GridMap: Enhanced Security in Cued-Recall Graphical Passwords

Nicolas Van Balen and Haining Wang(✉)

Department of Computer Science, College of William and Mary,
Williamsburg, VA 23187, USA
{njvanbal,hnw}@cs.wm.edu

Abstract. Despite their widespread usage, text-based passwords are vulnerable to password cracking as users tend to choose weak passwords. This is mainly because the more secure a password is, the harder it is for a user to remember it. As a promising alternative, various graphical password systems, which take advantage of the fact that humans are more sensitive to visual information than verbal text, have been proposed over the past decade. However, graphical passwords come with their own vulnerabilities, such as high susceptibility to shoulder surfing and hotspots. In this paper, we develop a new cued-recall graphical password system called GridMap by exploring (1) the use of grids with variable input entered through the keyboard, and (2) the use of geopolitical maps as background images. As a result, GridMap is able to achieve high keyspace and resistance to shoulder surfing attacks. To validate the efficacy of GridMap in practice, we conduct a user study with 50 participants. Our experimental results show that GridMap works well in domains in which a user logs in on a regular basis, and provides a memorability benefit if the chosen map has a personal significance to the user.

Keywords: User authentication · Graphical password · Grid · Map image

1 Introduction

Passwords have been widely used for decades as the most common method for user authentication. It is estimated that an average person normally uses passwords for authentication 7.5 times every day [10] in order to accesses information ranging from emails to bank accounts. Whereas the text-based passwords are the dominant method of online authentication for these daily scenarios, their security depends on creating strong passwords and protecting them from being stolen. A strong password should be sufficiently long, random, and hard to discover by crackers, while a weak password is usually short, common, easy to guess, and susceptible to brute-force and dictionary attacks. However, the dilemma in a text-based password system is that a strong password is hard for a human user to remember—and more often than not, users tend to choose to create weak passwords simply because they are easier to remember than strong ones.

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 75–94, 2015.
DOI: 10.1007/978-3-319-23829-6_6

Attempts to have users employ more secure passwords by either forcing them to follow certain rules when creating them or randomly assigning passwords, have not successfully addressed the problem because users experience more trouble remembering these passwords.

Psychological research [1,15,17] suggests that humans can remember visual information with more ease than textual information. This has led researchers to study the use of graphical passwords as replacements for text passwords with the assumption that the use of visual information will reduce the memory burden placed on users when using more secure passwords. Moreover, three different memory retrieval approaches have been proposed for graphical passwords. The first approach, called *recall*-based, requires a user to retrieve his password directly from memory, usually in the form of a drawn picture or pattern. The second approach, called *recognition*-based, relies on a user's ability to recognize visual information that has been seen before. This approach generally gives a user a portfolio of images as his password and asks him to choose these given images from amongst a set of decoys as the password entry process. The third approach, called *cued-recall*-based, relies on a user's ability to retrieve information from memory given a cue. This approach usually has a user create a password using the image as some sort of direct or indirect guide. In some cases the password is contained within the image itself, and in others it is simply based on the image.

Graphical passwords, while improving on text based passwords in many ways, have also introduced new problems unique to them. Most of graphical password schemes are vulnerable to shoulder surfing attacks, in which a password is stolen by observation or recording during a login session. In this case, the ease of visual memory actually works against the password security. Many cued-recall systems also suffer from a problem known as hotspots, which stems from the fact that some parts of an image are more likely to be selected by users than others. In addition, many graphical password systems have difficulty attaining a large theoretical key space.

In this paper, we investigate the use of a grid input system in a cued-recall password system, in which a geopolitical map is used as the background image. Our design has a user choose those elements of a map image that have personal significance for creating a password, and then, to input the password by using a grid. In particular, each cell of the grid contains text, and the user needs to locate those cells that constitute his password and enter the text from each cell into a password field as parts of his password. The text in each cell is randomly changed for every login session, making the capture of the password considerably more difficult. This method allows us to retain the key space and memory cuing benefits of cued-recall schemes while significantly hardening the security via randomly changed text input and impacting usability as little as possible.

We develop a prototype of the proposed graphical password system, called GridMap, and validate its efficacy by running a user study involving 50 participants who create passwords and then log in again after varying periods of time. From this user study, we observe that GridMap works well in scenarios where

users log in on a daily basis, but has the drawback that users tend to take longer to log in and, if left on their own, will often choose predictable passwords. We also observe that the users who can find higher significance in an image will perform better at recalling their passwords than the users to whom the image is less significant.

The remainder of this paper is structured as follows. Section 2 surveys related work. Section 3 details the design of GridMap. Section 4 analyzes the security benefits of GridMap. Section 5 describes the prototype implementation of GridMap. Section 6 presents the experimental methodology and results of our user study. Section 7 lists the limitations of GridMap, and finally Sect. 8 summarizes our work.

2 Related Work

In the area of recall-based schemes, the most known system is Draw-A-Secret (DAS) [14]. Originally designed for PDAs, DAS has a user draw a picture on a grid and records the password as a series of pen-up, pen-down, and edge-crossing events. However, users of DAS were found to choose very symmetric patterns for their passwords, and, to address this, an enhanced system called Background Draw-A-Secret (BDAS) has been proposed [9], in which an image is used as a background to the grid resulting in a reduction of symmetric patterns. Zakaria et al. [22] developed a variant of DAS used on smartphones, and they proposed different methods, including the use of decoy lines and snaking lines, to provide shoulder surfing resistance.

