

# UGIS 188: Inquiry 4 Proposal

Brandon Chinn, November 13 2016

## Introduction

Every person who owns a computer has had experience with creating and remembering passwords. Whether it's the password to log on to the computer or the password to their email account, today's society relies heavily on passwords. However, despite the importance of passwords in keeping confidential information private, people often choose poor passwords, such as "password", "123456", or "abc123" (Doel, 2012). Even requiring passwords to be of a certain format does not always create more secure passwords. Kelley et al. (2012) did a study where participants were asked to create a password in a certain format. For example, one participant might be asked to create a password of at least 16 characters (with no other requirements), while another participant might be asked to create a password of at least 8 characters without containing a dictionary word. Their study found that longer passwords were actually more secure than passwords with requirements (e.g. one number, one upper case letter, and no dictionary words), undermining the theory behind modern websites' password requirement schemes.

There has been some developing research in the area of image-based password schemes, where users supplement a text-based password with a series of pictorial challenges that is resistant to current password hacking techniques. J. Blocki et al. (2013) wrote a paper formalizing such a scheme, where, after the user creates a username and password, the website shows the user a set of images, randomly generated from the password. The user then writes labels for each image and sends the labels to the server. The server will then store the user's username, password (encrypted), and the labels. To log back in, the user will give their username and password, then the server will re-generate the random images. If the user gives the same password, the images would be exactly the same as before; otherwise, the images will look different (but the server still returns the images!). The server will also send back the labels, and the user will have to match the label they wrote to the corresponding image.

J. Blocki et al. did include a user study in their report, but the user study focused on the amount of time it took users to create an account and log back in. This study seeks to test the security behind the scheme, testing to verify that it would, in fact, be infeasible for an attacker to crack passwords using this scheme.

## Research Questions

- Does the accuracy of responses go down as the number of images in the password goes up?
- What are the best values to use for the scheme's security parameters in order to optimize user experience while maintaining security?

Regarding the second question, there are three security parameters that can be customized for higher security guarantees, but usually at a cost to user experience and usability. These parameters are:

- **Number of images:** as the number of images goes up (and the number of labels the user makes for each image), the security goes up in a similar fashion to a longer password. However, it might be harder for a user to remember all the labels for the images, and it would be impractical for a user to spend more than a couple minutes just to log in.

- **Accuracy threshold:** the server would typically allow some percentage of images to be wrong for the user to still log in. A lower threshold means more security (the user would have to be closer to 100% accurate) but might generate more false negatives (the correct user is locked out). A higher threshold means less security (an attacker has more opportunities to guess the right labels), so it might generate false positives. In addition, a higher threshold would take longer to authenticate, since the server has to enumerate every possible combination of labels, and there are more combinations when more labels are allowed to be wrong.
- **Hash iterations:** servers typically store hashed passwords in their databases rather than the actual password. A hash function is a function that always returns the same thing for the same password, but it's really hard to get back to the original password from the hash. In order to deter attackers, passwords usually go through multiple iterations of hashing so that it takes longer for an attacker to check each password (usually with negligible delay to the user). However, this scheme requires passwords be hashed multiple times to verify that the password is correct, so it might not be as negligible to the user as with usual text-based password schemes.

## Method

To test these research questions, the researcher built a website that allows users to create an account and log back in using the scheme outlined in the paper written by J. Blocki et al. The code used for the website can be found at <http://github.com/brandonchinn178/gotcha-password>. Creating an account is spread out into three pages. The first page asks for a username and password, along with a disclaimer as to the security risks inherent in the study. The second page generates images based on the password and asks the user to create a caption for each image. The third page is a confirmation page that informs the user that they have successfully created an account and that they should come back the next day to log back in. Screenshots can be found in the Appendix.

When a user creates an account, the server will randomly set the number of images they'll be labeling. Each image will be generated by creating an SVG containing 30 circles, 20 smaller circles, and 30 ovals, all of random colors and random coordinates. Then each ellipse will be mirrored on the other side of the image (symmetrical across the y-axis). After the user successfully completes all the steps, the server will store the following in the database: username, the hashed password, a random string of text, the number of images the user labeled, and the randomized order of the labels. The order of the labels typically wouldn't be stored on an actual server, but this study will need the order for efficient data analysis.

