

# Security Analysis of Picture Gesture Authentication

6.857 Project

Clare Liu (clareliu@mit.edu)

Denise Che (cdenise@mit.edu)

Srinidhi Viswanathan (srinidhi@mit.edu)

May 13, 2015

# **Table of Contents**

## **1 Abstract**

## **2 Introduction**

## **3 Background**

### **3.1 Current Authentication Methods**

### **3.2 Windows 8 PGA**

### **3.3 Potential Concerns for PGA**

### **3.4 Current Studies on PGA Security**

#### **3.4.1 Inside look from Windows Engineering Team**

#### **3.4.2 On the Security of PGA**

### **3.5 Computer Visions Algorithms**

## **4 Methodology**

### **4.1 Experiment 1 - User Image Selection**

### **4.2 Experiment 2 - Comparison of Gestures in Simple and Complex Images**

### **4.3 Experiment 3 - Shoulder Surfing**

#### **4.3.1 Shoulder Surfing for PGA**

#### **4.3.2 Shoulder Surfing for Text Passwords**

## **5 Results**

### **5.1 Experiment 1 - User Image Selection**

### **5.2 Experiment 2 - Comparison of Gestures in Simple and Complex Images**

### **5.3 Experiment 3 - Shoulder Surfing**

#### **5.3.1 Shoulder Surfing for PGA**

#### **5.3.2 Shoulder Surfing for Text Passwords**

#### **5.3.3 Comparison of Shoulder Surfing on PGA to Text Passwords**

## **6 Enhancements to PGA**

### **6.1 Feature Detection**

### **6.2 Strength Meter Prototype**

## **7 Limitations and Future Work**

## **8 Conclusion**

## **9 References**

## **10 Appendix**



## 1 Abstract

Picture Gesture Authentication (PGA) is a knowledge-based login scheme introduced in the Windows 8 operating system. Users set up PGA by selecting a background image and drawing three gestures on the image; they then repeat these gestures to login into their laptop every time. Without considering the background image, there are over 1 billion unique three-gesture sequences for a particular password. However, most users will only draw gestures on significant points of interest in order to more easily remember their passwords, which greatly reduces the number of possible password permutations. Picture Gesture Authentication is also believed to be more vulnerable to shoulder surfing and smudge attacks. Given that the Windows 8 operating system is widely used, it is important to investigate the security of Windows's picture gesture authentication and potential concerns.

We conducted various user experiments to better understand image and gesture preferences amongst users, as well as their general opinions towards PGA. We also tested the ease of shoulder surfing and estimated the average accuracies for guesses on both PGA and text-based passwords. Results from these experiments led us to develop a proposed enhancement to PGA: a 'strength meter' for picture passwords using feature detection algorithms. Overall, picture complexity does appear to affect security, and our strength meter should help users make a safer, and more informative choice for their picture in PGA.

## 2 Introduction

Touchscreens allow for different methods of authentication besides the traditional character-based passwords. Various devices have implemented Draw a Secret (DAS) password input schemes to replace the conventional alphanumeric authentication methods. Another new approach is Picture Gesture Authentication (PGA), a login option introduced in Windows 8, which allows users to login to their personal computers by selecting three gestures - taps, straight lines, or circles - on a picture of their choice [1]. Microsoft claims that PGA is a fast and fluid way to login into touchscreen devices, while providing comparable security to existing character set passwords.

The Windows 8 operating system currently runs on over 400 million computers and tablets, so it is important to investigate the security of Windows's picture gesture authentication. Since users are responsible for selecting the image and gestures for PGA, their choices can greatly affect the security of the login system. We performed three user experiments to analyze the relationship between image complexity (number of unique features) and gesture selection. We hoped to gain insight about why users select particular images and gestures in order to develop security enhancements to encourage smarter picture password selection.

In addition to conducting user experiments, we tested various feature detection algorithms in MATLAB in order to quantify the security of a background image selected for PGA. We then used these algorithms to develop a web interface to deliver a strength-meter for picture passwords, similar to strength meters for traditional text-passwords.

## **3 Background**

This section provides a brief overview of how Picture Gesture Authentication works, potential security concerns of PGA, and a summary of commonly used computer vision algorithms for feature detection.

### **3.1 Current Authentication Methods**

There are currently three main forms of authentication used in computer security systems: token based authentication, biometric based authentication, and knowledge based authentication. Token based authentication involves the use of smart cards or key cards, while biometric based authentication involves the use of fingerprints or facial recognition, which can be expensive to implement [1].

The most popular authentication scheme is knowledge-based, which includes both text and picture passwords. Text passwords continue to be the most widely used form of authentication, even though users often tend to pick short passwords or passwords that can easily be hacked. Picture passwords can be divided into two categories: recognition-based and recall-based graphical techniques. Recognition based techniques involve the user having to remember a group of objects they selected from a set of images during the registration phase. Recall-based techniques are slightly different, asking users to repeat a series of actions (such as gestures) on a particular picture that they executed in the registration phase [1].

Many picture password schemes have been proposed in the last couple of decades, such as PassPoints, which is a recall-based technique where users click on images rather than typing a long alphanumeric password. The theory behind adopting picture passwords is that humans typically remember images more easily than text. However, researchers have successfully been able to hack into these click-based schemes by tracking the general hotspots of a picture.

The picture gesture authentication scheme in Windows 8 allows more complex gestures (such as lines, circles, and taps) rather than a simple click, so it is in theory more secure than traditional click-based schemes used in picture password schemes.

### **3.2 Windows 8 PGA**

With the vast array of devices with touch-screen capabilities in the market, providing a fluid mechanism to sign into devices with touch is becoming more important, since typing in passwords can be unwieldy. The new Microsoft Windows operating system provides users with an alternative way to sign in to their personal computers. Instead of entering a traditional character-based password, users are asked to upload an image from their picture collection and to then perform three separate gestures on the image. Each gesture can either be a tap, a line, or a circle. After the image and the gestures are set up, users will be prompted to perform the same three gestures on the image in order to log in to their computers [2].

### **3.3 Potential Concerns for PGA**

Various concerns for PGA have been brought to attention by both researchers and users [3]. One major concern is the ease of shoulder surfing, which involves an attacker who observes the user's gestures and then reproduces them later on to gain access to the system. Other common concerns that have been raised include the smudge attack and the feature selection attack. Touchscreen users are vulnerable to smudge attacks, where the attacker recovers the password by examining the oily smudges left behind by the user. Feature selection attacks involve the creation of a selection function that models users' password selection preferences. The model can then be used to generate a set of likely passwords that would be used against users' systems.

### **3.4 Current Studies on PGA Security**

The Windows 8 operating system which has a picture password authentication system, based on gestures, is currently running on roughly 400 million computers and tablets. Thus, it is important to investigate the security of Windows' picture gesture authentication and potential concerns, considering that it's a widely used operating system. Below summarizes two studies that have been conducted to analyze the security of Windows 8 PGA.

#### **3.4.1 Inside look from Windows Engineering Team**

Sinofsky's *Building Windows 8* blog [2] provides a quantitative analysis on the security of PGA and how it compares to the security of a conventional alpha-numeric password.