Designed as an alternative to PIN numbers, a commercial recall-based system called grIDsure [13] uses a 5×5 grid of randomized single digit numbers combined with keyboard input. Such a design of grIDsure makes it difficult for a malicious observer to capture the PIN, leading to shoulder surfing resistance. An overview of security concerns of grIDsure is presented by Bond [3].

Research into shoulder surfing resistant systems has also been done with recognition based systems, in particular Passfaces [7] as the best known scheme in this category. Its basic idea is to have each user choose or be assigned a portfolio of images consisting of portraits of peoples faces. In order to authenticate, a user would go through multiple rounds, in each of which he would be displayed a set of nine images, one from his portfolio and the others as decoys, and need to click on the image belonging to his portfolio. One shoulder surfing resistant variation is studied in previous research [19], in which the shoulder surfing resistance of graphical passwords is compared to that of text-based passwords. In particular, the original Passfaces scheme is compared to alphanumeric text-based passwords and a variation of Passfaces which uses the number pad on the keyboard, instead of the mouse, for input. It is observed that the Passfaces variation outperforms both the original Passfaces scheme and the text-based passwords alike, in terms of shoulder surfing resistance. Another variation of Passfaces has been proposed by Dunphy et al. [8], which uses eye tracking technology to determine a user's choice by tracking where his gaze is on the screen.

In the cued-recall area, the most well known password scheme is PassPoints [20,21]. This scheme stores a password as a series of points on an image, in

which a user needs to click on. A variation of this scheme called Cued-Click-Points (CCP) [6] has also been proposed. In CCP, a user chooses one point on each of five different images rather than five points on a single image. As each point progressively maps to a different image, a user's password constitutes a path of images determined by the choices of points the user makes. However, both systems have been shown to have a problem known as hotspots, where certain points in an image are more likely to be chosen by a user than others. To tackle the hotspot problem, a variant CCP called Persuasive Cued-Click-Points (PCCP) [5] has been proposed, in which a user could only choose points from inside a given viewport that is randomly located on the image. The location of this viewport could be changed with a shuffle button. A recent variation called Cued-Gaze-Points (CGP) [11], similarly to Dunphy's variation on Passfaces, uses eye tracking hardware for the input of the users points in order to avoid shoulder surfing. Another cued-recall system is called Inkblot [18], in which a user is shown a series of images and asked to think of a phrase that describes each image and use the first and last letters of each phrase to form a password. This system, although much less vulnerable to dictionary attacks, has a considerable amount in common with text-based passwords than other graphical passwords.

3 Design of GridMap

While most graphical passwords are susceptible to shoulder surfing, click based schemes are particularly vulnerable as it is easy to visually follow the cursor on the screen and track the locations of the user's click points. Even more of a concern is the possibility of the screen being recorded, which can now be easily accomplished with the wide spread use of handheld recording devices such as smartphones.

To the best of our knowledge, previous efforts in this area have focused on solutions that require specialized hardware, or on systems that are designed for very specific user authentication environments. This suggests that an alternative input method that does not leave visual queues on the screen would be preferable, and for this, baring the use of specialized hardware, the keyboard is the best option.

The design guidelines of GridMap lie in two aspects. First, we should use an image that can provide enhanced memorability, and second, the input method must be able to meet the security requirements of general purpose imaged based passwords, including high key space requirements, resistance to phishing and shoulder surfing attacks, which are the security problems many graphical password schemes have been plagued with. GridMap meets these design guidelines by (1) using geopolitical maps as the memorability enhanced image and (2) creating an adaptation of the grid input system to address the security and usability concerns of a graphical password system. In general GridMap is capable of providing more secure user authentication, especially greater resistance to shoulder surfing. Meanwhile, GridMap is able to provide similar, if not much improved, usability as the existing click-based schemes.

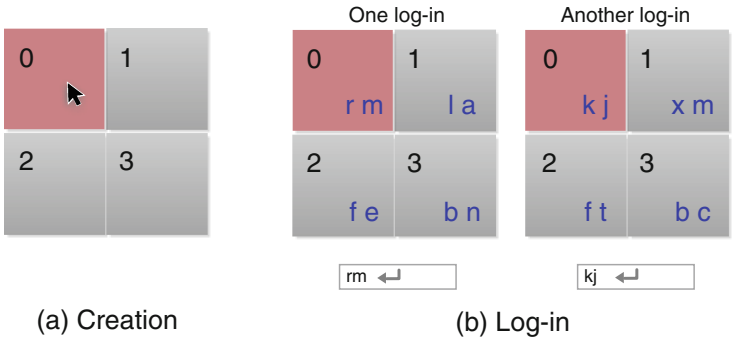

Fig. 1. On the left, sub-figure (a) shows what a grid would look like during the password creation phase. Sub-figure (b) on the right shows an example of the text used in the grid for verification and login. Note that the numbers remain constant while the letters change for different login sessions. Here the user's chosen cell is the top right one with the number of 0, highlighted in red, the letters from that cell would be entered as the password as seen in the text boxes in the example.

3.1 Basic Design

The basic working mechanism of GridMap is to superimpose a grid on top of the image of a map dividing it into cells. Each of these cells contains two forms of text. One is a variable (changes every session) text used to input the password, and the other is a fixed form of text used to aid in remembering the password. During the password creation phase, a user chooses a series of cells from the image as his password by simply clicking these cells via the mouse. And for the purpose future logins, the user needs to remember the location and related features, including the fixed number, for each selected cell. During a regular login session, the user recalls the chosen cells and types in the variable text inside each of these cells into a password field, which hides the typed text like it does for a text password. Once the entire string is typed into the password field, it is converted to the coordinates of the cells, which are the input to the system for user authentication. Note that the password comprises the cells chosen by the user, not the text that is entered into the password field. The text that the user inputs is dependent on what is displayed in those cells an will change with each login session. Figure 1 shows a very simple example of how this input method works with a 2×2 grid and one selected cell. For the presentation purpose, the variable text typed in the password field is not hidden.

