An Evaluation and Implementation of Passcode Strength Analysis on Arbitrarily-Sized Grid-Based Graphical Passwords

Riley James
30079140
Computer Science

Group 7

Christopher Kozbial
30061532
Computer Science

Lauren Mayes
30062361
Computer Science & Business

Grace Kelly Osena
30074352
Computer Science

ABSTRACT

Traditional Graphical passwords tend to focus on either small grid-based systems or image recognition/differentiation. For grid-based passwords, much work has been done to analyze different types of strength on 3x3 and 4x4 grids, but little work has been done to generalize these conceptions of strength to larger and non-square grids. This paper aims to analyze existing metrics against larger grids and discusses if these are still meaningful measures.

1 INTRODUCTION

Graphical passwords refer to the use of images or visual patterns to authenticate users, as opposed to alphanumeric strings. They seek to take advantage of the relative ease with which humans can memorize visual patterns and stimuli compared to alphanumeric sequences, for a user to recall their pattern while maintaining complexity reliably. One of the most popular forms of graphical passwords is Android's 3x3 pattern lock system, wherein the user must draw a pattern. To authenticate the user, the input pattern must match the pattern initially set. Android made this their default authentication measure for unlocking their phones, resulting in the popularity of the graphical password system.

Similarly, Microsoft has implemented a picture password in Windows 8 and beyond, allowing users to set secret motions on top of a background image to authenticate themselves. In this system, users can set any image as the background, then they do a series of gestures such as clicking and swiping that constitute the password.

Easy memorization, one of the strengths of graphical passwords, makes them vulnerable to "shoulder-surfing attacks" and "smudge attacks". Shoulder-surfing attacks occur when an unauthorized user observes an authorized user during the authentication process, ie. looking over their shoulder while the user draws their pattern. Because graphical passwords are easier for the human brain to recall, it is much more likely that an attacker will be able to reproduce a pattern after observing it being input into the device. This method is also effective against numerical passwords where the keypad is designed in a grid manner, like a phone pad layout. Smudge attacks are remnant analysis-style attacks where the attacker can analyze the greasy smudges left behind by a user's fingers on the screen of a device to determine the pattern. This is easier with graphical passwords because they leave a continuous path as a remnant, as opposed to the discreet marks left behind by pins or passwords, thereby making it much easier to reconstruct the pattern.

Popular implementations of grid-based graphical passwords (such as phone lock screens) are limited to a 3x3 grid. However, as is discussed below, this grid size imposes heavy limitations on the security strength of passwords. Further, many methods and tools have been developed to help users understand the strength of their 3x3 passwords, but few tools have been explored for larger grids. Our aim is to analyze their strength in larger grids and propose a method for conveying the strength of such passwords. We intend to test three existing strength equations as well as propose our own and visualize this to the user through a real-time meter. We will be comparing the Song, Sun, Andriotis, and Heidt-Aviv meters against our meter.

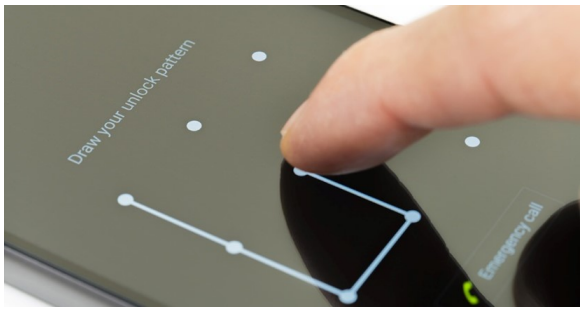

Figure 1: Android Lock Screen

We have designed a system that can dynamically change the size of the input grid, which allows us to see how the strength meters change as the size of the grid differs. This system also allows for non-square grids which will change the accuracy of some meters as they are based on square grids.

2 RELATED WORKS

2.1 Graphical Passwords: Learning from the First Twelve Years

[Biddle et al. 2012]
Biddle et al. outlines several types of graphical password schemes that can be used as an alternative to alphanumeric passwords. They outline three types of graphical locks:

(1) Recall based (19:4-19:8): Occasionally referred to drawmetric systems are based on users remembering and recreating a secret image or drawing. Typically, users will draw on a blank canvas or grid, part of the secrecy comes from the lack of prompts or cues. The pass-go system[Tao and Adams 2008] uses intersections on a go-board instead of cells. Both

the Microsoft picture password and the Android lock-screen were built off this. These passwords tend to have the same vulnerabilities to social engineering and phishing attacks in addition to the attacks described in the introduction.