To track the gestures in Windows 8, the longer dimension of the background image is divided into 100 segments and the shorter dimension is divided into segments on the same scale. The gestures are recorded according to their square positions on the grid. For taps, the square position is stored. For lines, the positions of the first endpoint and the second endpoint are stored. Finally for circles, the center position and the radius are stored.

According to Sinofsky, if an attacker is unable to gain any information from smudges or points of interest, there are 1,155,509,083 possibilities for a picture password that involves three gestures. This is similar in security to a simple a-z character set password with 5 to 6 characters; however in reality, it is highly unlikely for a person to select anywhere along the picture. There tends to be a limited number of points of interest where most gestures are performed. If an image has  $m$  points of interest and we assume that users only draw gestures on points of interest, and the length of the password is  $n$  gestures, the number of possible password permutations is:  
 $(m \cdot (1 + 2 \cdot 5 + (m - 1)))^n$  [2].

This is because there are  $m$  possible locations for a tap,  $m \cdot (m - 1)$ , possibilities for lines because there are  $m$  possible points for the first endpoint and  $(m - 1)$  possibilities for the second endpoint. For each circle, there are  $m$  possibilities for the center and an average of 10 possibilities for the radius.

Using the above formula, if an image has 10 points of interest, there are about 8 million permutations for a picture password of length 3. This is a significant decrease from the over 1 billion possibilities where gestures can be placed anywhere, when considering points of interest.

### **3.4.2 Security Analysis on PGA**

A paper submitted by Ziming Zhao et al. to the 2013 USENIX Security Symposium explores multiple vulnerabilities posed by PGA. They collected data to understand users' choices in picture selection, gesture location, gesture order, and gesture type. They then looked into the feature selection attack mentioned earlier by demonstrating how selection functions can be generated from sets of user gesture choices and how these selection functions can be integrated into a robust attacking framework [3].

The attack framework that they implemented was able to crack almost 50% of the passwords for previously unseen pictures in one dataset and 24% in another dataset with fewer than  $2^{19}$  guesses in a password space of  $2^{30.1}$ .

### **3.5 Computer Vision Algorithms**

Picture passwords are not immune to security threats, however. Since people naturally tend to select key regions or features of their image (to remember them easily), computer vision techniques such as object detection and feature detection can be used to indicate which features are the most prominent in an image and likely to be selected.

MATLAB, a high-level numerical computing environment, has a toolbox consisting of various computer vision algorithms. These algorithms often locate points or areas of interest using image contours and image intensity. It is important to note that feature extraction is very subjective in nature. There is no generic feature extraction scheme that works in all cases. MATLAB provides a variety of different corner and object detection algorithms, such as SURF features, MSER, and Canny edge detectors. We tested several in an attempt to suggest a 'strength' meter for picture passwords [4].

Speeded up Robust Features (SURF) detector is good for detecting blobs in an image and performs well even amidst object or image rotation and scaling. Maximally Stable Extremal Regions (MSER) detects more features and is more robust compared to SURF; regions are defined by the intensity function. Both these algorithms detect 'blobs' in images, but sometimes, it is more useful to detect the *edges* of objects within an image. MATLAB has an edge detector using the Canny method, which applies a Gaussian filter and finds the intensity gradients of an image to determine boundaries [5].

## **4 Methodology**

In an attempt to understand more about PGA and potential security issues, we conducted several user experiments, which are described below. In order to efficiently record gestures and tabulate results, we created our own simulation of the Windows 8 PGA framework.

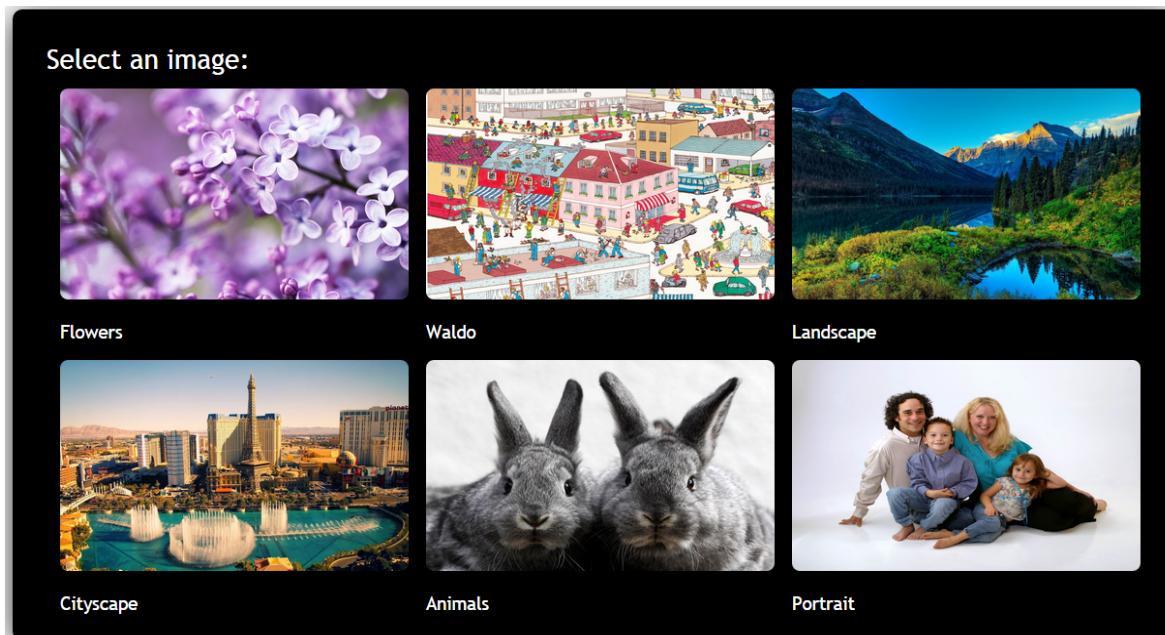
### **4.1 Experiment 1 - User Image Selection**

In this experiment, the goal was to determine the types of images users typically select when setting up Picture Gesture Authentication, and the reasoning behind their selections.

All participants were first informed of the definition and purpose of PGA. Then, users were asked to imagine setting up PGA on their own laptops, and were directed to select a background image from a set of six different images.

Our experiment was a slight modification of Zhao’s study [3], where all participants were asked to select their own image for use in Picture Gesture Authentication. In that study, all images chosen could be categorized into six classes: people, civilization, landscape, computer-generated images, animals, and others [3]. Therefore, for our experiment, we selected five representative images from these categories: “Family Portrait”, “Flowers”, “Cityscape”, “Landscape”, and “Animals”. We also selected a “Where’s Waldo” image because it is very complex and has many features.

All participants were given the choice of selecting one of these six images. Upon selecting an image, participants were instructed to draw and save three gestures as their password on our simulated Windows 8 PGA web framework. All of these gestures were recorded in our database. Circles were recorded with the coordinates of their center and radius, lines were recorded with the coordinates of the endpoints, and taps were stored as the coordinate of the point selected. Finally, participants were asked to complete a survey to indicate the reasoning behind their selections, their general opinion of PGA, and their concerns for picture password security. There were 37 participants overall in this experiment.