The user can also choose to change the map image used as the background or the alignment of the grid within the image. The image can be selected from a pool of available images, and the user can choose the one with the most meaningful features to him such that it would be the easiest to remember. The alignment of the grid within the image can also be changed so that the cells that comprise the password can line up better with the features chosen by the user. In our current design,

all the cells in the password must be chosen using the same image and grid alignment. Both configuration setups are saved by GridMap as a part of the password.

Upon submission, the password is sent to the server in the form of grid coordinates, i.e., the row and column numbers of the chosen cells, along with two characters which identify the chosen image and grid alignment. Since this graphical password information is simply a string of numbers, the server can treat it the same as a text password and save it using a hashing function. In other words, the server can treat the passwords generated with our scheme as same as regular text-based passwords. The graphical part of our scheme is implemented on the client side, and no change is needed on the server side.

3.2 Design Choices

Here we discuss a few design choices made in GridMap. The first choice is the use of maps as the background images, from which users choose passwords. The second choice is how many cells a grid should divide the image into and how many of these cells a user needs to create a password. Finally, we discuss the exact choices of the variable text, which is used to input the password, and the fixed text that is used to aid in memorability.

Image Choice (Maps). Although stock images are usually used for cued-recall graphical password systems, we choose to use maps instead. We believe that although, in a generic sense, there is no benefit to one image over another [20], images that have more significance or are more personally meaningful to a user would result in passwords that are easier to remember. For this reason, maps are chosen to be used as the images in GridMap since many users may give a personal significance to the location portrayed in a map. Users could then choose these locations as the password making it easier to remember. One concern with this design is that an attacker with intimate knowledge of a user could use his personal information to guess the password, however, gaining this type of personal information is costly and only affects one target rather than a large password corpus.

The particular maps we use are geopolitical maps or ones portraying commonly known landmarks and other characteristic of the region portrayed in them. A landmark or a state may hold more significance to a user than a specific address, so that this type of map is preferable to a street map. This also helps in the sense that with a street map a user is likely to choose an address of his own home, which would make it easier to guess than a vacation location or place where relatives live for example. A street map also poses a larger problem for implementation since it requires less detail or a smaller area, which is less likely to contain something significant to a user to be displayed, and, for this reason, we choose not to use them. An example map with a corresponding grid is illustrated in Fig. 2, showing the "key" state of Florida with a grid superimposed.

Number of Cells in Image and Password. We also need to decide on how many cells to divide the grid into. The problem comes with the difficulty of

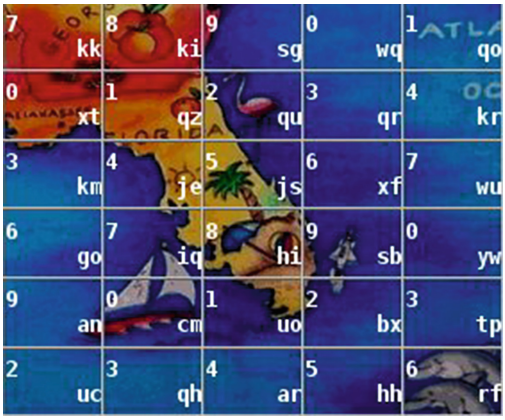

Fig. 2. A portion of a map showing the state of Florida with the grid superimposed

leaving the image uncluttered and the text visible. For this reason, the image needs to be very large taking up most of the screen. Taking low resolution screens, such as those on many laptops, into consideration we set the grid to have no more than 500 cells in it. We observe that much more than this number leaves the image too cluttered and the text in individual cells hard to focus on. If a larger keyspace is desired, we suggest to increase the number of grid alignment options or the number of maps available for a user to choose from, instead of the number of cells in an image. On the other hand, we do not recommend the use of less than 300 cells in an image as the keyspace becomes too small and too many features of the image end up in each cell. We make three different grid alignments available for a user to select from, which is consistent with existing systems, and recommend that the sum of the cells among all three grid alignments be no less than 1200 cells in total. The default grid contains 500 cells, but the user has the option of using a grid with 400 cells or a grid with 300 cells instead.

GridMap uses a minimum of five cells per password, which is consistent with most of the existing cued-recall schemes. If a loss of theoretical keyspace is acceptable, the number of cells in a password can be lowered to four to achieve better usability; however, we recommend that no lower than five cells per password be used in scenarios where keyspace is a concern.

Variable Text. The variable text, which the user types into a password field as input, is comprised of two lower-case letters. Both numbers and symbols are avoided because most users are more used to typing from the alphabetical part of the keyboard rather than the numerical portion, given the fact that most of the typing done by a user is for writing natural language. We also avoid using upper-case letters to eliminate the need for a user to press the shift key, especially given that the text in the password field is hidden and it will very hard for a user to see if he made a mistake by typing a lower-case letter where a capital should be or vice versa.

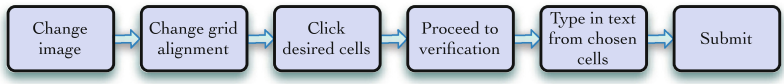

Fig. 3. Password creation and confirmation processes.

We set two letters per cell to minimize the amount of typing that a user needs to do when inputting the text. Using a single letter should be avoided. This is because there are not enough letters in the alphabet list to give each cell a unique letter, making it easier for an attacker to guess a password in a brute force attempt where guessing a single letter would cover multiple cells at once. Each cell could include three letters, which has the advantage of using actual English words in cells; however, we feel that the advantage of being able to use words is not significant enough to justify the extra typing time necessary to input them. We do not recommend the use of more than three letters as it leads to have the grid getting too cluttered with text and takes considerably longer to input.