(2) Recognition based (19:8-19:13): Also known as cognometric systems, these passwords have the user select images from a dataset. To authenticate the user proves to the authenticator that they know the secret images by: (1) Passfaces [Davis et al. 2004] where the user selects several faces from a dataset, or (2) Dejavu [Dhamija et al. 2000] where the user chooses a moderate number of random images presented in a 5x5 grid and must select their 5 images. This shares many similarities with the common captcha technique used to detect bots on websites. These passwords are defendant against shoulder-surfing attacks and users showed that they have a higher retention rate when compared to other graphical and alphanumeric passwords.

(3) Cued Recall (19:13-19:18): These passwords have the user select specific areas on an image. To authenticate users must prove their knowledge about these areas. Passpoints[Wiedenbeck et al. 2005] are where the user would click specific points on an image, and to authenticate they must repeat the initial actions in the same areas to prove the user knows about those points. This system holds the same vulnerability to shoulder-surfing as other graphical passwords, and because it's based on mouse click location malware screen scrapers pr mouse loggers would be sufficient to determine the secret password.

Biddle et al. state that all graphical passwords have two main weaknesses: Exhaustive searches, and Capture attacks (19:23). Exhaustive searches are prevalent, due to small sample spaces for memorability. Capture attacks involve intercepting, tricking the user into giving you the password, or reconstructing it from previous login attempts. Our project is building off the analysis of recall-based passwords in an attempt to see if changing the size of the grid has an impact on the types of attacks. It will further expand the types of recall-based passwords since the current implementation of them is based around fixed square grids of sizes 3-4.

2.2 A Study on Usability and Security Features of the Android Pattern Lock Screen

[Andriotis et al. 2016]
Andriotis et al. conducted a survey to collect usable and secure user patterns. They used three statistical measures to analyse the data: pattern complexity, Pattern symmetry, and usability features. Pattern complexity is measured by quantifying four metrics: pattern length, directional changes, overlapping nodes, and knights move (edges connective disjointed nodes). They identify two types of pattern symmetries: reflective and rotational. Usability features are based on user features such as handedness and linguistic style, which affect the direction users draw their patterns.

It was found that most people's passwords resemble letters or numbers. Sequential straight lines were the most popular patterns, where letters Z, M, N, L, and the number 7 are the most common. The top left node was the most common starting point, and the pattern tended to flow in the same direction the user reads (ie. left to right, top to bottom).

They ultimately found that due to this bias, there is a dramatic shrinking of the graphical password space. They created a case study that could feasibly reduce the possible password space of length 6 to 20 guesses out of 26,016 possible passwords. Our project aims to build off this study to see how changing the size of the grid and implementing a strength meter changes the entropy of the user password. We believe that changing the shape and size of the grid will encourage users to create more diversity in their passwords.

2.3 Graphical Passwords in the Wild

[Alt et al. 2015]
Alt et. al. evaluated the PassPoints system, where the user picks a set of ordered points on an image background. Users could guess with 50% accuracy the (unordered) points of someone else based just on the background image (317). Saliency masks do better than blind guessing with only about 20% guess accuracy, but humans do even better with the aforementioned 50% accuracy(320-321).

For grid-based graphical passwords, users tend to swipe in predictable directions (mostly down and to the right). Further, the more swipes there are, the shorter individual swipes tend to be (320). We will build off these findings to see if users will have the same tendencies if they are shown the current strength of their password, by implementing a graphical meter.

2.4 Refining Graphical Password Strength Meters for Android Phones

[Heidt and Aviv 2016]
Heidt and Aviv assert that current strength meters incorrectly assume that there is a linear relationship between pattern features since users have inclinations towards certain patterns as stated in 'A Study on Usability and Security Features of the Android Pattern Lock Screen'. In their paper, they created a unique strength metric based on commonly used features. There are 9 key features to consider which are: start (S_p), length (L_p), crosses (C_p), non-adjacent (A_p), knight-moves (K_p), turns (T_p), Euclidean distance (E_p), maximum vertical or horizontal distance (D_p), and non-repeated segments (Np). They displayed the strength through a real-time number guesser and a gradient meter bar. Instead of creating a single formula, for every pattern the generated Markov chains to guess the probability. Based on their own monotonic comparisons, their proposed meter performed better than other existing meters at representing the guessability of the password.

We are implementing the several equations listed in this piece to use as a comparison to our unique meter. Like their approach, we have implemented a visual strength meter. We are expanding our grid size to see if the equations remain consistent and reliably show the strength provided by different grid dimensions.