**Figure 1:** Participants were allowed to choose between these six images when “setting” up PGA.

## **4.2 Experiment 2 - Comparison of Gestures in Simple and Complex Images**

Upon completion of the first experiment, all participants proceeded to the second experiment; the primary aim of this part of the study was to determine how image complexity could potentially affect security. We sought to determine the correlation between the complexity of an image and the number/ types of features selected by participants. We selected the “Animals” image (bunnies) as a representation of a simple image because it contains relatively few features and the “Where’s Waldo” image as a representation of a complex image because it contains many people, buildings, cars, and other features within the image.

For this experiment, all participants were asked to draw three different gestures as their PGA for both the simple and complex image. As before, all coordinates of gestures were recorded in our database.

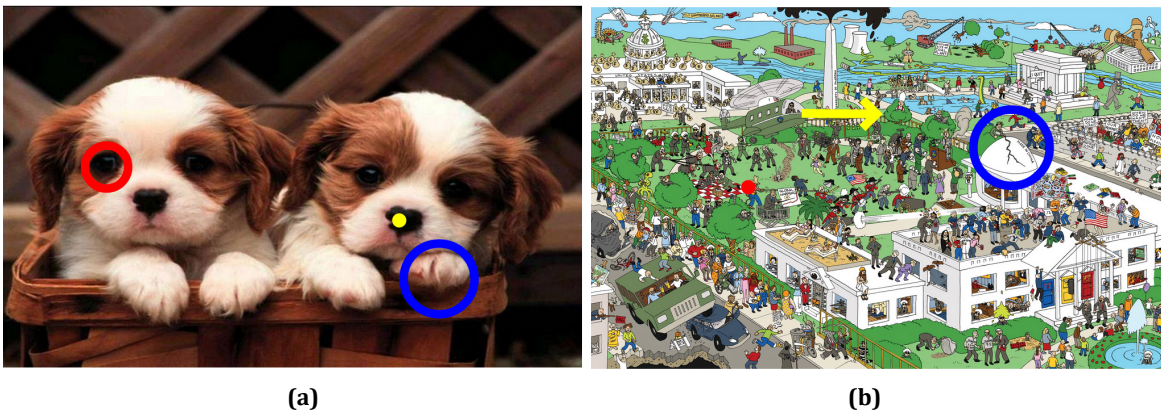
## **4.3 Experiment 3 - Shoulder Surfing**

The goal of our third and final user experiment was to determine the threats to security caused by shoulder surfing. Shoulder surfing occurs when an adversary observes a user entering his/her password and gets information.

### **4.3.1 Shoulder Surfing for PGA**

We wanted to compare the effects of *picture complexity* on the efficacy of shoulder surfing. To test the ease of shoulder surfing for PGA, we predetermined three gestures for both a simple image and a complex image: “Animals (Dogs)” and “Waldo (D.C.)”. We avoided using the same images from experiments 1 and 2 to ensure that users weren’t familiar with the images.

Each participant was first asked to observe us drawing gestures on one of these images and was later asked to enter their guess for our password. We wanted users to wait for a few minutes before entering their guess to better represent an actual case of shoulder surfing; thus, we had them complete another shoulder surfing test for text-based passwords during this time (described in 4.3.2). Overall, 7 users performed shoulder surfing on the Dogs image and 11 users performed shoulder surfing on the Where’s Waldo image.



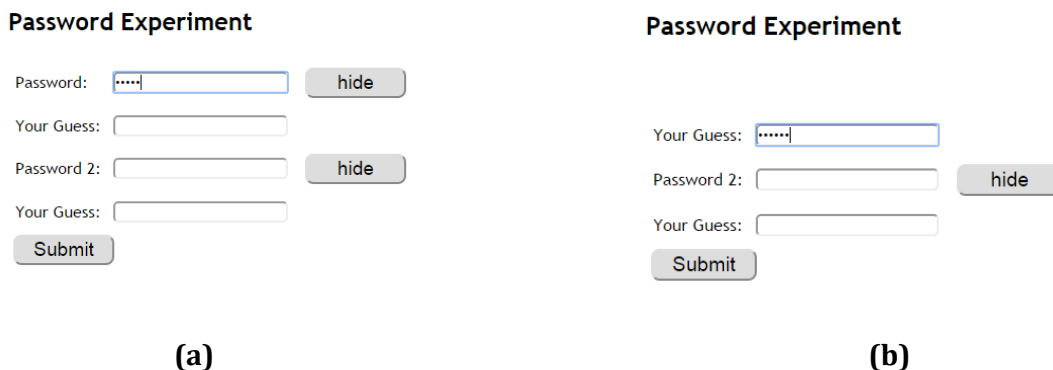
**Figure 2:** Our selected PGA passwords for (a) Animals (Dogs) image and (b) the Where’s Waldo (D.C.) image. The order of gestures is from left to right.

### 4.3.2 Shoulder Surfing for Text Passwords

Since people often compare PGA to traditional text-based passwords, we performed a similar experiment testing the ease of shoulder surfing for text passwords. We had users attempt to guess two different passwords: *Cat47* and *spRkle34*. These two passwords were selected randomly and had a good mixture of numbers, lowercase, and capital letters.

#### Experiment Protocol:

1. Participants were asked to select either “Animals (Dogs)” or “Waldo (D.C.)” image
2. They were instructed to shoulder-surf and remember our gestures as we drew them on the screen.
3. The screen was switched to the password experiment shown in Figure 3. We entered our first predetermined password, *Cat47* and hid our password to prevent users from determining the length of the password.
4. Participants were asked to enter their guess for the first password.
5. We entered our second predetermined password, *spRkle34* and hid the password.
6. Participants were asked to enter their guess for the second password.
7. The screen was switched back to the PGA experiment and participants were asked to enter their gestures for the picture password.



**Figure 3:** Interface for testing shoulder surfing on text passwords. (a) Depicts entering the first password (b) the first password field is hidden so participants cannot determine the length of the password.

## 5 Results

This section contains our results and analysis of our image selection, gesture selection, and shoulder surfing user experiments.

