statement #1: It did not take me long to input my password 3 times, there is insufficient evidence to demonstrate a difference between the two systems. 80 percent of the participants did not record a difference in their login time satisfaction (p = .330). On statement #2: Once I created my password I was able to input it correctly, there is insufficient evidence to demonstrate a difference between the two systems. 75 percent

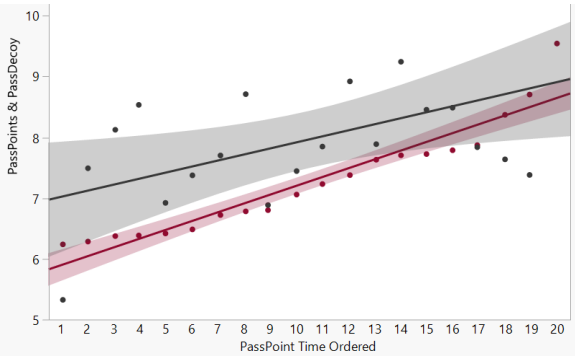

Figure 19: PassMatrix vs PassDecoy Login Time.

of the participants did not record a difference in their perceived ability to input the passwords correctly (p = .666).

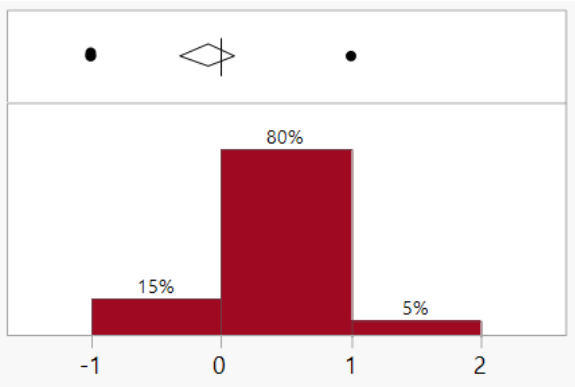

Figure 20: User Satisfaction - Statement #1

On statement #3: Registering my password was fast, there is insufficient evidence to demonstrate a difference between the two systems. 60 percent of the participants did not record a difference in their registration time satisfaction (p = .330). On statement #4: Inputting my password was easy, there is insufficient evidence to demonstrate a difference between the two systems. 55 percent of the participants did not record a difference, while 30 percent of the participants found PassDecoy to be harder to use (p = .330).

On statement #5: My password images are easy to memorize, there is sufficient evidence to demonstrate a difference between the two systems. 35 percent of the participants did not record a difference, while 55 percent of the participants found it harder to memorize their password in PassDecoy (p = .007). This result showed statistical significance.

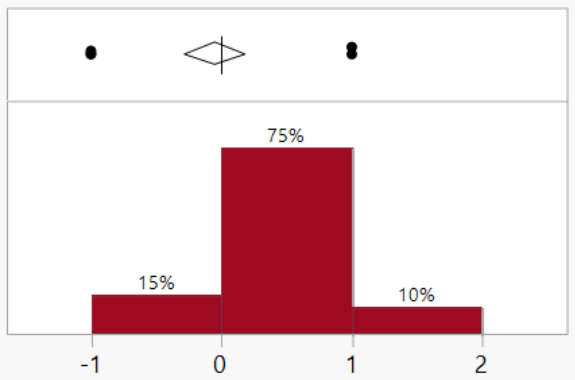

Figure 21: User Satisfaction - Statement #2

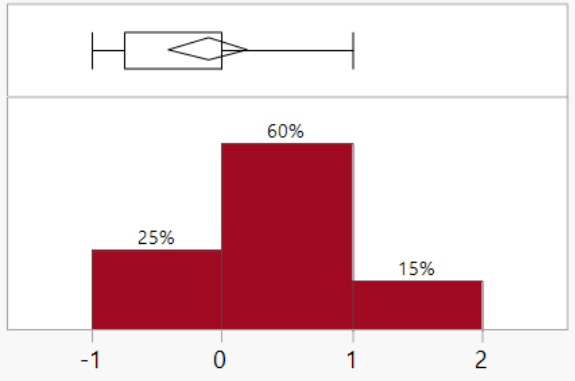

Figure 22: User Satisfaction - Statement #3

6 Discussion

PassDecoy has the security advantages of a large password space when compared to alphanumeric authentication. This advantage is also observed over Blonder and recognition-based graphical password. It is also robust against direct observation and shoulder surfing attacks. However, usability is a critical consideration described in the password problem. This is because a system must be efficient and effective to be implemented in real world applications. The results of the user test conducted on PassDecoy yielded mixed results.