2.5 Is Bigger Better? Comparing User-Generated Passwords on 3x3 vs. 4x4 Grid Sizes for Android's Pattern Unlock

[Aviv et al. 2015]
Aviv et al. assert that there is little entropy gain by expanding the

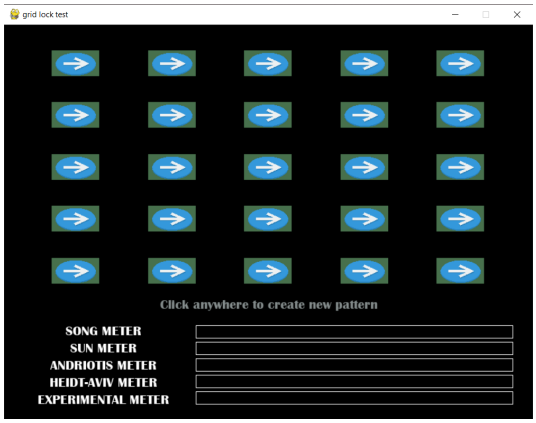

Figure 2: Our application with a blank 5x5 grid

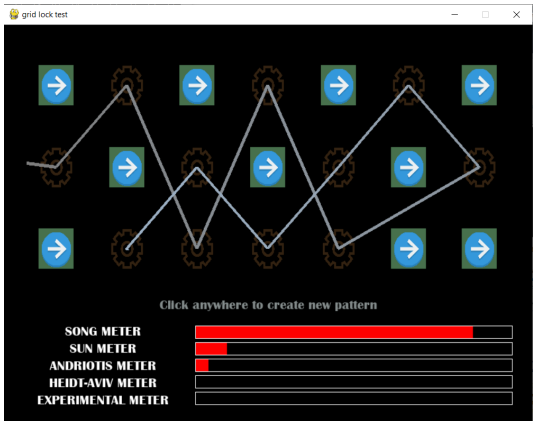

Figure 3: Our application with a 7x3 grid with a password

grid size of a pattern-based password system because users tend to use similar types of shapes. They conducted two large-scale studies to collect and compare the differences in human-generated 3x3 and 4x4 patterns. From their data, they show how users will have similar start and end positions and the length will be relative to the grid dimensions. By using Markov chains, they were able to compute the likelihood of a pattern based on their data set. Through this method, they could crack 92% of $3 * 3$ and 67% of $4 * 4$ passwords given unlimited tries, and 16% of $3 * 3$ and 67% of $4 * 4$ passwords given only 20 tries, which is the max allowance for a phone.

Our main idea to test the previous papers on differing grid sizes came from this paper. But we found the paper limiting as they only expanded the grid by 1 in each dimension. We aim to see how having different X and Y dimensions will affect the entropy as well as add more sizes to the study pool. Our system can create and set the size of the grid to any dimension provided X and Y are greater than 0.

3 PROPOSED WORK

We will implement the Song, Sun, Andriotis, and Heidt-Aviv meters [Heidt and Aviv 2016] by creating a Python application that simulates the Android grid-lock. The application will prompt a user to enter desired dimensions NxM ($M, N \in \mathcal{N}$) which the application will use to create an arbitrary-sized grid-lock. The user will be asked to draw a pattern using the grid and the application will determine and display its strength based on the Song, Sun, Andriotis, and Heidt-Aviv meters separately.

Through testing and observation of the outputs of the application on different patterns and grid sizes, we will analyze the differences between the meters and investigate the varying considerations they use to define strength in terms of the pattern's features e.g. defining pattern strength based on the length, linearity, or complexity. The goal of the work is to define pattern strength and effectiveness based on the specified meters and determine how strength changes in arbitrary-sized grids.

4 IMPLEMENTATION

For the project, we implemented an application in Python to simulate a grid-based graphical password lock. The application was created in Python using the PyGame library and takes a pair of numbers on startup to determine the dimensions of the grid. It was designed so the user can change the size of the grid to any NxM size where both N and M are greater than or equal to 1. The user creates a password by clicking and dragging across the nodes in a pattern and the strength meters are recalculated in real-time. We successfully created 3 visual meters to show the difference in ratings between the methods. The meters fill left to right based on the strength from weak to strong respectively. The meters update for every node added to the password pattern.

We do not have a password length limit for this project, but we did set self-imposed parameters to control the experiment later. The first rule is that any node can only be selected once. The second rule is that any straight line of nodes must connect to each node in the line. Unsurprisingly, the user must input 2 nodes onto their pattern for the meters to start displaying their strength.

4.1 Song Meter

[Song et al. 2015]

Developed by Song et al, the song meter measures strength based on the maximum vertical or horizontal distance (D_p), the number of non-repeated segments (N_p), and the number of crosses (C_p) in the password. Using their equation

$$M_p = 0.81 \left(\frac{Dp}{15} \right) + 0.04 N_p + 0.15 \left(\frac{min(C_p,5)}{5} \right)$$

They classify a pattern as being weak if between the range 0-0.33, medium between 0.34-0.67, or strong between 0.68-1. This equation was easily scalable with any size grid as it does not rely on the size or number of nodes in the equation. Since the Song meter relies on quantities that scale regardless of grid dimensions to determine strength it was easily adaptable to arbitrary grid sizes. The meter was interpreted as a percentage of the bar since the usual strength range for a 3x3 grid was between 0 and 1.