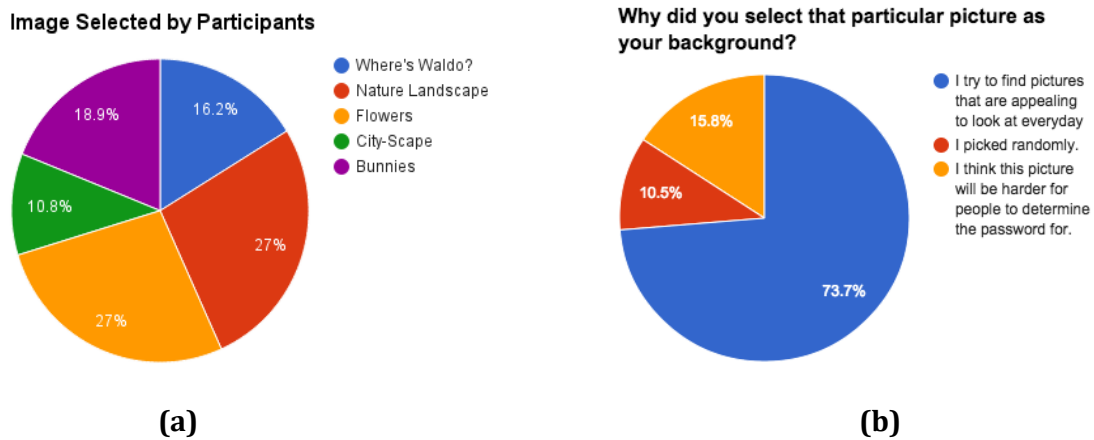
### 5.1 Experiment 1 - User Image Selection

The primary purpose of this part of the experiment was to determine which picture people typically preferred to use for their PGA, and to understand their reasoning behind selecting a particular background picture. We found that most people gravitated towards selecting the Nature Landscape or the Flowers image, but as Figure 4a) depicts, other images were selected by



a significant portion of participants as well. Notably, no one selected the Family Portrait image, likely because it was a family they didn't recognize or relate to. The actual gestures drawn for the images selected are analyzed more in Section 5.2 and the Appendix.

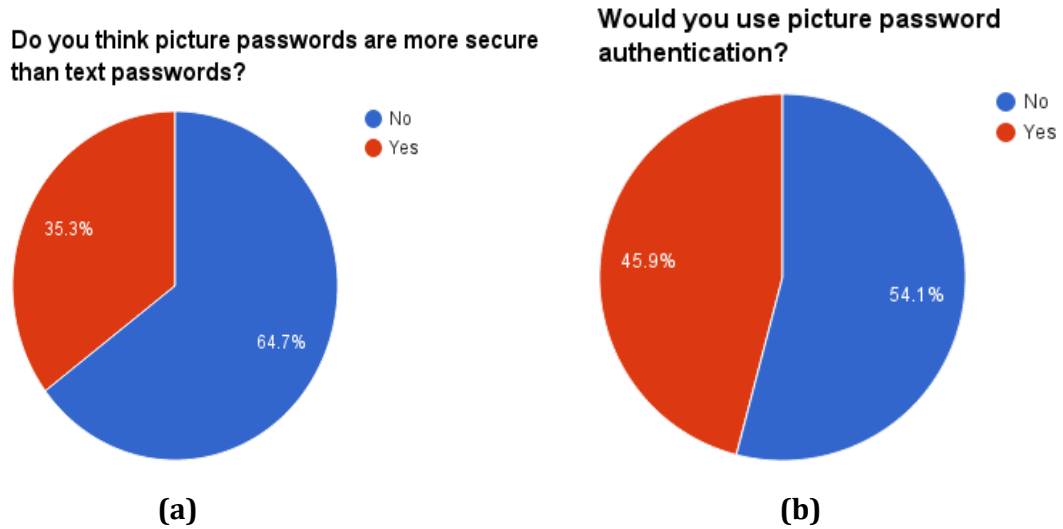
When we asked users to explain their reasoning behind selecting a particular picture, an overwhelming majority said they selected the picture since it was appealing to look at (Figure 4b)). This was understandable since users have to look at their image every time they log into their computers. On the other hand, only 15.8% thought about the complexity of the picture and its risks to security. This confirmed our hypothesis that people may select "less secure" picture passwords in favor of pictures that are more appealing to look at everyday.



**Figure 4:** Participants' answers to questions regarding their image selection (a) The image selected by users (b) The reasoning behind their selection

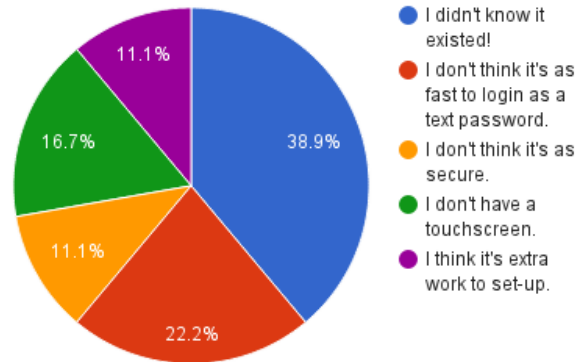
In addition, we also asked users about their opinion on using PGA to gauge current interest levels in the use of picture passwords in general. As Figure 5 depicts, a majority of people believed picture passwords would not be as secure as text passwords, but were still willing to use PGA. Of those using Windows 8 and not using PGA currently, almost 40% of participants remarked they had not even heard of PGA, indicating that changing the status quo of using text passwords is an obstacle to widespread adoption of PGA.





**Figure 5:** Participants' responses to questions regarding their opinion on PGA (a) The image selected by users (b) The reasoning behind their selection

**For those using Windows 8: Why do you not use picture passwords currently?**



**Figure 6:** Participants' responses to questions regarding their selection of gestures. A plurality remarked they were unaware PGA even existed.

## **5.2 Experiment 2 - Comparison of Gestures in Simple and Complex Images**

The primary goal of this experiment was to analyze the effect of picture complexity on the gestures selected. We asked users to draw gestures on a simple image with fewer features (bunny picture) and on a more complex image (Where's Waldo picture).

We captured all of our users' gestures and first qualitatively analyzed them by displaying each users' gestures and tracking the key features they selected. The tables below depict the percentage of user gestures that focused on various key features of the image. In general, we found significantly more variation in the features selected for the Waldo image in comparison to the bunny image.

Animals (Bunnies)	
feature	percentage
ears	28.07%
nose	20.18%
eyes	19.30%
space between bunnies	9.65%
space between ears	8.77%
face	7.02%
top of head	4.39%
other (body)	2.63%



