PassGame: Robust Shoulder-Surfing Resistance Through Challenge-Response Authentication

Jonathan Gurary*, Ye Zhu*, Nahed Alnahash†, and Huirong Fu†
*Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, Ohio
Emails: j.gurary@vikes.csuohio.edu, y.zhu61@csuohio.edu
†Department of Computer Science, Oakland University, Oakland, Michigan
Emails: nalnahas@oakland.edu, fu@oakland.edu

***Abstract*—Mobile devices are constantly exposed to the risk of shoulder-surfing by prying eyes and video surveillance. In this paper, we propose *PassGame*, a shoulder-surfing resistant mobile authentication scheme based on chess. PassGame can offer extremely high shoulder-surfing resistance, even against camera attacks, at some cost to usability. PassGame works by challenging a user with a random formation of chess pieces on a game board; successful authentication requires the user to alter the board so that a set of predefined rules are satisfied. We implement PassGame on Android. Our user studies show that PassGame can achieve 100% recall rates one week after password setup. Our user studies on the shoulder-surfing resistance of PassGame show that weak PassGame passwords cannot be shoulder-surfed even after viewing 5 complete recorded password entries, and strong passwords are resilient even against camera attacks.**

Keywords–Shoulder Surfing; Challenge Response; Mobile; Graphical Password; Authentication

I. INTRODUCTION

A short, preliminary version of this work was published at ACHI 2017 [1].

Mobile devices- such as smartphones and tablets- are becoming increasingly popular because of their nearly ubiquitous Internet access through various communication capabilities such as WiFi and their numerous applications and games. While users are enjoying the benefits of ubiquitous computing enabled by mobile devices, they are also becoming more vulnerable to shoulder-surfing attacks. Consider a user on a crowded subway train: the user may want to check emails as there are a few stops before a destination. But, to check emails through a smartphone, the user has to unlock the screen with possibly several pairs of eyes watching the whole authentication process from behind. Since current authentication schemes on mobile devices are not designed to resist shoulder-surfing attacks [2], users of mobile devices are in danger of password theft and its consequences. Harbach et al. [3] suggest that mobile phone users unlock their devices an average of 48 times per day (about 3 unlocks per hour), and users perceive shoulder-surfing to be possible in 17% of these instances.

Designing an authentication scheme for mobile devices is a challenging task because the scheme should be both *secure* and *usable*. For mobile devices, a secure authentication scheme should be shoulder-surfing resistant for ubiquitous computing and the scheme should have a large password space, i.e., a large number of possible passwords. Usability of an authentication scheme is of the same importance for mobile devices: (1) The scheme should be easy to use, (2) Passwords generated by the scheme should be easy to remember.

In this paper, we are concerned primarily with knowledge-based passwords, not biometric methods such as fingerprint scanning and facial recognition. At this time, biometric authentication on Android and iOS is always backed by a knowledge-based fallback authentication scheme. Furthermore, biometric schemes face unique security and usability challenges that are outside the scope of this paper.

In this paper, we propose *PassGame*, a shoulder-surfing resistant mobile authentication scheme based on board games. PassGame is essentially a challenge-response authentication scheme. In our current design, PassGame is based on the popular game of chess. Authentication starts with a random chess board, i.e., a chess board with randomly selected game pieces on randomly selected tiles of a game board. The random chess board serves as a challenge to the user. To finish authentication successfully, the user responds to the challenge by making adjustments to the random game board so that a set of predefined rules are satisfied. The adjustments can be moving game pieces, adding new game pieces, and removing existing game pieces.

PassGame supports both rules without any requirements on chess knowledge and rules requiring only basic chess knowledge. The design consideration is to make sure every user, including those who have no knowledge of chess, can use the authentication scheme. The latter rules require only basic chess knowledge, more exactly, the knowledge of how game pieces attack. We include these rules requiring basic knowledge of chess to take advantage of the popularity of the game because we hypothesize that chess knowledge or previous experiences in chess games may improve memorability of PassGame passwords.