There was insufficient evidence to demonstrate a difference for most of the metrics recorded. While this appears to show usability was upheld, it may also show that the type of data recorded may not allow for stronger statistical inference. This was especially apparent in the number of errors and number of failed login attempts. The definitions of an error may have been too specific, so the overall number of errors was very small. This may also be attributed to the simplicity of the proposed systems. A new definition of an error is appropriate, but the proposed systems simplicity is also very important because it reduced the overall variation of the results. This meant that the addition of the decoy image caused the majority of the

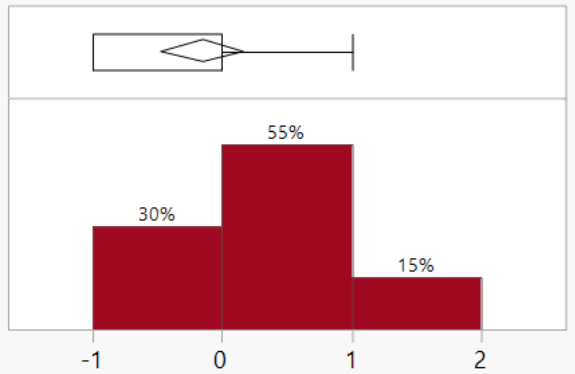

Figure 23: User Satisfaction - Statement #4

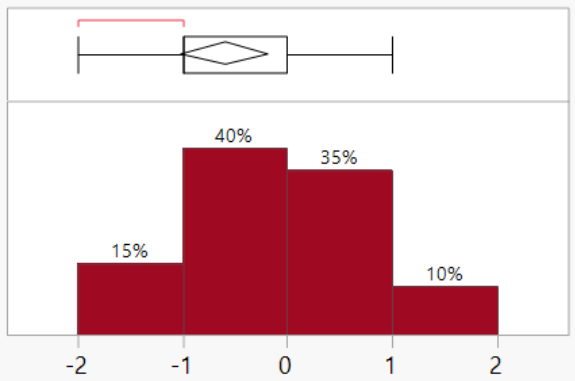

Figure 24: User Satisfaction - Statement #5

variation in the results.

The results showed that PassDecoy will take users an additional .25 to 1.13 seconds per login attempt. However, this was a difference that the users did not perceive. On the statement: Registering my password was fast, 75 percent of the users tested thought that Passdecoy was the same, or better than PassMatrix. The remaining 25 percent were able to perceive this difference. This is an example of why evaluating user satisfaction is important, because additional conclusions can be made.

7 Conclusion and Future Work

The proposed PassDecoy system shows promise as a usable and secure authentication mechanism. By taking advantage of a user's ability to remember strong graphical passwords, decoy image blending, and a large password space PassDecoy has advantages over alphanumeric and previous graphical authentication. The decoy image effectively lessens the possibility for a shoulder surfing attack, a severe security threat observed in previous work. In the user study, the relative usability of PassDecoy was determined, and shows that future work is needed before the system is acceptable for real world application.

The most obvious future work is related to improving the experimental design. Additional metrics, specifically continuous data is needed to make greater statistical inferences. Currently, the majority of the results state insufficient evidence to demonstrate a difference. This is because the hypothesis of this study is to accept the null, or the lack of an effect of increased security on GUA usability. In the future additional experiments would be designed to look for an effect. This could be evaluating how removing security features may add to usability.

One major take away from the study was that users believed it was harder to memorize their passwords when the decoy image was used. In session comments by participants, as well as post-study evaluations of PassDecoy reveal that users were associating color-based memory cues with their password click points. When the participant attempted to login using their memory cue of the yellow cement truck, they found that the color had been removed from their password image to achieve the desired effect of hybrid imagery. In future works, the color will be removed from the registration phase to better resemble the password images in the login phase.

Another important addition to future work will be implementing a test to evaluate how differences in visual capability effect the usability of PassDecoy. In practicing good design principles, you must design your system to be accessible by all people. Currently, the proposed PassDecoy system is heavily reliant on components of the human visual system, but variation in human visual system capability was not accounted for. First, a test must be conducted to determine the degree of effect on usability. If it is observed that it has a large effect, the system must be designed to be adaptable, possible through the incorporation of a vision test during registration.

The last element within future work is additional user tests, to see if login time can be reduced through practice. This would be a fairly straightforward test, where the original 20 participants are evaluated in a longitudinal study, to evaluate the effect of learning and practice on login times. Figure 19 shows that when login times a short, PassMatrix can be more than a full second longer than PassDecoy. When login time is long, the difference between the two systems is much smaller. This hints PassMatrixs ability to be learned more than PassDecoy, which would be a serious threat to usability. This relationship needs to be evaluated further through a longitudinal study.

In conclusion, the contributions of this work can be divided into two categories, usability and security. With respect towards security, it was described that the dificulty of maintaining strong alphanumeric passwords has lead to poor practices and lower overall security. Yet, Graphical User Authentication has an observed robustness against many security attacks. It is easy to obtain large password spaces, without convoluted requirements, because images have hundreds of theoretical click points. With the proposal of PassDecoy, the system was able to prevent direct observation and the shoulder surfing attack. With respect

to usability, which is the main focus of this study, the results indicate that users can quickly and easily create graphical passwords in PassDecoy. The majority of the participants were able to quickly learn to input the password sequences, limiting the number of errors and failed logins. Lastly, the addition of a decoy image results in slower login times, and may have adverse effects on memorability. These observations motivate future work on PassDecoy, and the field of usable security in general.