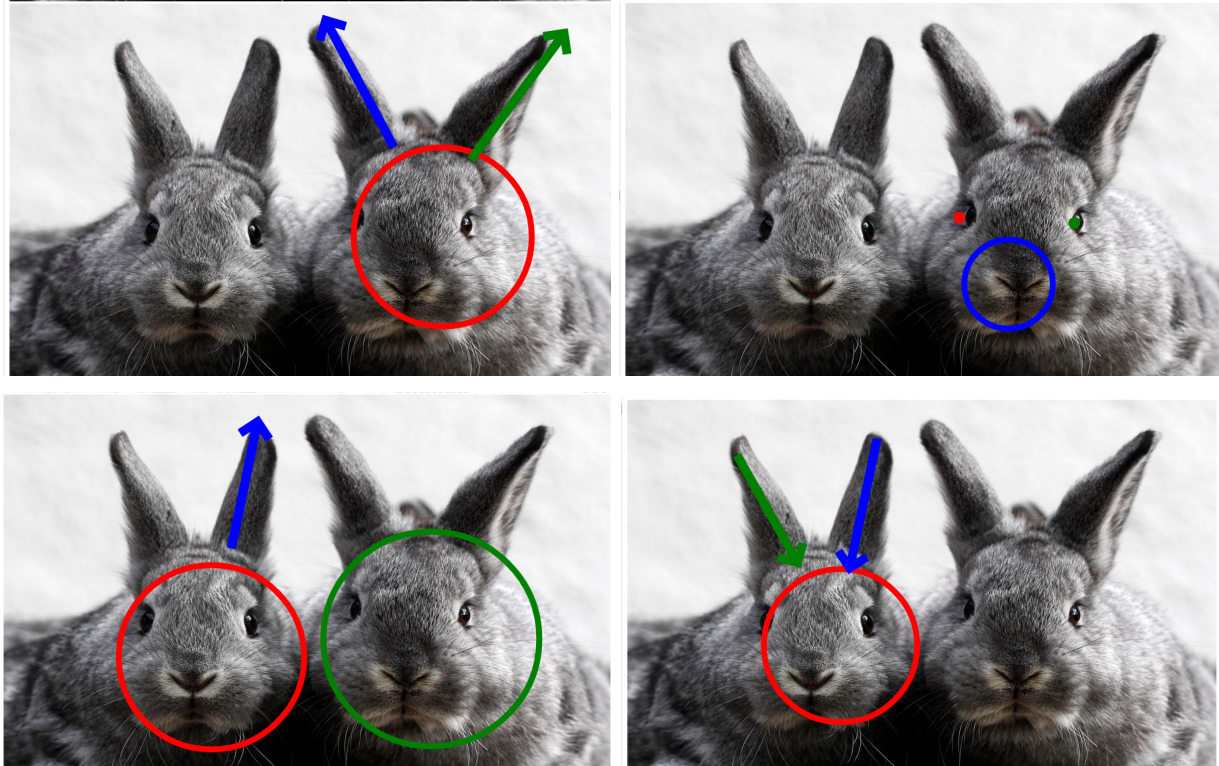
**Table 1:** Percentage of gestures selected on key features for the simple image. Most of the focus was on the ears, nose, and eyes.

Where's Waldo (town)	
features	percentage
<i>People (Combination)</i>	21.82%
Waldo!	8.18%
Guy on building	4.55%
Guy on fountain	2.73%
People on street	2.73%
People on roof	1.82%
Firefighter	1.82%
Cars	19.09%
Line across building	18.18%
Ladders	7.46%
Street	6.36%
Fountain	6.36%
Building	5.45%
Corner of building	3.64%
Awning	3.64%
Other	3.64%
Clock	2.73%



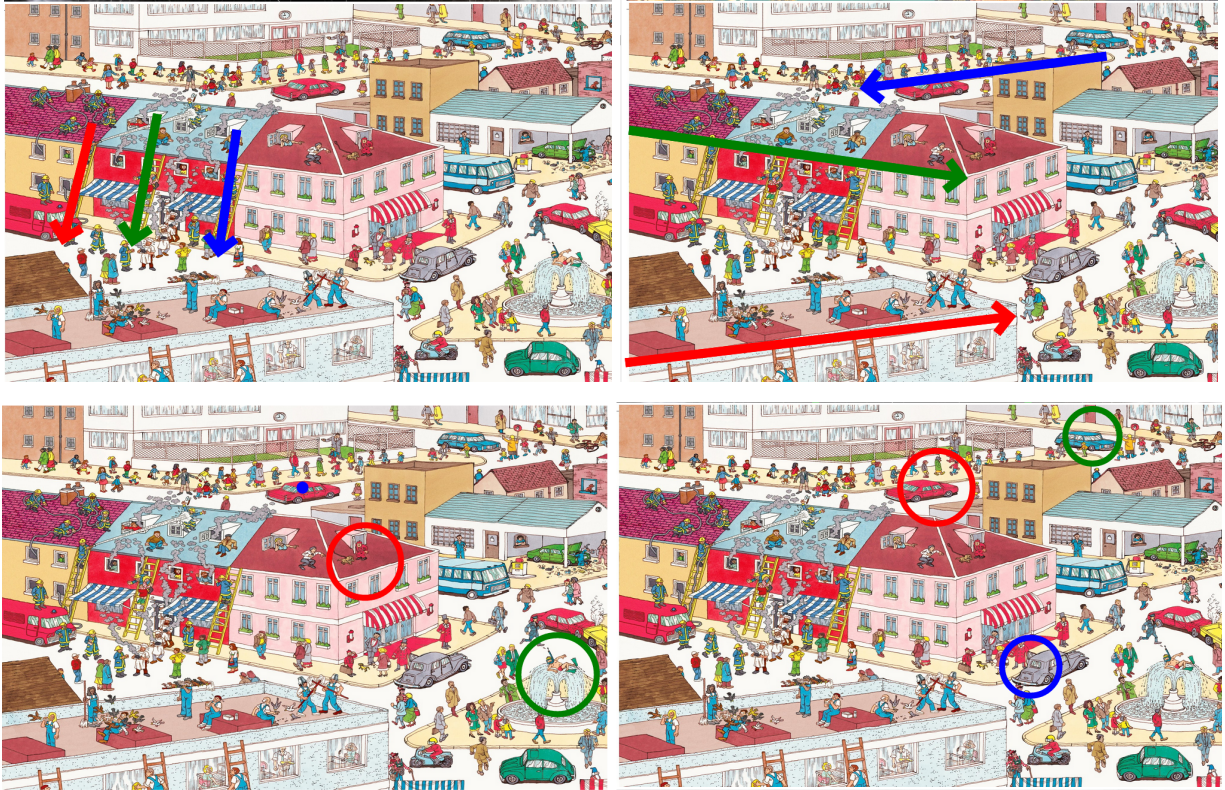
**Table 2:** Percentage of gestures selected on key features for the complex waldo image. There were many features selected, with no particular feature standing out amongst the mix.

We also performed a qualitative analysis of the gestures drawn by participants on these two images. As Figure 7 depicts, many users drew similar gestures on the bunny image, focusing on the face, ears, or eyes. In contrast, gestures covered many diverse points of interest on the Waldo image.

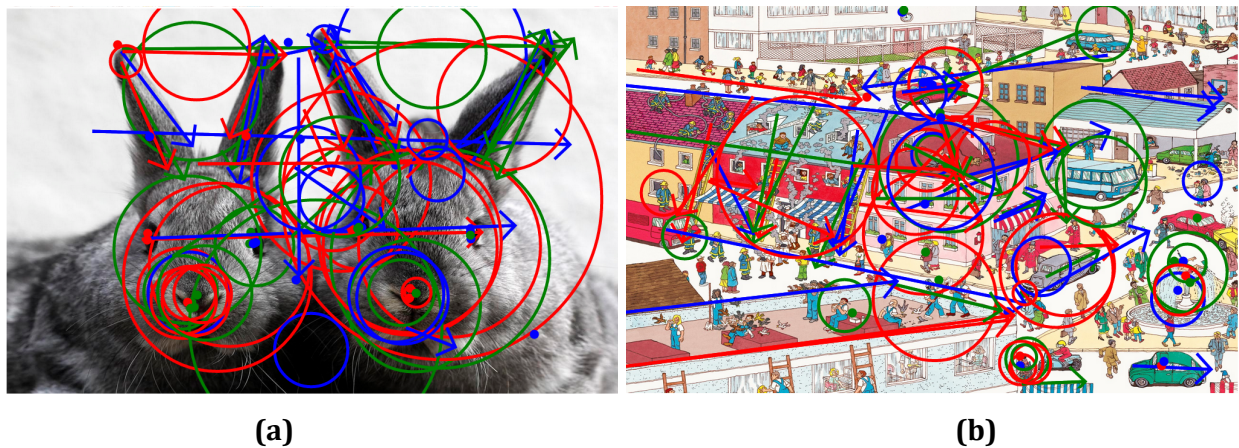


**Figure 7:** The four images depict the gestures drawn by four different users on a *simple* image. The coloring of the gestures represents the order in which they were selected.





