An Analysis of Graphical Authentication Techniques for Mobile Platforms as Alternatives to Passwords

Robert J Hannah
University of Illinois: Urbana-Champaign
Senior Research
Advised by Professor Klara Nahrstedt

Abstract

The username and password authentication design has been a long standing institution in the wide set of interactive, user-based technologies. More recently, the emergence and growth of mobile technologies has prompted the need for an evolution in authentication avenues that have increased touch-input friendliness for users, while maintaining a similar level of security that passwords have oft provided. Our research evaluates three graphical authentications designs via simple implementations on the Android platform, volunteer feedback on basic usage of the implementations, and conceptual analysis of the security offerings for each, all while weighting pertinent aspects against the common password authentication approach.

I. Introduction

Authentication is a basic access control concern for applications and services in the technological landscape that usernames and passwords have commonly addressed. Alongside the likes of certificate based authentication, hardware tokens, and personal identification numbers, passwords have taken the forefront in modern authentication standards with functional recognizability, in that passwords are so common, the average user recognizes what they are and how to use them without much instruction. Although as Bonneau, Herley, Oorschot, and Stajano state, "over forty years of research have demonstrated that passwords are plagued by security problems and openly hated by users."[1] In recent years, mobile and touch screen technology has seen massive growth and has carried over these well-founded authentication techniques, but present novel challenges unique to the mobile macrocosm, as well as the inherent deficits passwords already face. In example, full size keyboards allow efficient entry of alphanumeric strings on typical workstations, while touch screens often sharply diverge in this aspect by offering compact graphical keyboards that unavoidably raise the difficulty to use. Likewise, smaller screen constraints on mobile devices often cause issues for graphical text entry elements that necessitate an explicit programmatic approach to maintain visibility and ease-of-interaction. Further, touch screens allow an entirely new facet of interaction that current password implementations simply do not employ.

This research presents three graphical authentication techniques that aim to trend towards the combinatorial strength of passwords, while effectively utilizing touch interaction with means to enhance the authenticating experience for mobile device users. Thereafter, we evaluate each approach as a singularity and also in contrast to the password model in order to propose alternatives better suited for the mobile era.

II. Foundation

The interactive touch display of mobile devices is an integral part to their ease and wide spread use, which itself suggests graphical means of authentication.

The first design, hereafter referred to as *Word-Image Association*, allows a user to choose several proposed words, each corresponding to one or more images in a preexisting storage. When authenticating, the user is presented with several of the images randomly chosen from the storage, and the images corresponding to the user's chosen words must be chosen (tapped) to authenticate. Order may or may not matter.

The second, *Cell Sequence*, presents the user with a grid of cells (e.g. 4x4). Credential creation simply has the user choose (tap) one or more cells, and the sequence thereof is saved as the credential. Authenticating is then just a matter of replying the sequence of cell selections into a login grid. At its base, this technique is reminiscent of Android's "Pattern" unlock feature, where a sequence of dots is used as the credential. We extend this conceptual avenue by adding a single, differing integer in each cell, which are changed on every login, and by adding a background image to the grid of cells. To the former, it is often necessary to authenticate over a network to an authenticating server. The sequence of cells tapped will be represented by the digits in the cells, the placement of which would act as a shared secret between the client and server, and would change on every login attempt. This would be reminiscent of hardware tokens (e.g. RSA tokens). The background image would assist a user in recognizing and differentiating cells.

The third, *Shape Builder*, presents a grid that the user may fill with given single-cell shapes, creating one big image from the placement of the single shapes. Credential creation has the user create a custom image from the individual shapes, while authenticating has the user recreate their previously set image.

III. Metrics

To modularize our inspection, we use simple qualitative metrics to evaluate each design in a security-versus-usability perspective. This falls short to consider many deloyability[1] and implementation specific concerns, as we aim to gauge the theoretical viability of the proposed conceptual designs on a basic level, as opposed to a deeper expose on associated real world implications.