4.2 Sun Meter

[Sun et al. 2014]

Developed by Sun et al. this meter calculates the strength based on 4 metrics: total code length (L_p), the Euclidean distance between

points (E_p), the number of crosses (C_p) and the number of non-adjacent points (A_p). Using the equation

$$PS_p = L_p * \log_2(E_p + C_p + A_p)$$

The result will be a number between 6.24-46.807 with the scale increasing in strength as the output gets larger. The sun meter was not significantly changed since the quantities it relies on can scale with the grid without losing their intended functionality. The strength results on arbitrary-sized grids exceeded the original values. To represent this as a meter we normalized the resulting strength against the difference in grid size from a 3 by 3 grid. This is a purely cosmetic change and did not alter the raw scores that were used in the analysis phase.

4.3 Andriotis Meter

[Andriotis et al. 2014]

Developed by Andriotis et al. the meter calculates strength based on 5 metrics: the start point (S_p), the total length (L_p), the number of turns (T_p), the number of knight moves (K_p) and the number of non-adjacent points (A_p). Given by the equation,

$$\Delta = N \cdot X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_p \\ L_p \\ T_p \\ K_p \\ A_p \end{bmatrix}$$

This meter measures the strength based on weak, medium, and strong as the results fall between 0-1, 2, and >=3 respectively. The Andriotis meter relies on knight moves that needed to be revised in order to maintain its intended effect on the calculated strength. Knight moves (K_p) are defined as a move that is 1 node in one direction plus 2 nodes in the other direction. Since there are only 3 possible types (straight vertical/horizontal, 45 diagonal, and k-moves) of moves in a 3x3 grid this definition was adequate. Since more moves are possible in higher dimensions, these needed to be redefined: $Kp = log_2(d - 1)$. Where d is similar to hamming distance: $d = |\Delta x| + |\Delta y|$.

4.4 Heidt-Aviv Meter

[Aviv et al. 2015]

Developed by Heidt and Aviv their meter is based on the guessability of the patterns. They generated Markov guess probability by utilizing the pattern points as tri-grams and based it on previously collected data. They collected all the patterns and ranked them where the weakest passwords had the highest probabilities, and the strong passwords had a lower guess number.

We were unable to implement this meter because we did not have enough data to adequately represent the probabilities of patterns on arbitrary grid sizes. This meant that any implementation would likely be a guessing game. In order to apply Heidt and Aviv's method to arbitrary-sized grids, we would need a sizeable body of passwords that people use for each specific grid dimension. This is needed in order to create the probability of each possible tri-gram. Unfortunately, such a dataset does not currently exist, so an attempt to implement this meter would require either large amounts of guesswork.

5 EXPERIMENT

To assess the performance and accuracy of our adjusted meters we tested the working equations on 3 different sized grids. Firstly, to create a baseline we used a traditional 3x3 grid. We then used this to quantify our results from a 5x5 and a 2x8 grid. For the baseline, we created over 50 passwords ranging from weak, medium, and strong. These passwords were a mix of curated and randomly generated. For the curated passwords we defined their strength in human terms based on their complex characteristics such as the number of crosses and knight moves. For the random passwords, we used a web-based pseudo-random number generator to pick the next node. This had human oversight as there were move limitations due to the pre-established rules.

6 EVALUATION FRAMEWORK

Our project will either reproduce the findings from [Heidt and Aviv 2016] regarding the relative efficacies of the different meters in this new paradigm of variable grid sizes and analyze what that means, or contradict the findings and analyze why that might have been the case. Based on those findings, we will be able to compare our new strength meter for arbitrary-sized grids to the existing measurements. We will know our proposal is successful if it communicates to the user when their password makes good use of the larger size and is not easily guessable. We will use the metrics defined by Andriotis[Andriotis et al. 2016] to help with this analysis.

One of the primary challenges in this analysis will be that we cannot survey anyone outside this group for datasets, and we were unable to find any pre-existing datasets for non 3x3 or 4x4 grids.