Then the user can also log back into the website by providing their username and password. The website will then take them to another screen with the same three images, if they provided the same password. If they gave a different password, the images will look different, and will be an indication to the user that they typed the wrong password. For each image, the user now has to choose the right label for each image. Each image has a list of all the labels, in the same randomized order saved earlier. After matching a label for each image, the website will show a screen informing the user the number of labels they correctly paired with its image. Additionally, the server will store the following information for each login attempt: the user who logged in, whether they gave the right password or not, the number of correct labels they matched, the password they used to log in, and the order of labels they gave in this attempt. The passwords are

encrypted with a fixed encryption key before being stored in the database as an extra security measure against people who might gain unauthorized access to the database.

## Data Analysis

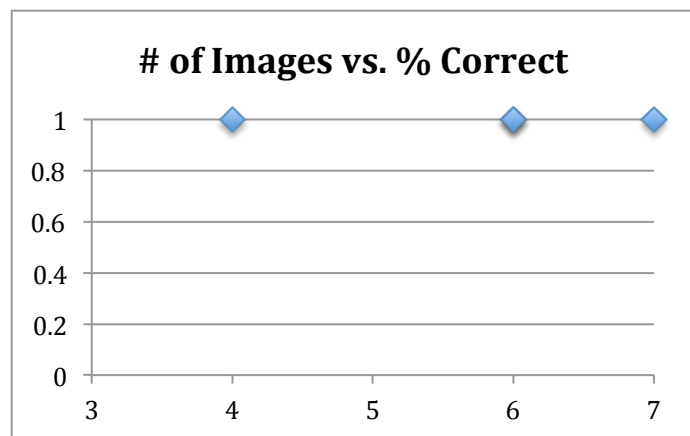
After a sufficient number of users create accounts and attempt to log in at least once, data from every login attempt will be compiled and analyzed to study various relationships in the data. The first relationship that will be analyzed is plotting the number of images a user had to match against the percentage of images the user correctly guessed. After plotting the data, the researcher will find the regression line and the correlation coefficient ( $R^2$  value).

Then, to study the relationship between security parameters and user experience, the researcher will run an algorithm that would run every time the user logged on using the saved data from users who have logged in for the study. The algorithm will be timed, first adjusting the accuracy threshold and holding the iterations constant at 1 iterations (for reference, this is normally set to 24000 iterations), then adjusting the number of iterations and holding the accuracy threshold constant at 2 (at most 2 labels can be mismatched). Each situation will be plotted on a graph, distinguishing plots between users with different number of images. After plotting each graph, the researcher will find the regression lines and correlation coefficients for each graph, one for each number of images.

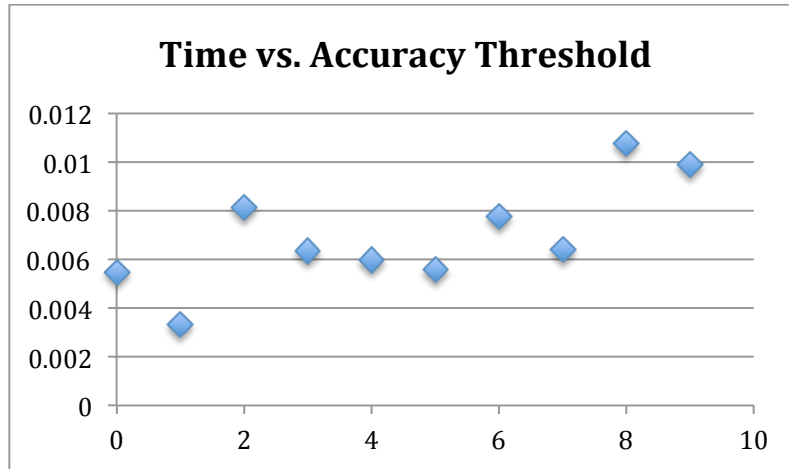
Results from the pilot study can be seen below. Note: numbers for the percentage of correct images don't include people who logged in with the wrong password.