For usability, we consider:

- *Ease of Use* is high when the user has a high amount of intuition on how to use the interface, as well as when a user can provide credentials efficiently and unencumbered.

- *Rememberability* is high when there exists aspects about the credential or authentication that make it easier to recall from memory.

For security we'll consider:

- *Strength Against External Observation* can be viewed as being inversely related to the level at which any physical onlooker to a user entering their credential would be able to learn the secret. As such, this metric would be high if a viewer could witness a user authenticating, yet be unable to learn the credential. This is especially important for the mobile platform, as mobile devices are frequently used in public spaces, as opposed to common desktop platforms.

- *Strength Against Internal Observation* references the ability of software to capture the credential during the authentication process. This includes software that my eavesdrop from within the device itself as well as eavesdrop on network communication. This metric is high if such software would be unable or unlikely to capture the credential, or unable to impersonate the owner with captured data.

- *Strength Against Brute Force* refers to the sheer size of the combinatorial space of differing credentials, as well as the ability to make reasonable assumptions about real users' choice of credential for the purpose of increasing likelihood of success during a brute force attempt. The latter is much like using a dictionary attack against a password authentication since real users are more likely to use real words in their passwords. The metric is high when the combinational space is large and there are few reasonable assumptions to be made for probabilistic gain.

IV. Design & Development

We introduce three implementations, each respective to one of the conceptual authentication techniques described previously: *Word-Image Association, Cell Sequence,* and *Shape Builder*. We designed each as a basic form of the concept it represents, and developed all three in one mobile application. Each implementation allows creation of a single credential per technique and subsequent authentication with simple feedback of login success or failure. The application was built using the Cordova[2] framework allowing for simple cross-platform compiling, although the majority of user testing was performed on the Android platform. It is important to note here that certain implementation specific choices were made in the designs to follow that may be altered for increased security or usability without diverging from the core authentication concept. We will discuss this further during our analysis, later in this paper.

Figure (1) illustrates the selection menu. "Auth Type[s]" A, B, and C are *Word-Image Association, Cell Sequence,* and *Shape Builder*, respectively. Figure (2) shows the login success/failure feedback screens.

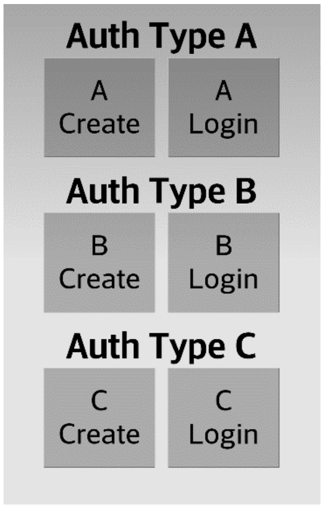

Figure (1)

Figure (2)

The *Word-Image Association* credential creation, seen on the left in figure (3), presents a predefined list of selectable words, of which the user may choose three (e.g. bird, envelope, and lock are selected in figure (3)). The associated authentication screen, seen in the right in figure (3), presents a grid of images that relate to the credential word options on a one-to-one basis. The images are inserted into the grid randomly from the storage. The user must select (tap) the images corresponding their chosen credential words. In our implementation, order doesn't matter; the user may select their images in any order. Since the images are displayed at random and less images are displayed at a single time than there are credential word options, the user may press refresh to have the grid refilled with another round of randomly selected images from the storage. Once three image selections have been made, the authentication is evaluated and the appropriate feedback screen is displayed.