7 FINDINGS

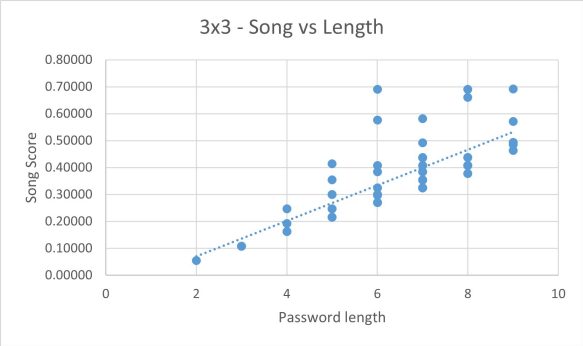

Figure 4: 3x3 - Song vs Length

For our baseline, we found that all 3 meters increased in strength as the passwords increased in length as seen in figures 4,5 and 6, which is as expected. Both Sun and Andriotis have a tight relation between strength and length whereas Song was better at assessing the complexity of a password more independent of its length. Their linear behaviour is consistent with their original results from their respective papers. The range of Sun and Andriotis changed slightly due to how we adjusted their values. For sun it was a slight change in the length calculation and for Andriotis was the refinement

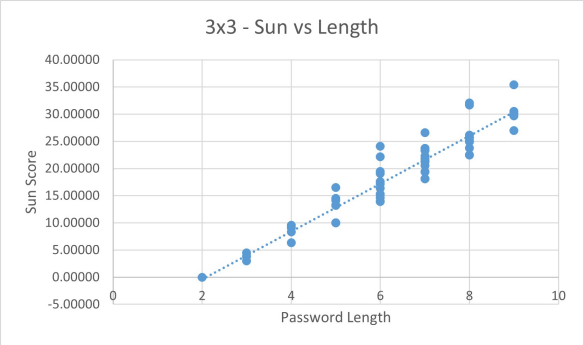

Figure 5: 3x3 - Sun vs Length

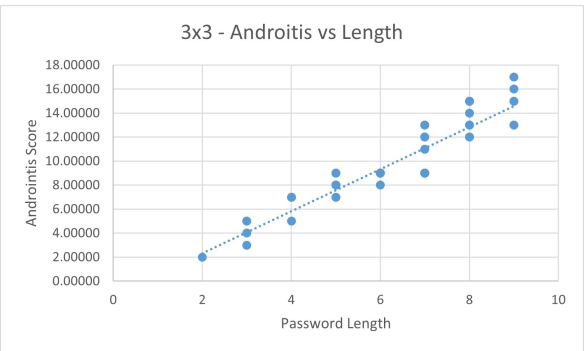

Figure 6: 3x3 - Andriotis vs Length

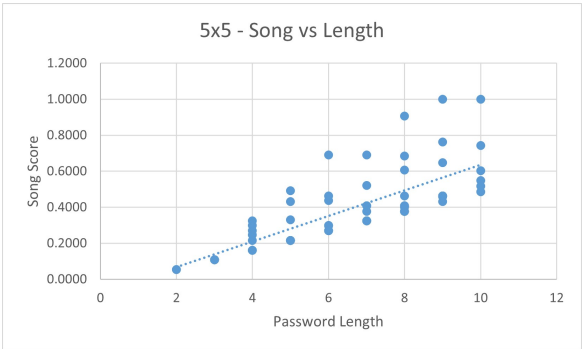

Figure 7: 5x5 - Song vs Length

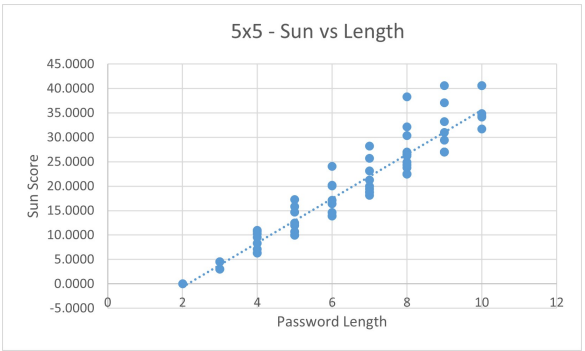

Figure 8: 5x5 - Sun vs Length

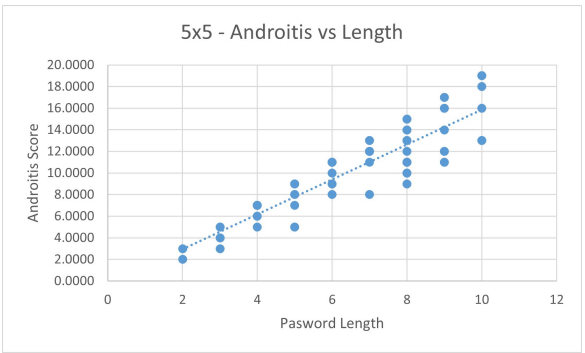

Figure 9: 5x5 - Andriotis vs Length

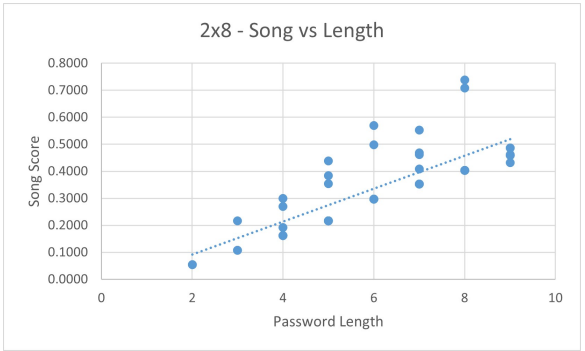

Figure 10: 2x8 - Song vs Length

of non-adjacent moves. We reflected the range variations in our graphical meters so that they would reflect the strength better.