We hope that gamifying our scheme can make authentication better in learning, user experience, and user behavior. Hamari et al. [4] and Kroeze et al. [5] assert that gamification can lead to positive effects in learning and user behavior, and that improvements in user behavior can make the scheme more secure. We anticipate that gamifying our scheme will offset some of the usability costs associated with challenge-response authentication. Chess players are trained to analyze the game board and move pieces quickly, as moving quickly is part

of normal game etiquette. In other words, chess players may already be trained to solve the challenge quickly.

Our contributions can be summarized as follows: (1) PassGame is designed to counter shoulder-surfing attacks. Our security analysis based on information theory shows that the scheme is better in shoulder-surfing resistance than previous schemes. PassGame also has a large password space to counter brute force attacks. (2) We implemented the PassGame design on the Android operating system. Our user studies with the implementation show that PassGame passwords generated with two rules can achieve 100% recall one week after setting the passwords. PassGame passwords generated with two rules already have a larger password space than 4-digit PIN, an authentication scheme widely used on mobile devices.

In general, shoulder-surfing resistant schemes incur relatively higher usability costs such as longer password entry time. We believe PassGame can be used as a shoulder-surfing resistant option for accessing high security features of the device or for authentication in public places. A user may want to access their phone when on a bus or subway, in plain view of strangers and potentially camera surveillance. There is always an intelligence cost (and thus a tradeoff for usability) when using a challenge-response scheme, so we think PassGame will be best suited as a supplementary security scheme. The user may rely on their simplier scheme when alone or for data with low security importance, and authenticate with PassGame for high security data or when in public. In a high risk environment, users may be willing to pay the usability cost.

PassGame is not designed to replace existing mobile authentication schemes, such as Google's pattern unlock and the four-digit PIN widely used on smartphones. Instead PassGame can be a supplemental scheme for use in crowded places or places with camera surveillance. PassGame can also be a choice for high security authentications on smartphone operating systems supporting different security levels in authentication such as Android.

This paper has been extended from its original version [1] in several ways: we have added a section to address our threat model (Section III), added a section to explain the functionality of our Android implementation (Section VI), added a section on security analysis- including a theoretical framework for measuring shoulder-surfing resistance and an analysis of the lower bound password space of PassGame (Section V), added analysis of user choice in PassGame (Section VII), added a new shoulder-surfing user study (Section VII), and finally extended our discussion and conclusion to address some plans for future work and to cover the new material above (Sections VIII and IX).

The rest of the paper is organized as follows: We review related work on graphical passwords and shoulder-surfing resistant authentication schemes in Section II. We introduce the threat model considered in this paper in Section III. Then, we present the design details of PassGame in Section IV. We analyze the security and theoretical shoulder surfing resistance of the scheme in Section V. We present our user studies on the usability and memorability of PassGame in Section VII. We conclude the paper in Section IX.

II. Related Work

PassGame, like most existing authentication schemes for mobile devices, can be classified as a graphical password scheme. Graphical password schemes rely on the "pictorial superiority effect" [6], the concept that humans have a much better memory for images than they do for numbers and letters, to increase memorability. Since Blonder's pioneer work [7], researchers have proposed various graphical password schemes [8], [9], [10].

Two graphical password schemes are widely available for commercial use on mobile devices. Google's pattern unlock scheme allows users to form a password by connecting dots arranged in a three by three grid (in newer versions of the Android OS, a larger grid can be used). The scheme has high usability as authentication can be finished with one long gesture. The cost of the advantage in usability is its relatively small password space [11]. Microsoft's picture password requires users to form a password by drawing gestures on top of an image that they select. A circle, line, or single touch are considered a single gesture. The gesture direction and location are recorded as a picture password. In addition to graphical password schemes for mobile authentication, 4-digit PIN and alphanumeric authentication are still available in both Android and iOS. While biometric schemes like fingerprint scanning are also available on some devices, these schemes always require a fallback password, typically a PIN. All current authentication schemes on mobile devices, including the pattern unlock scheme, the picture password scheme, and the 4-digit PIN, are vulnerable to shoulder-surfing attacks.

Research suggests that users are aware of the vulnerability of graphical schemes to observation, and perceive a greater risk of having their password observed in a graphical scheme versus a conventional keyboard based scheme [12]. Previous research also suggests that many graphical password schemes are more vulnerable to shoulder-surfing attacks than text-based password entry [2], [12].