Fixed Text. Each cell additionally has a single digit number in it. These numbers do not change between sessions and are organized in such a way that two cells with the same number will not be closer than 4 cells away. We use numbers here for two reasons. One is to avoid having users confuse this text with the variable text which does not use numbers, and more importantly the other is to help a user to create a meaningful sequence, like a zip code, to aid in memorability. Sometimes a user may remember the general area, in which a cell in the password is located, but not the exact location of the cell. Thus, with the help of numbers, the user can pinpoint the exact cells in the password. And, since the two numbers with the same value are far enough away from each other, it is unlikely that the same number will show up twice in the same area. This will aid a user in remembering the order of the chosen cells, which can be of concern as shown in previous research [16]. The authors of [16] compared the memorability of text passwords with that of passwords strictly based on images, and they observed that most users have more trouble remembering the order of the items of their password than the exact contents.

3.3 Password Creation and Confirmation

The password creation procedure of GridMap is very different from the login procedure. We assume that the password is created in a private environment like a home or office with the user having a mouse and keyboard available for input. The image is presented to the user with the fixed number in each cell, but without the variable text used for input. Then he just moves the mouse and clicks on the chosen cells rather than typing in text from them. Before that, the user needs to make a decision on the choices of image and grid alignment. Such a creation process allows the user to concentrate on the image without the text

and prevents the user from attempting to form a password based on the variable text, which would change every login session. Note that although GridMap is vulnerable to shoulder surfing attacks during the password creation phase, as it is expected to be conducted in a private environment, and only once per user account, we believe that the security risk is low. Meanwhile, users will simply be warned of the risk and use discretion when creating a password.

Once the password is created, a user will be asked to re-input the password in a confirmation step. During the confirmation phase, the image with both numbers and letters is shown to the user, and the user resorts to the regular input method (i.e. typing the letters into the password field). The purpose of the confirmation is to help the user be familiar with how GridMap works and memorize the password. Figure 3 illustrates the password creation and confirmation processes.

3.4 Password Login

During a regular login session, GridMap acts as same as its confirmation process. A user has to use the text from the cells as input via the keyboard. For user convenience, GridMap could give a user the option of choosing to either type the text from the keyboard or simply click the cells via the mouse. If users are in a private environment like home, they may choose this more user friendly clicking method for input. However, in a general case, users should use the default input device—keyboard—to type the text into the password field.

4 Security Analysis

The theoretical keyspace for GridMap is dependent on the number of images available, the number of grid alignment options available, the number of cells in a given grid alignment, and the number of cells in a user's password. The following equation is used to calculate the keyspace measured in bits:

$$log_2 \left(\sum_{i=1}^{K} m \left(\frac{n_i!}{(n_i - r)!} \right) \right),$$

where m is the number of images available, r is the number of cells in a password, K is the number of grid alignment options available, and n_i is the number of cells in a grid alignment i. In our design, m could range from one to three, r could range from four to five, while K is set to three and then the value of n_i, corresponding to individual grid alignments, will be 300, 400, and 500, respectively. Note that although our implementation provides two-image options, in this analysis we set the value of m to one for ease of comparison with existing systems whose keyspace calculation assumes only one image. Below we show the theoretical keyspace in bits for GridMap.

$$log_2 \left(\left(\frac{500!}{(500-5)!} \right) + \left(\frac{400!}{(400-5)!} \right) + \left(\frac{300!}{(300-5)!} \right) \right) = 45.28 \ \ bits$$

Table 1. A comparison of GridMap and the two most similar schemes, Passpoints and grIDsure.

	grid input system	Passpoints	grIDsure
Theoretical keyspace	45	43	18
User choice resilience	None	None	None
Variant response	Yes	No	Yes
Server probes	0 - 1	1	0

It is clear that the keyspace of GridMap is within the range of 40–60 bits that accounts for the average keyspace of text passwords. This value may increase depending on how many image choices are available in the deployment and how the images are used.

The combination of a grid and random input text enables GridMap with a higher shoulder surfing resistance than either click-based graphical passwords or traditional text-based passwords. An attacker trying to shoulder surf would need to keep track of every letter combination a user types in as well as locate the cells in the grid that match the typed letters before the user submits the password. This makes it very difficult to steal the password since both the letters typed by the user and the text filling the grid must come from the same session, and memorizing one ahead of time would not give any advantage. It would still be possible to capture the password with a recording device, but it would be much more difficult due to the need of recording both the screen and the keyboard. This would make it impractical to use a handheld device such as a smartphone for recording, since only the screen can be easily seen from a distance and getting close enough to record the keyboard would likely make the attacker's intention obvious. Mounting an attack with recording devices would require very discrete cameras that can see both the screen and the keyboard well enough to distinguish what the user is typing, which can only be achieved under very limited circumstances.

This resistance is also able to defend against malware like keyloggers. Even though the input is done via the keyboard, a keylogger alone would not suffice to capture a password. The random variant nature of the text would require an attacker to capture the screen as well as the keyboard input to actually recover the password.