For the 5x5 grid, we similarly created a set of passwords, this set was larger by 20 passwords of increased lengths to account for the larger grid size. From Figures 7, 8 and 9 can see that all 3 meters performed consistently with the baseline. We concluded that the meters would accurately display the strength of the password as the grid size increases. Again, we see the range of Sun and Andriotis increase slightly, confirming the need to adjust the visual bars to the size of the grid.

Our final test was performed on a 2 by 8 rectangular grid to see if the meters held for non-square or 'arbitrary' sized grids. As we can see in Figures 10, 11, and 12, our adjusted meters were able to produce results in line with the baseline measurements. Due to the limited space in the 2x8 grid, we did see larger ranges in strength with the longer passwords compared to both the baseline and the 5x5 grid. This is largely due to the restricted space limiting the number of special moves, such as crosses, and knight moves. Again, we see more variation along the song meter as the length increases which has been consistent throughout all the tests.

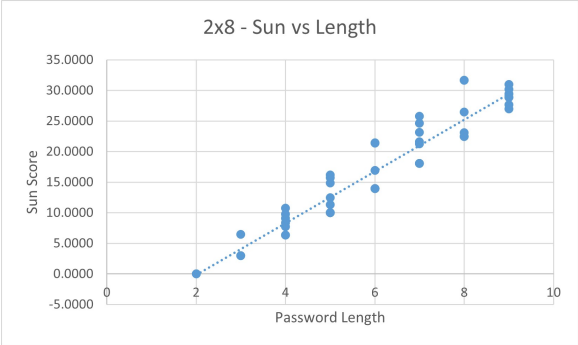

Figure 11: 2x8 - Sun vs Length

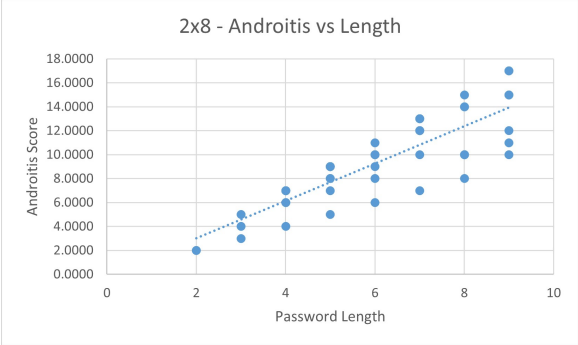

Figure 12: 2x8 - Andriotis vs Length

The song meter is the least dependent on length as a measure of security. From this, there is more variation in strength as the length increases. As we can see in Figures 4, 7, and 10 there gets to a point where a user can create a sufficiently strong password with less length. For all our grid sizes a password of length 6 can be stronger than a longer password of length 9. While increasing the length creates a more secure result floor, it does not necessarily lead to a strong password. From this, we concluded that the password's length is not the best indicator of its strength even in arbitrary-sized grids.

Additionally, we wanted to see how strength changed with just length. Pattern length was calculated by considering at what length will the weakest possible pattern in a square grid be considered as 'medium' strength in terms of reaching or going past the 50% mark of all meters Sun, Song, and Andriotis. We decided to conduct the observation on square NxN grids since pattern locks are typically in that format. Since we've observed in our study that the usage of more diagonal and knight moves made for a stronger password, we defined the 'weakest' possible password as one that only uses straight moves which would be a square spiral pattern. With this, we observed in Figure 13 the number of nodes needed in a square spiral password in order for all the Sun, song, and Andriotis meters to reach or go past 50% of their scales which resulted in this semi-cubical parabola graph with an approximate value of $0.71(N-3)^2+9$ where N is an integer >= 3.