**Figure 8:** The four images depict the gestures drawn by four different users on a *complex* image. Coloring of the gestures represents order in which they were selected.



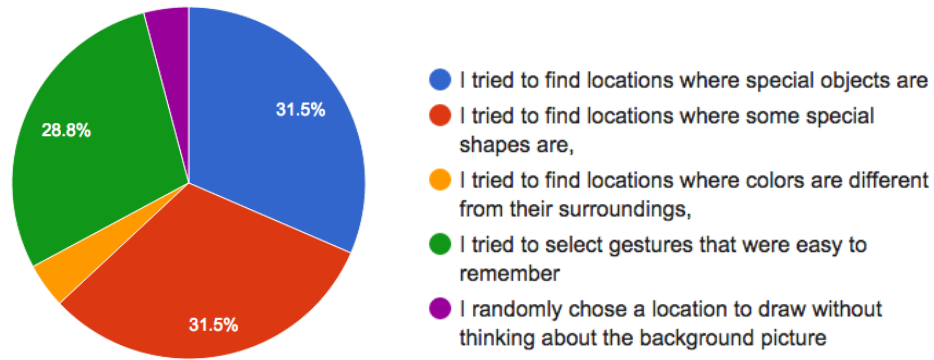
**Figure 9:** Compilation of all gestures drawn by participants in this study on: (a) simple bunny image (b) more complex waldo image

Figure 9 displays an overlay of all the user gestures, to display the general areas of the image that were targeted. Gestures for the bunny were centered around the face and the ears, but some users went against the grain and opted to select white spaces between the ears and face. These were often the users who tried to select locations that would be more secure. Gestures on Waldo were far more varied however, and spanned the entirety of the picture. After completing the



experiment, we asked users to explain their reasoning behind their gestures, and found the majority of them selected gestures based on special objects and shapes (Figure 10). This was one of the factors that led us to use feature detection algorithms on our images as part of our proposed enhancement for PGA, discussed more in Section 6.

**How did you select your gestures?**

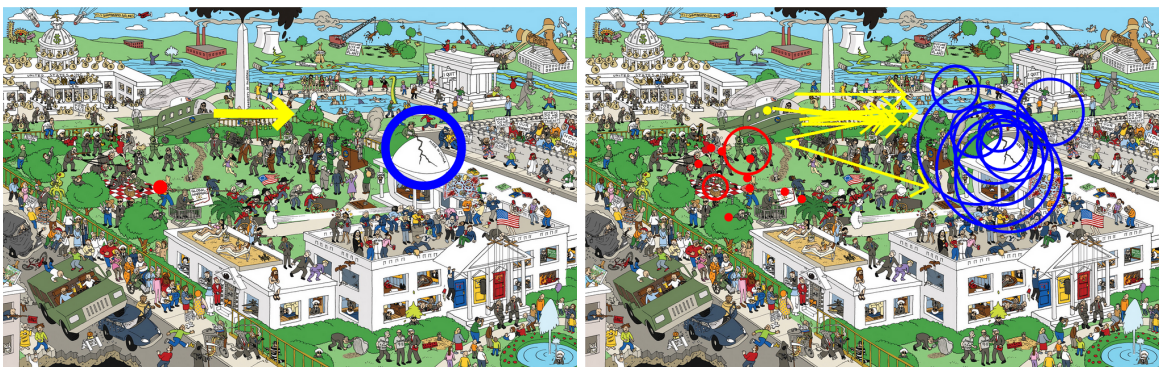


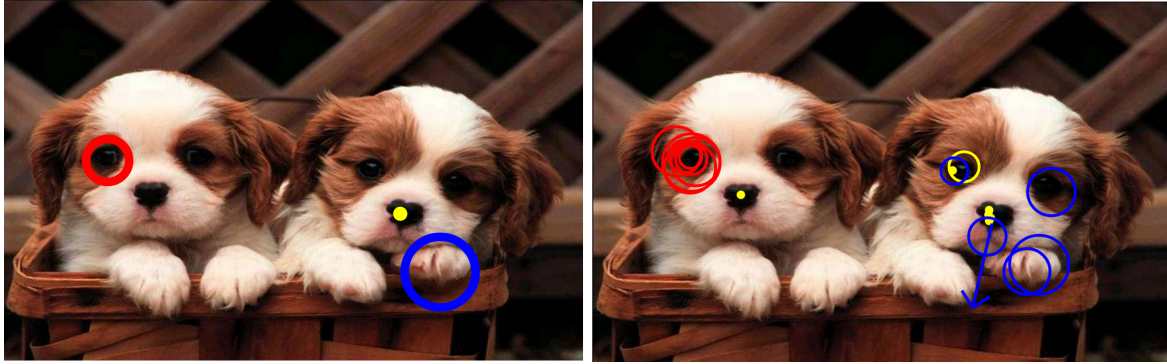
**Figure 10:** Participants’ responses to questions regarding their selection of gestures on the images. Most people focused on locations with either special objects or shapes.

**5.3 Experiment 3 - Shoulder Surfing**

**5.3.1 Shoulder Surfing for PGA**

The primary purpose of our last user experiment was to analyze the effects of picture complexity on the ease of shoulder surfing. Figure 11 shows the correct gestures along with a compilation of all users’ guesses. The vast majority of participants correctly guessed all three gestures types (circle, tap, line). They also selected the correct region of the images, but did not get the specific position correct. For example, in the Where’s Waldo image, almost all users knew that the first gesture was in the group of people on the left, but only two participants correctly selected the man holding a rifle.





**Figure 11:** The images on the left depict the actual gesture password we drew as our predetermined PGA password, while the images on the right represent the compilation of all the gestures users drew in the study. The red, yellow, and blue drawings on the images represent the first, second, and third gestures, respectively, drawn by participants.

We performed a computational analysis to determine the correctness of users' guesses by applying a strategy similar to that used by Windows [2]. The longer dimension of the image was divided into 100 segments, and the shorter dimension was then divided on that scale to create square grids. Since the longer dimension of our image was 1020 pixels, the resulting grids had a side length of 10.2 pixels. In order for a tap to be considered a match, it had to be within 3 times the grid-size from the correct grid. For a line gesture to be correct, both the starting and ending points had to match. We came up with our own metric to score circle gestures because we were unable to find sufficient Windows documentation for circle gestures. Similar to the tap gesture, both the center and radius error tolerance for circles were 3 times the grid-size.