Resistance to phishing attacks can be built into GridMap, but it's effectiveness depends on how GridMap is implemented. A strong resistance against phishing attacks can be gained by eliminating the need for a user to select the image at the login time. With this method, when a password is created, the user would still choose, or be assigned automatically, an image to use as the background for the grid; however, when the user returns to login, the chosen image will always be shown as the background automatically so that there is no need for the user to choose the correct image for login. Without knowing the right background image for login, a phisher cannot create a close to real phishing page

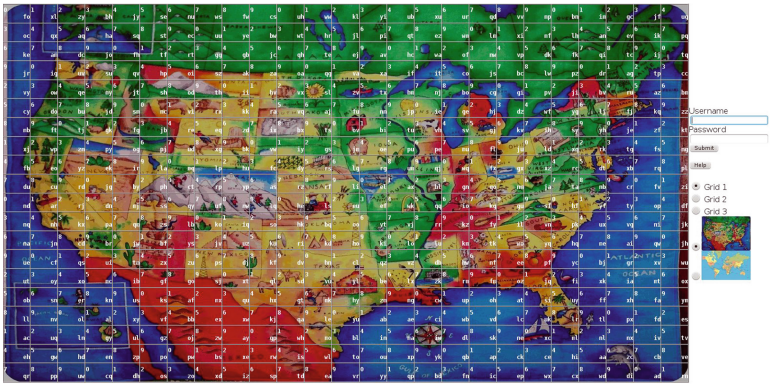

Fig. 4. A screen shot of GridMap.

to deceive a user. The drawback to this method, however, is that the keyspace is reduced to the case in which m is set to 1.

Compared with previous schemes, GridMap has no obvious advantages on the issues of hotspots and user predictability. The grid is conducive to predictable patterns, such as five cells in a row, and the map image is still just as likely to have hotspots as in an existing click based system. However, we theorize that GridMap will have an increase in patterns due to the grid and a decrease in hot spots because (1) the grid lines split many of the images features and (2) the maps used are more likely to have different features be significant to different users. It is possible to further reduce the problems by applying persuasive technology such as that used in PCCP, which will be explored in our future work.

In Table 1, we can see a side by side comparison of GridMap with the two most similar schemes, Passpoints and grIDsure. The data on these two existing systems is taken from the graphical password survey by Biddle et al. [2].

5 Implementation

A prototype implementation of GridMap is developed for this study. This prototype mainly consists of two web-based user interfaces: one used to create a new password, and the other used as a login page. Both user interfaces are written using HTML, CSS, and Javascript, and each of them has a corresponding PHP script on the server.

The grid portion is created using an HTML table, in which each table data element corresponds to a cell and the image is set as its background. The table is generated using a javascript loop, and every table data element is divided with two $< div >$ elements. The first $< div >$ contains the static number, which is generated using the pattern described before and displayed on the top left corner of the cell. The second $< div >$ contains a two-letter string (i.e., two lower-case randomly changed letters) displayed at the bottom right corner. These strings

are read into an array from a file containing all possible combinations of two lower-case letters. The array is then shuffled and used to fill in the cells by order of index. The array is re-shuffled on every page refresh. Since the number of strings in the file is larger than the number of cells in any single grid alignment, it is possible that two sessions will have different sets of strings filling the grid.

For all these numbers and letters in the grid, bold font is used for visibility. Opposing corners are used so as to cover up the least amount of the image displayed in each cell as possible. Upon implementation, we noted that if the space given to the table is too small, it is not easy to view the image over the text, and in some cases, there is even no enough space for the text. To deal with this problem, large images are used and the table is set to automatically take up as much visible space as possible. This entails taking up the entire vertical space that the browser allows a webpage, while taking up whatever horizontal space left by the authentication form.

To simulate the scenarios where three grid alignments line up differently, we choose to change the number of cells and divide the image into 500, 400, and 300 cells, respectively. When the number of cells changes, the size of each cell changes as well to accommodate filling the image. This makes a grid alignment with less cells have bigger cells, resulting in the cell borders to locate in different parts of the image.

The authentication form contains two text fields: one for username and the other for password, like those used in text-based passwords. To input a password using the typing method, the user would simply need to type the two letter strings from the bottom right corners of the chosen cells into the password field. For the click input method, each table data element is given a onclick event handler. When a user clicks a cell, a Javascript identifies the two-letter string for that particular cell and appends it to the end of the current content of the password field. In both cases, the user can simply erase the string in the password field and start over if the user thinks he may have made a mistake. The form also contains two sets of radio buttons: one set allows the user to change the grid alignment, and the other set allows the user to change the background image. When one of the radio buttons with a grid alignment option is pressed, a Javascript function regenerates the table with the new number of cells. The radio buttons with the image options each show a thumbnail of the image and also call a Javascript function which changes the image and, in some cases, the color of the font to create enough contrast with the image. In some cases, it is even necessary to reduce the brightness of the image to draw enough contrast and see the characters.

When the user clicks the submit button, a Javascript function is called. This function reads the content of the password field and replaces each pair of letters with the indexes of the row and column of the chosen cell. This step is necessary because the pair of letters in each cell randomly changes with every session, GridMap cannot store the password as those letters. Instead, it must store the coordinates of the chosen cells. This is done on the client side to avoid the overhead of sending all the mappings between text and coordinates of cells to the server.

Fig. 5. Image options provided to users. The U.S. map on the left is set as default, and the World map to the right could be switched to if desired by users.

The function also performs error checking, such as passwords are too short or text does not match with the letters in the corresponding cells. If a problem is detected, the form is not submitted and the user is given an alert indicating the error. Should no error be found, two numbers, one identifying the image and one identifying the grid alignment are appended to the end of the text in the password field. Then, the form is submitted to the server. A PHP script on the server checks if the username exists and the password is correct. It then gives the user feedback by either notifying a successful submission, or by displaying an error message indicating that either the username does not exist or the password is wrong, and provides a link back to the authentication page.