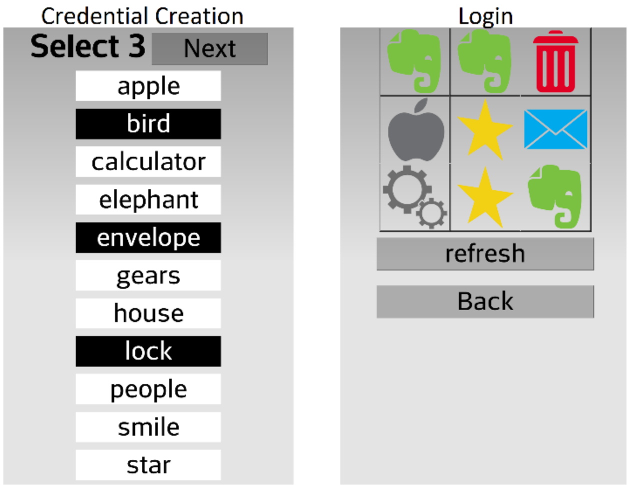

Figure (3)

Cell Sequence, shown in figure (4), has a similar interface for both credential creation and authentication. The main crux of the display is the grid of selectable cells. During credential creation, the user may tap one or more of the cells in any order they choose, allowing each cell to be chosen only once. A number is displayed after each cell selection to indicate the order in which the cells were tapped. This acts as user feedback for review before confirming a series of selections. The left panel in figure (4) illustrates a creation in progress, as the user has already chosen the top-middle, middle-left, and bottom-right cells, in that order. The background image behind the grid seen in the figure acts as visual memory assistance. The login view, seen in the right panel in figure (4), presents the same grid and image, but with numbers one through 9 inserted into the cells. In our implementation, these numbers are randomly placed in the grid, but in they serve to represent a rotating cell encoding scheme described later in our analysis. To authenticate, the user simply needs to reenter their secret sequence of cells and tap the "Login" button.

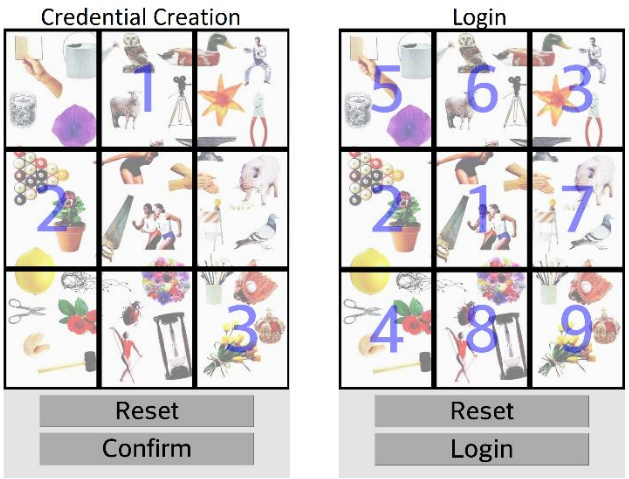

Figure (4)

The *Shape Builder* implementation in figure (5) presents yet another grid, but also four selectable, colored shapes. A user may tap any of the color-shape options and the selection will be highlighted with a red border. Thereafter, taps in the grid will place the current selected color-shape option in the tapped cell. Tapping the red-highlighted color-shape option a second time will de-select the option, and allow tapping of filled cells to remove the placed color-shape. This interaction is similar for both creation and login. Figure (5)'s left panel shows a credential creation in progress, where many of the shapes have been placed in various cells, while the right panel similarly show an authentication attempt in progress. The clear buttons simply clears the grid of color-shapes.

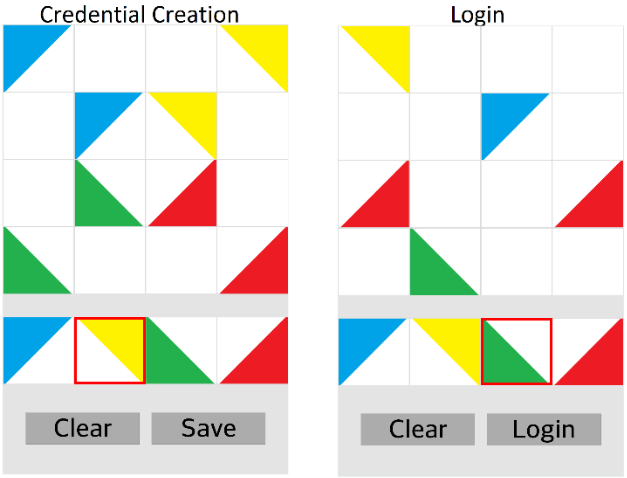

Figure (5)

V. Analysis

We focus on a pure usability versus security approach to evaluating each design independently, as well as in relation to the common password technique, using the metric described previously. Much of our consideration assesses the theoretical advantages and limitations of the designs, while the basic implementations serve to better observe and collect feedback on real world usage of authentication employing such design concepts.