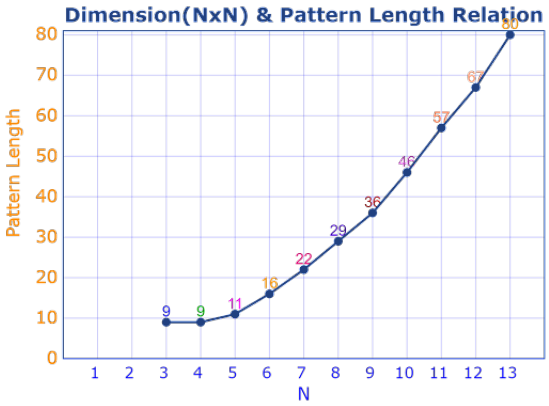

Figure 13: Graph showing required password length in order to achieve 50% on all meters for each grid size

From our implementation and experiment, we have 2 main results. The first result is that we successfully implemented 3 strength meters and created visual representations of them that are updated in real-time as the user creates a password. Our second result was key insights into what will make a strong graphical password. In terms of the meters, we believe that the Song meter is the best in terms of strength representation in arbitrary-sized grids.

We found that there are 4 things to consider when making a password in a grid-based system. The most important is pattern usability, the user should know and utilizes the full extent of moves that are allowed in the system. Notably incorporating knight moves, crosses and peg moves where the user is allowed to backtrack and skip over already selected nodes. Secondly, the user should utilize the full grid size by maxing out either the vertical or horizontal distance. Thirdly, the user should have little to no straight lines, as we discussed earlier the strength increases faster the more non-straight moves that are incorporated. Lastly, if the users incorporate the previous 3 features, then a password's strength will increase with its length.

8 PLANNING AND EXECUTION

8.1 Reflection

8.1.1 The Unexpected. During the project, we encountered two main unexpected obstacles; Heidt-Aviv data requirements and the difficulty of calculating 'knight' moves in non 3x3 grids. We discovered the first one when implementing the meters. We were so caught up in the math behind the meter we didn't realize there was no dataset for what we needed, and Heidt and Aviv didn't include their complete dataset for a 3x3 grid in their paper. The second also came up when we were implementing the meters, upon realizing that the Andriotis meter was returning scores that were lower than expected. Despite these, we succeeded in achieving our goals of implementing and gaining sufficient analysis of the differences of the other meters (Song, Sun, Andriotis) which helped us create a definition of pattern strength and effectiveness based on our findings thus successfully realizing our study's purpose.

8.1.2 What We Learned. Throughout this project, we learned a great deal about graphical passwords, such as the different kinds and their strengths and weaknesses. Due to the way we split the workload for this project, each person learned something different:

(1) Riley: While researching this project I came across a number of interesting insights about the strength of passwords and what makes a password strong. Some ideas made more sense than others, but this deep dive into the discussion around what makes a good password lead me to think about my passwords and whether (or why) I considered them secure. Also, learning about the other forms of passwords prompted me to try them out on some of my devices, which was an interesting experience even though I ultimately decided they were not as good as my current alphanumeric passwords.
(2) Chris: I have used PyGame before, but it was in a limited capacity and a while ago. Implementing our application, I learned about how applications implement graphics, such as PyGame's use of buffered frames to prevent screen flickering. I also did a lot of reading before we began, and I was fascinated by the variance in graphical password systems. I assumed going in they were all grid-based passwords, but we also found some examples of image recall passwords such as Dejavu [Dhamija et al. 2000] where users use entire images as a secret, and cued recall such as Passpoints [Wiedenbeck et al. 2005], where users use an image to create a set of secret points. Exploring these more in-depth was fun, and made me realize just how much there is in this field.
(3) Lauren: I learned that the existing strength estimators rely too much on length as an indicator of strength. I believe that the complexity of the password should have more weight, as it does in Song. After our analysis, I realized that our sample space was quite limited and I would have liked to see how real users would have created weak, medium, and strong passwords and how that would affect the strength vs length relationship. I personally learned that when working in groups that it is important to meet on a more frequent basis. I believe that our project would have proceeded better with more communication. This was most evident in our presentation where we ran over time due to us not practising as a group enough.
(4) Kelly: Having no previous experience with Pygame, I was delegated the task to implement the GUI for meters in the project application, thus I learned how to create GUIs with Python using Pygame. I also learned to think critically when making an observation about how password strength changed with length. When making this observation, I realized it was important to know basic calculus to produce results from graphs. In terms of working with my team, I learned that communication helped to unify all our delegated tasks together to complete the project. Additionally, I think it might have helped if we did more research on our proposal to see if all our goals are doable given the restrictions and instructions of the project. However, it all worked out because we had enough proposed meters to do the study.

8.1.3 What We Would Do Differently. The application itself runs fairly well, however, some of our decisions early in its development with how we track position made the meters difficult to implement. If we were to do it again, we would take advantage of classes in Python, and include more information about the nodes, such as the cardinal position in the grid. We would also try to include a wider survey to see how users interact with the various meters. Further, we would have liked to implement a new meter designed specifically with arbitrarily sized grids in mind. however, we ran out of time and could not include it in this project.