References

[1] G.E. Blonder. *Graphical password*. US Patent 5,559,961. Sept. 1996. URL: https://www.google.com/patents/US5559961.

[2] S. Chiasson et al. "Persuasive Cued Click-Points: Design, Implementation, and Evaluation of a Knowledge-Based Authentication Mechanism". In: *IEEE Transactions on Dependable and Secure Computing* 9.2 (Mar. 2012), pp. 222–235. ISSN: 1545-5971. DOI: 10.1109/TDSC.2011.55.

[3] Sonia Chiasson, P. C. Van Oorschot, and Robert Biddle. "Graphical Password Authentication Using Cued Click Points". In: *Proceedings of the 12th European Conference on Research in Computer Security*. ESORICS'07. Dresden, Germany: Springer-Verlag, 2007, pp. 359–374. ISBN: 3-540-74834-2. URL: http://dl.acm.org/citation.cfm?id=2393847.2393880.

[4] Sonia Chiasson et al. "Multiple Password Interference in Text Passwords and Click-based Graphical Passwords". In: *Proceedings of the 16th ACM Conference on Computer and Communications Security*. CCS '09. Chicago, Illinois, USA: ACM, 2009, pp. 500–511. ISBN: 978-1-60558-894-0. DOI: 10.1145/1653662.1653722. URL: http://doi.acm.org/10.1145/1653662.1653722.

[5] Sonia Chiasson et al. "User interface design affects security: patterns in click-based graphical passwords". In: *International Journal of Information Security* 8.6 (May 2009), p. 387. ISSN: 1615-5270. DOI: 10.1007/s10207-009-0080-7. URL: https://doi.org/10.1007/s10207-009-0080-7.

[6] Fei Ye Haichang Gao Wei Jia and Licheng Ma. "A Survey on the Use of Graphical Passwords in Security". In: *Journal of Software* 8.7 (2013), pp. 1678–1698. URL: https://pdfs.semanticscholar.org/774c/7ac9d7bd9c73ff16c8b2cc8a21ae08739371.pdf.

[7] Janna Lynn Dupree et al. "Privacy Personas: Clustering Users via Attitudes and Behaviors toward Security Practices". In: *Proceedings of the ACM Conference on Computer and Human Interaction*. CHI '16. May 2016, pp. 5228–5239.

[8] T. Miyachi et al. "A study on memorability and shoulder-surfing robustness of graphical password using DWT-based image blending". In: *28th Picture Coding Symposium*. Dec. 2010, pp. 134–137. DOI: 10.1109/PCS.2010.5702441.

[9] Aude Oliva, Antonio Torralba, and Philippe G. Schyns. "Hybrid Images". In: *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06. Boston, Massachusetts: ACM, 2006, pp. 527–532. ISBN: 1-59593-364-6. DOI: 10.1145/1179352.1141919. URL: http://doi.acm.org/10.1145/1179352.1141919.

[10] Elizabeth Stobert et al. "Exploring Usability Effects of Increasing Security in Click-based Graphical Passwords". In: *Proceedings of the 26th Annual Computer Security Applications Conference*. ACSAC '10. Austin, Texas, USA: ACM, 2010, pp. 79–88. ISBN: 978-1-4503-0133-6. DOI: 10.1145/1920261.1920273. URL: http://doi.acm.org/10.1145/1920261.1920273.

[11] H. M. Sun et al. "A Shoulder Surfing Resistant Graphical Authentication System". In: *IEEE Transactions on Dependable and Secure Computing* PP.99 (2016), pp. 1–1. ISSN: 1545-5971. DOI: 10.1109/TDSC.2016.2539942.

[12] Julie Thorpe et al. "The Presentation Effect on Graphical Passwords". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 2947–2950. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557212. URL: http://doi.acm.org/10.1145/2556288.2557212.

[13] Susan Wiedenbeck et al. "Authentication Using Graphical Passwords: Effects of Tolerance and Image Choice". In: *Proceedings of the 2005 Symposium on Usable Privacy and Security*. SOUPS '05. Pittsburgh, Pennsylvania, USA: ACM, 2005, pp. 1–12. ISBN: 1-59593-178-3. DOI: 10.1145/1073001.1073002. URL: http://doi.acm.org/10.1145/1073001.1073002.

[14] Susan Wiedenbeck et al. "PassPoints: Design and longitudinal evaluation of a graphical password system". In: *International Journal of Human-Computer Studies* 63.1 (2005). HCI research in privacy and security, pp. 102–127. ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2005.04.010. URL: http://www.sciencedirect.com/science/article/pii/S1071581905000625.