Table 3 shows for each gesture, the percentage of users who were able to guess the gesture type correctly and the percentage of guesses that would be marked as correct (i.e. accepted by PGA) by the metrics mentioned above. In general, users were able to remember the gesture *type* correctly regardless of picture complexity, but had significantly more difficulties pinpointing the exact gesture location.

In addition, for the complex image (Where's Waldo D.C.), a lower percentage of users were able to repeat the gesture correctly compared to those who shoulder surfed on the simpler image of Animals (Dogs). 28% of users got all three gestures correct for the simple image, while no one got all three gestures correct for the complex image. In fact, no one correctly guessed more than one gesture. These results demonstrate that a image complexity can affect security, and using a complex image can mitigate the risks of shoulder surfing.

### Animals (Dogs)

Gesture #	correct gesture type	correct gesture
1- circle	100%	86%
2- tap	86%	57%
3- circle	71%	28%

### Where's Waldo(D.C.)

Gesture #	correct gesture type	correct gesture
1- tap	82%	18%
2- line	82%	9%
3- circle	100%	36%

**Table 3:** Analysis of Shoulder Surfing Results. Displays the percentage of users who guessed each gesture type correctly along with the percentage who guessed the gesture location correctly.

### 5.3.2 Shoulder Surfing for Text Passwords

In addition to performing a test on shoulder surfing for complex vs simple images, we also tested the effects of shoulder surfing on traditional text passwords. We used Python's difflib library to quantify the similarity between the original password and the guessed password by each user. The difflib library's ratio method returns a measure of the similarity between sequences as a float value between 0 and 1. A ratio of 1 represents a perfect match and 0 represents no similarity [6].

For the first password, *Cat47*, the average ratio was 0.291. For the second password, *spRkle34*, the average ratio was 0.386.

The difflib library calculates the similarity ratio between two strings using the formula  $2 \cdot M/T$ , where  $M$  is the number of matches and  $T$  is the total number of elements in both sequences. To be counted as a match, the characters must be in the longest common subsequence and must be in the correct relative order. For example, if the original password is "abc", a guess of "acb" would count as two matches and a guess of "cba" would count as one match. [6]

Surprisingly, the guesses were more accurate for the second, longer password than for the first password. One factor that may have contributed to this result is the fact that one error in a 5 character password has a greater impact than one error in an 8 character password since the password lengths are taken into account. In addition, from looking at the data, more participants correctly guessed the last numbers (34) for the second password than for the first password (47). This may have been because participants were able to better anticipate our typing patterns. Therefore, this part of the experiment could be improved if we test the two passwords separately on different users.

### 5.3.3 Comparison of Shoulder Surfing on PGA to Text Passwords

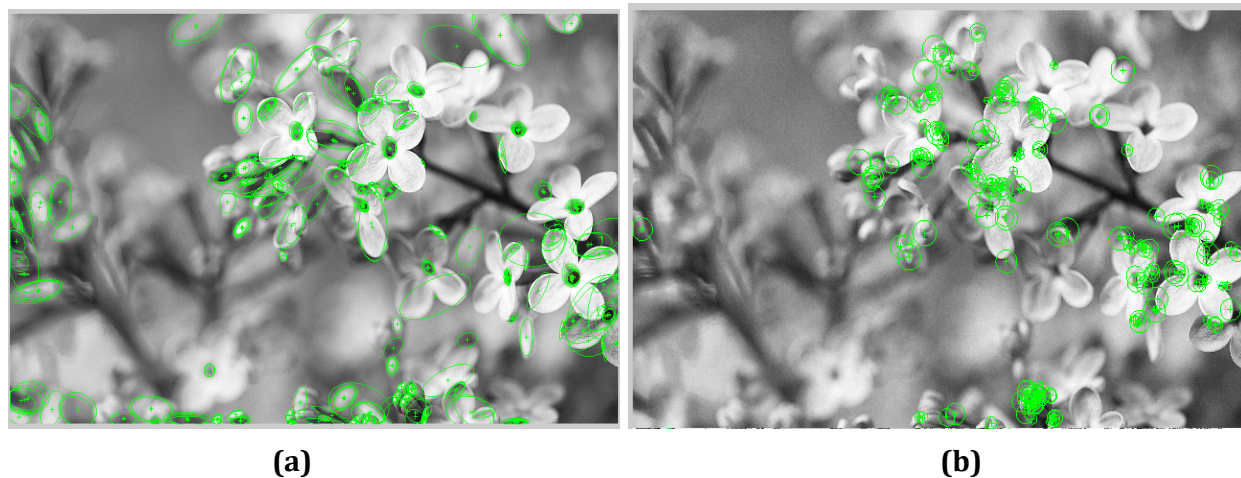
Although it is difficult to directly compare the results of shoulder surfing on PGA to traditional text passwords, it is interesting to note that 28% of users were able to correctly guess all three gestures for the simple image (Dogs), while no users were able to guess either the 5 character or 8 character password. The complex image fared better than the simple image, since no users were able to guess all three gestures correctly, but many still identified the gesture types properly and were in the correct general region of gestures. People often reported feeling overwhelmed when trying to guess text passwords, but felt significantly more comfortable and confident when trying to shoulder-surf for PGA. Thus, a more complex image can provide more security, but may still not be comparable to a traditional text-based password.

## 6 Enhancements to PGA

### 6.1 Feature Detection

Noting from our previous experiments that picture complexity does appear to affect security, we attempted to use computer vision algorithms that would identify prominent features in an image to determine whether a selected picture had sufficient complexity for PGA. However, feature detection is not an exact science. We tested several corner and object detection algorithms from MATLAB on our images, in an attempt to find an ideal algorithm to serve as a 'strength' meter for picture passwords.

The figures below show the features detected when running the MSER and SURF feature detection algorithms on a couple of the images we tested.



**Figure 12:** Features detected in the flower image are highlighted in green. (a) MSER regions (b) SURF regions





**Figure 13:** Features detected in the Waldo image are highlighted in green. (a) MSER regions (b) SURF regions.

In general, the MSER algorithm not only detected more features than SURF, it also was more accurate in outlining the borders of objects in images. SURF seemed more random in its feature selections, missing many points of interest in the Waldo image and branches in the flower image (Figures 12 and 13). We also tested the Canny method which found boundaries of all objects (not shown here), but failed to target key elements of interest.

Thus, we opted to select the MSER feature detection algorithm in MATLAB as our primary metric for developing a strength meter for picture passwords. MSER has a parameter that can display the number of regions, so we used these values to quantify the number of regions detected. As we would suspect, both Waldo images had the highest number of regions, while the animal pictures had significantly less.

Image	# of features detected
Waldo (town)	7809
Waldo (D.C.)	7412
Landscape	3496
City	1539
Bunnies	356
Flowers	338
Family	313
Dogs	79

**Table 4:** Number of MSER regions detected for all images tested in our study sorted from highest to lowest. Entries shown in green are categorized as strong by the prototype; an orange background corresponds to a moderately secure image, and a red background suggests that the image would result in weak passwords.

## 6.2 Strength Meter Prototype

We used the MSER metric as our basis for designing a PGA password strength meter similar to the ones currently available for alphanumerical passwords. We created the prototype as a web application that allows users to upload an image, runs the feature selection algorithm on the image, and then displays the results via a strength meter, as shown in Figure 14.