8.2 Project Plan

8.2.1 Phase One. The first phase of this project was to research our subject further and prepare our proposal documentation. We each researched several papers pertaining to graphical passwords, on January 26th we discussed each paper and how we would incorporate its ideas into our project. After a discussion, we finalized the 5 papers outlined in the related works section. When it came to the proposal the work was again distributed evenly as we all wrote an equal amount, which was completed on February 5th.

8.2.2 Phase Two. On February 28th we started phase two by planning project deliverables and dividing the workload, with the goal being to complete the implementation. Chris took the lead on developing the grid GUI. The GUI included the ability to create custom dimensions, select individual nodes, and link nodes together. Kelly created the visualization for the meters so that they were connected to both the user input and the entropy equations. Riley implemented the existing Song, Sun, Andriotis, and Heidt-Aviv meters. Lauren created the rough draft of the final paper by expanding on the introduction and related works. She also documented the planning and member contributions that were missing from the initial proposal. This would complete the implementation of our project and have a deadline of March 21.

During this phase, we found we were unable to implement the Heidt-Aviv meter, and after some discussion decided we didn't have the data to make it work. We considered implementing an approximation but decided it wouldn't add anything to the project.

8.2.3 Phase Three. Immediately after phase two, we began phase three where we would test our new meter and create a small dataset to compare the meters. Lauren and Kelly developed the new measure and its meter. While Chris and Riley detailed the implementation of the project in the final report. All group members participated in filling the dataset by manually generating unique passwords. While this limited our findings it gave a good comparison between the meter's performance. With the dataset completed on March 26th, we preceded to complete the questions and presentation.

8.2.4 Phase Four. For the final phase, we equally prepared for the presentation and the final paper. This consisted of presentation rehearsals and refining our slide deck. For the final paper, we already had a complete draft by the end of phase 3, so we focused on editing

and adding details with the aim to complete the project on April 11th.

REFERENCES

Florian Alt, Stefan Schneegass, Alireza Sahami Shirazi, Mariam Hassib, and Andreas Bulling. 2015. Graphical Passwords in the Wild: Understanding How Users Choose Pictures and Passwords in Image-Based Authentication Schemes. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 316–322. https://doi.org/10.1145/2785830.2785882

Panagiotis Andriotis, George Oikonomou, Alexios Mylonas, and Theo Tryfonas. 2016. A study on usability and security features of the Android pattern lock screen. 24, 1 (March 2016). https://www.emerald.com/insight/content/doi/10.1108/ICS-01-2015-0001/full/html

Panagiotis Andriotis, Theo Tryfonas, and George Oikonomou. 2014. Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method. *In Human Aspects of Information Security, Privacy, and Trust* (2014), 115–126.

Adam J Aviv, Devon Budzitowski, and Ravi Kuber. 2015. Is bigger better? Comparing user-generated passwords on 3x3 vs. 4x4 grid sizes for Android's pattern unlock. In *Proceedings of the 31st Annual Computer Security Applications Conference*. 301–310.

Robert Biddle, Sonia Chiasson, and P.C. Van Oorschot. 2012. Graphical passwords: Learning from the first twelve years. *J. ACM* 44, 4, Article 19 (Aug. 2012), 41 pages. https://doi.org/10.1145/2333112.2333114

Darren Davis, Fabian Monrose, and Michael K Reiter. 2004. On user choice in graphical password schemes.. In *USENIX security symposium*, Vol. 13. 11–11.

Rachna Dhamija, Adrian Perrig, et al. 2000. Deja Vu-A User Study: Using Images for Authentication.. In *USENIX Security Symposium*, Vol. 9. 4–4.

Susanna Heidt and Adam J Aviv. 2016. Refining graphical password strength meters for android phones. In *Twelfth Symposium on Usable Security and Privacy*, Vol. 16.

Youngbae Song, Geumhwan Cho, Seongyeol Oh, Hyoungshick Kim, and Jun Ho Huh. 2015. On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks. In *33rd Annual ACM Conference on Human Factors in Computing Systems*. 2343–2352.

Chen Sun, Yang Wang, and Jun Zheng. 2014. Dissecting pattern unlock: The effect of pattern strength meter on pattern selection. *Journal of Information Security and Applications* 19, 4 (2014), 308–320.

Hai Tao and Carlisle Adams. 2008. Pass-go: A proposal to improve the usability of graphical passwords. *Int. J. Netw. Secur.* 7, 2 (2008), 273–292.

Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. 2005. PassPoints: Design and longitudinal evaluation of a graphical password system. *International journal of human-computer studies* 63, 1-2 (2005), 102–127.