In this prototype system, no password hashing is implemented for two reasons. The first reason is that it would not allow for certain types of analysis, such as hotspot analysis, to be performed; and the second is that hashing is not directly related to what we are attempting to address and would only be an additional step that requires implementation. We assume that in a real deployment the passwords should have been hashed.

6 Evaluation

We conduct a usability and user predictability study involving 50 participants with age from 18 to 36. The majority of participants are college undergraduate students from a variety of majors. The rest are grad students in Computer Science, except for two who are professional software developers and one who is an office manager. All participants are regular computer users. Twenty one of the users completed the study as part of a class while the rest did the study over the Internet at their leisure. The methodology used in this study has been approved by the University's board of ethics for testing on human subjects.

Each of the users is directed to a webpage with instructions on how to use GridMap. The instructions are presented using hypertext as recommended by Forget et al. [12]. The participants from the class session are also given a demonstration by an instructor, while the remaining participants only have the

Table 2. The number of successful logins in 3 and 5 attempts and unsuccessful logins for participants who waited 1 day, 1 week, and 2 weeks between creation and login.

User Group	3 attempts	5 attempts	Unsuccessful
1 day	18/21	18/21	3/21
1 week	14/23	14/23	9/23
2 week	3/6	5/6	1/6

provided instructions. There are no other differences in the experimental methodology between the two groups. During the first session, the users are asked to create a password and then re-enter it as a confirmation. As mentioned before, in the creation of the password, the users are shown the grid with the static numbers in the cells only, and users click on the chosen cells via mouse to form the password. The participants are able to choose between a map of the United State and a map of the World, as shown in Fig. 5, with the U.S. map set as default. Some users create passwords with four cells and some create passwords with five or more cells. For the confirmation step, given the grid with both numbers and letters, the users are asked to re-input their passwords by typing the random text into the password field via the keyboard.

Certain rules that the participants are not aware of have been applied at password creation. These rules disallow the use of more than two consecutive cells in the same row, more than two consecutive cells in the same column, more than two consecutive cells in the same diagonal line, and the use of more than two corners. These represent the patterns observed in previous trials of the similar input system [4]. If a user violates one of these rules when creating his password, an alert box will be displayed to make the user aware of the rule being violated. The violation of a rule is recorded. The password field is then reset to empty and the participant has to create a new password.

During the second session, the users are asked to attempt to log in within five trials after either one day, one week, or two weeks. If a user is unable to log in within the five attempts, then the system simply informs him that he is done and does not ask for the password to be input anymore. A group of 21 participants, called the *1 day* group, completed the login portion of the study after at least 12 hour but less than 48 hours. Another group of 23 participants, called the *1 week* group, completed the login portion of the study after waiting at least 7 days, but less than 14 days. Finally, a group of 6 participants completed the login task after waiting more than 14 days. We refer to this final group as the *2 week* group.

An additional survey is also filled out by 42 of the participants, asking the following questions:

- How many years have you lived in the United States? Please give an answer as a whole number rounded down, e.g., use 0 if less than one year.
- How many states within the U.S. have you visited/lived in?
- How many Countries have you visited/lived in?

Table 3. The number of successful and failed logins of users with 4 and 5 or more cells for all 50 participants involved in the study.

User Group	4 cells		5 cells	
	Succeeded	Failed	Succeeded	Failed
1 day	7/7	0/7	11/14	3/14
1 week	6/9	3/9	8/14	6/14
2 week	3/4	1/4	2/2	0/2

Table 4. Password creation and login times displayed in seconds.

	Creation (second)	Log in (second)
Mean	136.6	51.8
Max	514	223
Min	18	4

This survey is made available as we theorize that the amount of travel done by users can effect their passwords and image choices.

In rest of this section, we summarize our data analysis and findings with regard to the usability and predictability of user choice of GridMap.

6.1 Success Rates

We record two success rates for each of the groups 1 *day*, 1 *week*, and 2 *weeks*, respectively. The first one records the number of users who are able to correctly reproduce their passwords within 3 attempts, and the second one records the number of users who are able to correctly reproduce their passwords within 5 attempts. Across all three groups, we achieve a 70 % success rate within 3 attempts and a 74 % success rate within 5 attempts.

Table 2 shows in detail how many users are able to successfully log in after 3 and 5 attempts as well as how many are unable to remember their passwords. We note that after 1 day 86 % of users are able to remember their passwords, but after one and two weeks only 61 % and 83 % of users remember their passwords, respectively.

It is also worth mentioning that the 2 week group was originally comprised of much more than six participants. However, out of this larger group only the six participants shown in Table 2 were able to remember their usernames at login time. As such, the others are excluded from this study since we are only interested in participants who can at least correctly recall their usernames. This accounts for the higher success rate after two weeks than after one week in our data. Note that all the users in the 1 day and 1 week groups were able to remember their usernames.

We also compare the success rates of users who used 4 cells in their passwords with those who used 5 or more cells. After one day, 79 % of users who with 5 cells and 100 % of users with 4 cells were able to log in, but in the cases of the groups who logged in after one and two weeks, only 63 % of users with 5 cells and 69 % of users who used 4 cells were able to successfully log in. These results suggest that using a password length of 4 cells, instead of 5, can improve a user's ability to remember his password when login is done on a regular basis; however, as the time lapsed between logins increases, the memorability benefit provided by the shorter password decreases and is no longer justifiable due to the loss in security. The detailed results are listed in Table 3.

6.2 Timing

There are two timing metrics we are interested: (1) the amount of time taken by the participants to create a password and (2) the amount of time taken to input the password during a login session.