### Select an image you would like to upload.

C:\Users\Denise\Docume Browse... Upload



**Figure 14:** Screenshot of our security enhancement prototype. The image of the dogs is indicated as a weak password.

The percentage displayed by the strength meter depends on the MSER metric for the image. The images are categorized as weak, moderate, and strong, each corresponding to a meter color of red, orange, or green. The cut-offs were determined by examining the MSER values in Table 4. We set the cutoffs such that images with complexity comparable to that of the Where's Waldo image (MSER metric  $> 3500$ ) are identified as strong; images with features comparable to that of the landscape image ( $360 < \text{MSER metric} < 3500$ ) are characterized as moderately secure; lastly, images with features comparable to that of the animals images (MSER metric  $< 360$ ) are considered weak.

When creating the prototype, we discovered that MATLAB was not the most ideal feature selection package to use. In order to allow MATLAB code to run on a system without an installed copy of MATLAB, the code needs to be compiled as an operating system dependent standalone executable. The user then has to download a copy of MATLAB Runtime (MCR), which is on the order of 1.5 GB, in order to run the standalone executable. Since the executable can't return a value, we pushed the number of features determined by the feature detection algorithm to Firebase using a python script run within the MATLAB code. In order to run the .exe file from a browser on Windows, we created an ActiveXObject object, which is currently only supported by

Internet Explorer due to security reasons. In the future, we would like to test other feature detection packages (i.e. openCV) that are often more compatible with different operating systems and browsers.

Incorporating this prototype into the current PGA framework would require users to select images with greater complexity. As shown in our user experiments, complex images allow for more diversity in gesture selection. This leads to greater security by increasing the difficulty of guessing the correct password using a feature selection attack.

## 7 Limitations and Future Work

Some of our results may have been skewed due to the small number of users we tested. Ideally, we would have liked to conduct experiments on ~100 users, which would have increased the accuracy of our analyses for image and gesture selections. In addition, the accuracy of our strength meter is dependent on MATLAB's feature detection algorithms. Our cutoffs for password strength levels could also be finetuned with a larger set of images and more users.

Future work includes incorporating a more accurate feature detection algorithm that is more compatible with commonly used operating systems and browsers.

Currently, our password strength calculation is based on the assumption that all features in an image are selected uniformly; however, even if a picture is very complex, it may contain several distinctive features that almost all users will gravitate to. Therefore, we could also modify the feature detection algorithm to detect these extreme features and assign a lower password strength level in these cases.

Additionally, our strength meter is only based on the number of *features* found in the image. However, the gestures selected by the user can also greatly impact password security. For example, a password with three taps on the same object is much less secure than a password including a combination of gesture types in more diverse locations. In the future, we could create a more robust strength meter that not only analyzes the image but also the particular gestures a user plans to use for their PGA.

Another possible extension is to improve the security of password storage and retrieval. The Windows Engineering Team did not describe how each gesture is stored or encrypted, so in the future we could incorporate additional hashing and encryption methods to prevent adversaries from learning password information even if they gained access to the database.

## 8 Conclusion

First we conducted user experiments to better understand user preferences in image and gesture selection. The results indicated that users tend to choose images that they find appealing rather than considering the security of the background image. Our final user experiment demonstrated that picture based passwords are more vulnerable to shoulder surfing than traditional text based

passwords. In addition, we tested the effects of picture complexity on shoulder surfing, and discovered that more complex pictures are harder to crack than pictures with fewer features. These findings led to the conclusion that a mechanism to predict the estimated password strength of an image is necessary to ensure sufficient password security. We used a feature detection package from MATLAB to determine the complexity of an image, and displayed that to the user through a password security strength meter.

## 9 References

[1] Xiaoyuan Suo, Ying Zhu, G. Scott. Owen. (2005). *Graphical Passwords: A Survey*. Paper presented at Annual Computer Security Applications Conference: ACSAC, Los Angeles. Georgia State University.

[2] Sinofsky, Steven. "Signing in with a picture password." Blog. *Building Windows 8*. 16 Dec. 2011. Web. 1 Jan.

<<http://blogs.msdn.com/b/b8/archive/2011/12/16/signing-in-with-a-picture-password.aspx?>>.

[3] Zhao, Ziming, et al. "On the Security of Picture Gesture Authentication." *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)* (2013):

<[http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/12316-sec13-paper\\_zhao.pdf](http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/12316-sec13-paper_zhao.pdf)>.

[4] Carmichael, G.; Laganière, R.; Bose, P., "Global Context Descriptors for SURF and MSER Feature Descriptors," *Computer and Robot Vision (CRV), 2010 Canadian Conference on*, vol., no., pp.309,316, May 31 2010-June 2 2010

[5] Mathworks. (2011). *Point Feature Types* (r2015a). Retrieved May 10, 2015 from <http://www.mathworks.com/help/vision/ug/point-feature-types.html>

[6] Python Software Foundation. The Python Standard Library, version 2.7.10rc. Available at <https://docs.python.org/2/library/difflib.html>

## 10 Appendix

### 10.1 Implementation Code


All code written for this project can be found here:

<https://github.com/cdenise/6.857-PGA-Project>

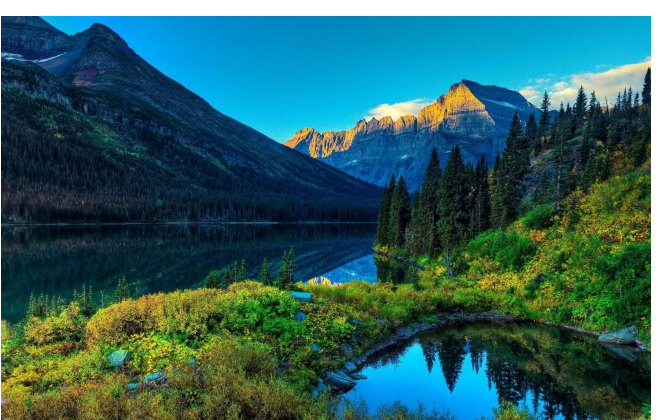
### 10.2 Additional Feature Tables

As mentioned in Section 5.1, the tables below show the common features selected by users in the Cityscape, Landscape, and Flowers images.

City (4 users)	
feature	percentage
Globe	33.33%
Fountains	33.33%
Eiffel tower	25.00%
Sky	8.33%



Landscape (10 users)	
feature	percentage
Lines of sides of mountains	36.67%
Top of mountain	16.67%
Rock in field	13.33%
Top of tree in pond reflection	6.67%
Top of tree	6.67%
Between mountains	6.67%
Tree in reflection	6.67%
Edge of mountain/river	3.33%





Little pond	3.33%	
-------------	-------	--

Flowers (10 users)	
feature	percentage
Left flower - center	14.71%
Middle flower - center	14.71%
Left flower	11.76%
Other	11.76%
Other flowers	11.76%
Right flower - center	8.82%
Middle flower	5.88%
Space between flowers	5.88%
Other flowers - center	5.88%
Branches	5.88%
Buds	5.88%