A number of research efforts have been aimed to add shoulder-surfing resistance into existing schemes. Roth et al. [13] proposed to add shoulder-surfing resistance to the classic 4-digit PIN by splitting the PIN entry pad into two sets (black and white buttons) and asking users to choose which set their digit is in. The process is repeated several times to confirm the choice of a digit and repeats again until all the digits are chosen. Since then many schemes to add shoulder-surfing resistance to the 4-digit PIN have been proposed, including SwiPIN [14], ColorPIN [15], and The Phone Lock [16]. While these schemes can improve shoulder-surfing resistance of PIN-based schemes, they still suffer from inherently weak security strength of PINs and these schemes can be easily compromised by brute force attacks.

Zakaria et al. [17] proposed to improve the shoulder-surfing resistance of Draw a Secret [8] by erasing strokes as they are drawn. Their user study shows the improvement can reduce the rate of medium-strength passwords captured by an attacker after a single observation from 80% to roughly 40%. Lin et al. [18] proposed to add a grid to Draw A Secret. In addition to matching the Draw a Secret gesture, users in this scheme must

also match the direction (e.g., up, down, left, right) in which some strokes of their gesture pass through the added grid lines. Their user study reports that 0 of 10 participants were able to shoulder surf the password after one viewing, as opposed to 7 out of 10 for plain Draw a Secret, but memorability of the scheme was impacted significantly.

Convex Hull Click (CHC) [19] is a graphical password scheme designed to counter shoulder-surfing attacks. CHC asks users to choose icons to represent their passwords. Rather than clicking the icons, users are required to click somewhere inside the triangular area bounded by their chosen icons. CHC suffers from long authentication times because multiple click sessions are required and it takes time for the user to find their icons. The CDS scheme [20], a combination of Draw a Secret [8] and Story [21], arranges a series of images randomly into a grid and asks users to draw a line through the images they choose to represent their passwords. The shoulder-surfing resistance of CDS depends largely on the behavior of the user and how many images an attacker can remember.

PicassoPass [22] asks users to choose individual elements from several layers, such as letter, color, or shape, which must then be tapped in order. The layers are superimposed over each other during authentication, so when a user taps a location, the attacker cannot tell which layer was part of the user's password. Zero out of 22 participants were able to successfully shoulder surf a PicassoPass password after a single viewing, but no usability study is available for comparison.

PassGame can be considered a multi-dimensional password, as proposed in [23]. PassGame uses many dimensions such as rule, color, piece type, and number of attacking pieces.

A. Hardware-based Schemes

Some approaches to mitigating shoulder-surfing propose to add hardware to the device. Adding hardware can be problematic because of additional incurred production costs, additional points of failure in the device, and additional software requirements. Back-of-Device Shapes (BoD Shapes) [24] has users authenticate by using additional touch hardware at the back of the device. A shoulder-surfer would need to look up from the floor in order to see password entry. Glass Unlock [25] puts the authentication image on the user's private near eye display (e.g., Google Glass), using the touchscreen only as a nearly blank input device. EyePassword [26] reduces shoulder-surfing by gaze-based password entry. Eye-tracking software and hardware are used to track a user's gaze on screen to input sensitive information through an on-screen keyboard. A shoulder-surfer would need to see the orientation of the user's eyes to have enough information to crack the password. A gaze-based method may not be suitable for mobile authentication because of much smaller screens on mobile devices and the requirement for additional hardware such as a high-resolution front-facing camera and IR illumination.

Bianchi et al. [16] propose to use audio cues for authentication, but audio is not always available to a user when in a public place, for example in a movie theater. De Luca et al. [27] propose VibraPass, a shoulder-surfing resistant scheme for bank terminals that uses vibration cues from a mobile phone, which relies on access to a mobile phone with vibration enabled. Biometric schemes such as facial recognition and fingerprint scanning are immune to shoulder-surfing attacks, but they are vulnerable to theft of biometric data. Chaos Computer Club defeated the Apple iPhone 5s fingerprint scanner within 48 hours of its release, using only a photograph of the fingerprint from a glass surface [28]. PassGame does not require extra hardware and it does not rely on biometric data.