- *Word-Image Association*
  Ease of Use benefits from an efficient login procedure. Simple taps on a few images in succession (three in our implementation example) quickly allow a user to authenticate. Our specific example suffers from the random filling of the login grid with images as there may be the same images many times, and the user's credential image may not be displayed at all which could require multiple refreshes. Also, the design requires few instructions and is easy to learn. In relation to a password, *Ease of Use* rates

higher in this design as we've removed the need for any keyboard-esque input which improves speed of authentication and better utilizes single-finger touch input.

Rememberability rates moderately for this approach. Our implementation specific choice of three possibly unrelated words creates some difficulty, but the conceptual design allows for alterations to increase this metric. One possibility may be to allow custom user words that they may be able to personally relate to, or entire phrases that relate to the images instead of single words. Also, remembering many multiple-word credentials for authenticating to different services employing this design could prove difficult as well. Given it's simple object-name-to-image association scheme as opposed to a mixed alphanumeric and symbols string, we deem this design easier to remember than a password, although the real world usage drawback of using the same credential as a memory crutch may not exist for this design as it does for passwords, since differing implementations may use different set of words.

Strength Against External Observation rates low, as a physical onlooker could easily observe the authentication process and know the secret. In a more expansive implementation, a feature could involve multiple pictures in relation to a single word in such a way that the secret bearer could easily realize their credential word visually, yet remain uncertain to an onlooker. Passwords rate higher in this metric, as common implementations obscure the password on screen using asterisks or the like.

Strength Against Internal Observation can vary depending on implementation specifics. Our application simply handled the credential in the clear, but encryption or hashing could easily be added to decrease the internal visibility window. The design does not intrinsically protect against network capturing so a TLS layer or similar solution would be necessary in this prospect. Malware that may monitor touch input would be unable to accurately gain information about the secret due to the rotating image placement on screen. In this regard, *Word-Image Association* has an advantage over passwords since an effective keylogger would be infeasible. Otherwise this metric rates similarly for passwords.

Strength Against Brute Force rates low in our implementation, which offers ten word options and ten associated images. Since our application does not care about image order, and words can only be chosen once in a credential, we have only

$$\binom{10}{3} = 120$$

combinations of possible credentials. Requiring that order matters increases this to

$$\frac{10!}{7!} = 720$$

permutations. Clearly this is unacceptably small, but we can trend higher with larger word bases. A dictionary-sized word base would quickly increase the combinational space. Against passwords, this metric rates far lower for our design.

- *Cell Sequence*
  Ease of Use, in a conceptual context, rates very well, as the design allows for quick

authentication and the design is familiar to existing authentications, such as Android's pattern lock screen, lowering the learning curve. In our implementation, the background image and displayed numbers upon login creates visual noise for the user, slightly lowering this metric. We rate this design's *Ease of Use* higher than that of password usage due to its simplicity.

Rememberability rates well with a single instance of this credential. The background image assists visual memory in associating chosen cells with objects and further differentiating cells from one another, other than pure position in the grid. As with passwords, multiple instance of this authentication for different services may lead to credential repetition or loss of rememberability.

Strength Against External Observation is low for this design. While the rotating numbers in the login pane serve to obscure whether the user is pressing specific numbers or pressing specific cells, this would only be effective against a single view from an attacker with no prior knowledge of the authentication process. Also, inspection of the device may divulge common screen presses via fingerprints that would reveal the secret. This would rate lower than that of passwords.

Strength Against Internal Observation rates well, as the rotating number scheme serves to encode the cell sequence entry differently on every authentication attempt. Proper implementation could ensure that the secret is never in memory or transferred via a network the same way twice, defending against on-device and network monitoring. In relation to passwords, this design has the advantage.