The time a participant spent for creating a password is measured by taking a time stamp when the page has been fully loaded and a second time stamp when the user successfully submits a password to the server. The measured time of a participant for creating a password is reflective of the entire process since a failed submission attempt, such as one that violates a rule by having too many cells in a row, will not cause the second time stamp to be taken. For the login process, we use a similar method, but every password submission to the server, correct or incorrect, is logged separately. In other words, if a participant makes three attempts to get the correct password, three separate times would be recorded. This is because we are interested in how long it takes a password to be input but not how long an entire login process would take.

The means along with the maximum and minimum values for the creation and login times are listed in Table 4. We believe that all the values for password creation and the mean for login times are accurate; however, it is not likely that the maximum and minimum values from the login column would be observed often in practice. In the case of the lower end, it is observed that all the values but two under 20 s are caused by those users who are unable to log in. It is likely that most of these users give up trying to remember their passwords and simply enter the easiest password possible to use up all five tries. In the case of these two users who are able to log in, one of them uses the same cell multiple times in the password, and the other has two cells in a row followed by two cells immediately below the first two. The rules for creating a password in our implementation simply disallow a user to have more than two cells next to each other in a row, column, or diagonal, but does not put any restriction on repetition. Thus, none of the cases mentioned above are in violation of these rules.

Figure 6 illustrates the distribution of login times, i.e., the times taken by the users to enter their passwords. The values at the two extremes, i.e., less than 10 or higher than 70 s, are not likely to be observed in practice. There are instances of users who were distracted while the login page was open or who forgot their passwords and simply tried to complete the five trials as fast as

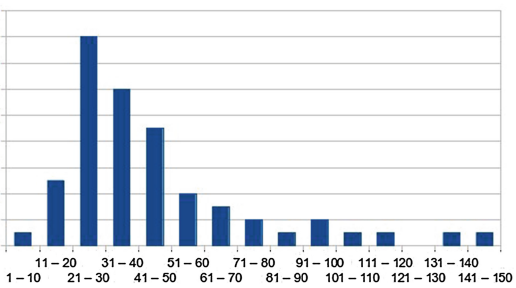

Fig. 6. The distribution of login times. The x-axis shows the time in ten second intervals, and the y-axis shows the number of users who logged in with that time interval.

possible, which likely account for the values at the two extremes. There are also cases in which the users had typos leading to the letters not matching with those in the grid. In these cases, the form fails to submit and the user needs to reenter the password with the correct characters, resulting in a longer login time. Due to this observation, we believe that in practice most users would display login times between 18 and 35 s, but this would require more extensive testing to confirm.

6.3 User Predictability

Due to the tendency of users to create predictable passwords, we enforce certain rules to prevent users from creating what we believe are the most common passwords, a straight line and the four corners; but we record those attempts that violate the rules. In this way, we are able to know how many users would have created one of those passwords if allowed and still measure the predictability of passwords without these common cases.

We observe that 24 % of the 50 participants attempted to make one of these passwords. On inspecting the data, we observe that many users still created predictable passwords such as every other cell in a row or column, and two adjacent cells in a row followed by two adjacent cells in a row directly below.

In order to visually characterize this user tendency, we measure the distance between each cell in the password with every other cell in both the vertical and horizontal directions. For example, if a password has a cell in row 5 and another in row 6, the vertical distance between these two cells would be one since you would need to move over a distance of one cell to move from one point to the other. Equivalently, if two cells in a password are both on the same column, their horizontal distance is zero.

The frequency with which each distance occurs is represented as a bar shown in Fig. 7. The `x` axis represents a distance and the `y` axis represents the number of pairs of cells that are found to have that distance from each other. The top graph displays the calculated vertical distances (i.e., the number of rows between cells) and the bottom graph displays the horizontal distances (i.e., the number

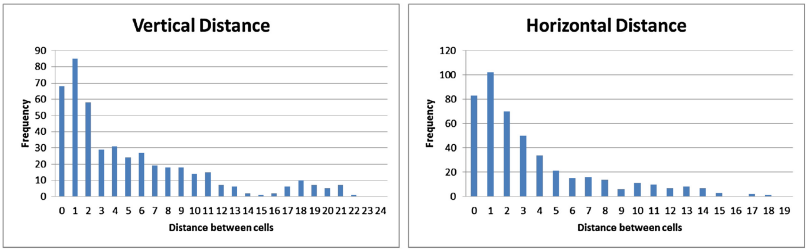

Fig. 7. Distributions of distances between cells in passwords.

of columns between cells). As a whole, each of these graphs can be viewed as a probability distribution of the distance between cells.

Both graphs have the very similar shape with the higher frequencies in the lower distances and the highest frequency occurred in the distance of one cell. This implies that users are more likely to choose cells that are close to each other rather than those cells that are farther away, with the most probable distance of being one.

6.4 Other Observations

We also study whether a user's history of travel and residence can affect his performance under GridMap, which uses geopolitical maps as the background image, using the data gathered from the survey.

We observe that users who are able to successfully remember their passwords have traveled, on average, more than those who could not remember their passwords, with an average of 10.48 states and 4.24 countries visited for the former group, and 5.88 states and 2.66 countries in the latter. We also observe that among users who choose the U.S. map, the ones who are able to successfully log in have lived in the United States with an average of 17.53 years as opposed to 10.66 years for those who cannot remember their passwords.

We think that these results are due to the fact that users who travel more and spend a longer time in the locations depicted in the map have a higher familiarity with the locations in cells on it. This would mean that there are more cells that are significant to such a user available as choices in a password making it easier to remember later. This would suggest that providing a user with a map of an area that is familiar and significant to him will increase the chances of remembering his password.