Total accounts created	9
Total login attempts with invalid passwords	5
Min percentage of correct images	0
Max percentage of correct images	1
Average percentage of correct images	44.44%
Median percentage of correct images	100%

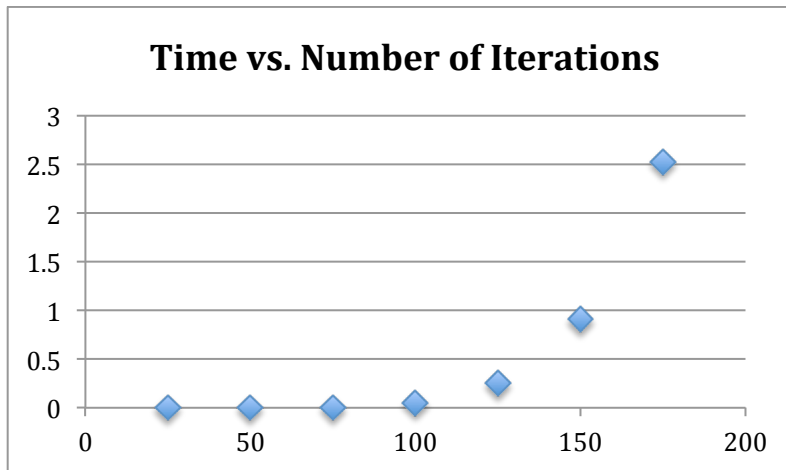
The plot of number of images against percentage of correct images can be seen to the right. This plot does not include login attempts with the wrong password. The plot looks really sparse because everyone who logged in with the right password got 100% of the images correct. Some of the people who did the pilot logged in soon after making the account since it was just the pilot, but in the actual research, people should wait a full day before logging back in.



Due to time limitations in running the pilot, the following is a plot for just one person, timing how long it takes to check the password they logged in with against the password they created the account with, holding iterations constant and adjusting the accuracy threshold. With more data, everyone with the same number of images will be plotted in the same color.



The following is a plot for the same person, measuring the same thing except holding the accuracy threshold constant and adjusting the number of iterations.



## References

J. Blocki et al. (2013). *GOTCHA password hackers!* Retrieved from <http://dl.acm.org/citation.cfm?id=2517319>

K. Doel (2012). *Scary logins: Worst passwords of 2012 and how to fix them.* Retrieved from <http://www.prweb.com/releases/2012/10/prweb10046001.htm>

P. G. Kelley et al. (2012). *Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking* Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6234434&isnumber=6234400>

## Appendix

### A STUDY ON IMAGE-BASED PASSWORD SCHEMES

UC BERKELEY: UGIS 188, FALL 2016  
BRANDON CHINN

#### Create an account

Username:   
Required. 30 characters or fewer. Letters and digits only.

Password:

Verify password:

#### Disclaimer

Usually, passwords are practically impossible to decode when stored on a server. However, I will be storing an encrypted version of your password every time you log back in, for the purpose of data analysis (methodology and plan for analysis can be found [here](#)). The encryption should also be secure (full specs [here](#)), but if someone manages to steal the encryption key (highly unlikely), your password could be vulnerable. Since I do have access to the encryption key, I have the power to decode your passwords, but I promise that I won't directly see your passwords. If you have any questions, please contact me at [brandonchinn178@gmail.com](mailto:brandonchinn178@gmail.com).

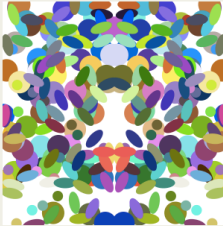
I consent to participating in this study and understand the associated risks: ☐


Favicon by Designmodo, licensed under CC BY 3.0. Study based on research done by J. Blockl et al. (2013). Full research proposal [here](#).

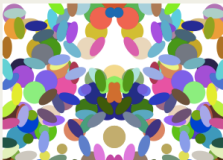
Figure 1: Creating an account with the disclaimer

#### Create an account

For each image below, provide a descriptive label that represents that image. The label should be something you'd recognize later for the same image.








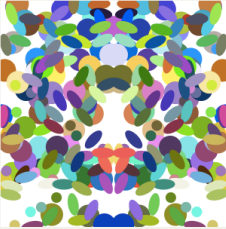


Figure 2: Labeling images when creating an account

### Login

Select the label you wrote for each image. If you don't recognize any of the images, it might be because you provided the wrong password.




skeleton

ice cream scoops

lion

flower



skeleton

ice cream scoops

lion

flower

**Figure 3: Logging back in, selecting labels for each image**