Strength Against Brute Force is a major drawback for this design. In order to trend combination space higher, the number of cells would have to be increased, but in a small amount of steps increasing cell count, we view a large drop in *Ease of Use*. For our example application, we use a 3-by-3 grid and require one or more cells in the sequence, thereby creating a combinational space of size

$$\sum_{i=1}^{9} \frac{9!}{(9-i)!} = 984{,}409$$

The design itself doesn't have many options for expansion in this regard without heavily degrading usability, as previously stated. Passwords fare much better in this metric.

- *Shape Builder*

  Ease of Use is rated moderate for this design. More complex and therefore more secure credentials with this authentication would require more input time and longer authentication time overall. On the other hand, this design is simple to learn and provides an almost game-like interaction that makes it feel less like an obstacle to overcome and more of an enjoyable user experience overall. Our specific implementation uses many taps to select and place shapes, although an alternate avenue may be to allow drag-and-dropping of shapes onto and around the grid which would feel more natural on touch devices. We find this design to have slightly higher *Ease of Use* than password usage.

  Rememberability for this design is heavily tied to the complexity of the built credential. If the credential built is a meaningful image to the owner in some respect, the credential retains a large amount of rememberability without necessarily increasing guess-ability by a would-be attacker. We argue that this design offers more rememberability than passwords.

Strength Against External Observation is fairly low, since one view of the authentication process would divulge the secret. We rate this lower than for passwords.

Strength Against Internal Observation follows previous design discussions in that a level of added encryption or hashing would limit the credential exposure. Static secrets such as this often suffer from being able to be replayed.[1] We rate this similar to passwords.

Strength Against Brute Force is respectably high for this design. Considering our example application, the grid is 4-by-4 and we provide 4 shape options, 5 accounting for a blank cell. This totals 5^{16} combinations of credentials. We could easily increase this space by increasing grid size and/or increasing the number of shape options. While increasing the grid size would increase the combination space at a faster rate than increasing the shape options, increasing grid size also causes a more rapid deficit in usability than an increase in shape options. While passwords still have the advantage in this area, expansion of this design may trend toward that of passwords.

Feedback

For usability insight, we gathered feedback on each of our example implementations from a small set of anonymous research volunteers. To facilitate a minimal quantitative measure, we devised a simple one-to-five rating system for volunteers to respond with on four areas: general ease of use, credential rememberability, personal feeling on level of security (i.e. how secure they would feel using the authentication on a personal device), and viewpoint on technical security. Also, we had volunteers gauge each implementation in relation to passwords in each area as less than, equal to, or greater than, which we represent with a -1, 0, 1 scheme for calculation. Figures (6) and (7) show computed averages of our received feedback.

	Ease of Use	Rememberability
WIA	4.25	3.50
WIA v Pass	0.75	-0.25
CS	4.00	3.50
CS v Pass	0.50	-0.50
SB	3.50	4.50
SB v Pass	0.00	0.00

Figure (6)

	Personal Security	Technical Security
WIA	2.75	2.50
WIA v Pass	-1.00	-1.00
CS	3.75	3.00
CS v Pass	-1.00	-1.00
SB	4.50	3.00
SB v Pass	-0.75	-0.75

Figure (7)

This data is simply a small sample to roughly gauge common users' perspective on the designs in the form of our basic implementations.

VI. Conclusion

Of the options explored, we conclude that the *Shape Builder* approach offers the best usability/security balance with room for scalable expansion. While none may be the perfect replacement for a password authentication, we offer the designs and our analysis thereof as advancement in the mission to simply the authentication process on mobile and touchscreen platforms.

VII. References

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. "The quest to replace passwords: a framework for comparative evaluation of Web authentication shcemes". University of Cambridge, Computer Laboratory. 2012. <http://research.microsoft.com/pubs/161585/QuestToReplacePasswords.pdf>

[2] Cordova - http://cordova.apache.org/