7 Limitations

One drawback of GripMap is the amount of time it takes for a user to input a password. This is expected because the user must perform the task of visually locating his cells on the image first and then typing those cells' text into

the password field. This procedure, for most users, involves looking from side to side across the screen with intermittent typing in between. In consequence, the resulting times recorded during login sessions are slightly slower than desired. However, we feel that many of these numbers are skewed towards one of the extreme cases: some users either spend a lot of time trying to recall their passwords or perform other tasks while the webpage is already up and has started to count time; while other users simply submit blank or bogus passwords to fulfill the five tries as fast as possible. We believe that in a real deployment, for those users who are familiar with GridMap, the time will be between 18 and 30 s, but more research is required to verify whether this is true or not.

Another problem of GripMap is the tendency of users to choose passwords with predictable patterns. About one quarter of the users in our study attempt to create highly predictable passwords. Given the restricted rules for creating a password, we can still see a high degree of clustering among the created passwords, which could be exploited to mount a dictionary attack. However, this security threat can be greatly reduced by employing persuasive technology similar to that used in PCCP [5]. A number of cells in the grid would be randomly chosen and grayed out, forcing the user to choose from the cells that are still clear. A shuffle button would allow the user to gray out a different set of cells if the current selection is not to his preference. This issue will be investigated in our future work.

8 Conclusion

Based on grid input and geopolitical maps, we have proposed a new cued-recall graphical password system called GridMap, which is more secure than the existing graphical password schemes in terms of keyspace and shoulder surfing resistance. In addition, the robust design of GridMap defends against malware like keylogger and phishing attacks. We have developed a prototype of GridMap and conducted a user study involving 50 participants. Our experimental results show that GridMap works well for user authentication on a daily basis. Moreover, we have observed that those users who are more familiar with the map images have less difficulty recalling their passwords. This observation implies that we can further improve the memorability of GridMap by providing map images that are more significant to users. In our future work, we will investigate how to shorten the password input time and will apply the persuasive techniques for GridMap to reduce user password predictability.

References

1. Paivio, T.R.A., Smythe, P.C.: Why are pictures easier to recall than words? Psychon. Sci. **11**(4), 137–138 (1968)
2. Biddle, R., Chiasson, S., Oorschot, P.C.V.: Graphical passwords: learning from the first twelve years. ACM Comput. Surv. **44**(4), 1–41 (2011)

3. Bond, M.: Comments on grIDsure authentication, March 2008. http://www.cl.cam.ac.uk/mkb23/research/GridsureComments.pdf
4. Brostoff, S., Inglesant, P., Sasse, M.A.: Evaluating the usability and security of a graphical one-time pin system. In: Proceedings of the 24th BCS Interaction Specialist Group Conference, pp. 88–97 (2010)
5. Chiasson, S., Forget, A., Biddle, R., van Oorschot, P.C.: Influencing users towards better passwords: persuasive cued click-points. In: BCS HCI, vol. 1, pp.121–130 (2008)
6. Chiasson, S., van Oorschot, P.C., Biddle, R.: Graphical password authentication using cued click points. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 359–374. Springer, Heidelberg (2007)
7. Passface Corportion.: The Science Behind Passfaces. http://www.passfaces.com/published. Accessed June 2013
8. Dunphy, P., Fitch, A., Olivier, P.: Gaze-contingent passwords at the ATM. In: Proceedings of COGAIN 2008, September 2008
9. Dunphy, P., Yan, J.: Do images improve "draw a secret" graphical passwords? In: Proceedings of ACM CCS 2007, October 2007
10. Florencio, D., Herley, C.: A large-scale study of web password habits. In: Proceedings of WWW 2007, pp. 657–666 (2007)
11. Forget, A., Chiasson, S., Biddle, R.: Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In: Proceedings of CHI 2010, pp. 1107–1110 (2010)
12. Forget, A., Chiasson, S., Biddle, R.: Supporting learning of an unfamiliar authentication scheme. In: AACE E-Learn, E-Learn 2012. AACE (2012)
13. GrIDsure. http://www.gridsure-security.co.uk. Accessed May 2013
14. Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., Rubin, A.D.: The design and analysis of graphical passwords. In: Proceedings of USENIX Security Symposium 1999, August 1999
15. Kirkpatrick, E.A.: An experimental study of memory. Psychol. Rev. **1**, 602–609 (1894)
16. Komanduri, S., Hutchings, D.R.: Order and entropy in picture passwords. In: Proceedings of Graphics Interface 2008 (2008)
17. Shepard, R.: Recognition memory for words, sentences, and pictures. J. Verbal Learn. Verbal Behav. **6**, 156–163 (1967)
18. Stubblefield, A., Simon, D.: Inkblot authentication. Microsoft Research Technical report, (MSR-TR-2004-85)1–16 (2004)
19. Tari, F., Ozok, A.A., Holden, S.H.: A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In: Proceedings of SOUPS 2006, July 2006
20. Wiedenbeck, S., Waters, J., Birget, J.-C., Brodskiy, A., Memon, N.: Authentication using graphical passwords: effects of tolerance and image choice. In: Proceedings of SOUPS 2005, July 2005
21. Wiedenbeck, S., Waters, J., Birget, J.-C., Brodskiy, A., Memon, N.: Passpoints: design and longitudinal evaluation of a graphical password system. Int. J. Hum.-Comput. Stud. **63**, 102–127 (2005)
22. Zakaria, N.H., Griffiths, D., Brostoff, S., Yan, J.: Shoulder surfing defence for recall-based graphical passwords. In: Proceedings of SOUPS 2011, July 2011