B. Gamification

Hamari et al. [4] demonstrate that gamification generally produces positive effects in learning, user experience, and user behavior. We hypothesize that certain good behaviors from chess will carry over to PassGame. For example, common etiquette in chess is for players to analyze the board and make their moves quickly, which may encourage users to enter their passwords quickly, especially when first learning the scheme. Kroeze et al. [5] speculate that adding game elements to authentication can improve user behavior and make them more secure. We attempt to base PassGame on a game that most people are able to play. We hypothesize that increasing familiarity will improve both memorability and usability for many users.

III. Threat Model

In this paper, we consider three different threat models:

1) An observer watching over the victim's shoulder for a small amount of time, long enough to observe a small number of successful entries. This is by far the most common threat, although it can carry relatively little severity. As Harbach et al. [3] note, many users are aware of aware of threats from curious attackers such as friends, acquaintances, and children- all of which can have frequent line of sight access to the password entry. A password without shoulder-surfing resistance, such as PIN, can easily be cracked with a single clear view of the password entry.

2) An observer watching over the victim's shoulder for a longer period of time, observing many successful entries. In this case, the attack is likely premeditated. Shi et al. [29] demonstrate that in general, viewing multiple entries of a shoulder-surfing resistant password significantly increases the probability of cracking it.

3) An observer who records the victim entering the password via camera or other means, allowing infinite reviewing of recorded entries. With any scheme based on information, the password can eventually be determined if sufficient entries are recorded and the intersection between them is analyzed. In general, it is useful to know how many entries are necessary to crack a password with intersection, with typical values at 2-3 entries [30].

In all three cases, we assume the observer has the opportunity to watch one or several password entries by following the victim and observing them. We assume the observer is familiar with the scheme. We also assume that the observer is able to completely see the screen with no obstructions. Once the observer is confident in their ability to bypass the victim's

authentication, they may steal the device or otherwise access it without the user knowing.

We assume the observer is not able to access data on the device by any means other than authenticating themselves as the user, due to some encryption on the device.

IV. The PassGame Design

In this section, we first present an overview of PassGame and describe the design details of PassGame.

A. Overview

The current design of PassGame is based on the popular game chess. PassGame is essentially a challenge-response authentication scheme. In PassGame, a mobile device challenges a user with a randomly generated chess board, i.e., a chess board with randomly selected game pieces placed on randomly selected tiles. The user responds to the challenge by making adjustments on the chess game board including adding new games pieces, removing existing game pieces, and moving existing game pieces. A correct response will be an adjusted game board satisfying some predefined rules. For example, one rule of PassGame is to move game pieces by n_{tile} tiles in total. Any move of a game piece, including moves that would be illegal in chess, are allowed. Moving a game piece to the right or the left by one tile increases or decreases one tile from the total. Similarly, moving a game piece up or down by one row increases or decreases eight tiles from the total, as one row on the board has 8 tiles. A user can increase or decrease the number of tiles moved by adding a new game piece to the board or removing a game piece from the board. As long as the sum total of tiles moved is equal to n_{tile}, the rule is satisfied and the user will be authenticated (if no other rules are in use). Otherwise, the authentication is unsuccessful.

PassGame supports both rules that do not require knowledge of how to play chess and rules requiring basic chess knowledge. The design is to make sure every user, including those who have no knowledge of chess, can use the authentication scheme. The other rules require only basic chess knowledge of how game pieces attack. We include these rules requiring basic knowledge of chess to take advantage of the popularity of chess because we hypothesize that chess knowledge or previous experiences in chess games may improve memorability of PassGame passwords.

A PassGame password can be formed with multiple rules. In general, using more rules to form a PassGame password can make the PassGame password more complex, and in turn more resistant to brute force attacks and shoulder-surfing attacks.

As long as the rules of a password are satisfied, PassGame allows users to make unrelated adjustments to the board. In other words, a user can add, remove, and move game pieces that are not involved in any rules used to form the password. These unrelated adjustments to a game board allow a user to further mitigate shoulder-surfing attacks as a shoulder-surfer can not tell which adjustments are involved in the rules used to form the PassGame password.

To make PassGame more usable, the design does not enforce the rules of chess. Any piece of either color can be positioned on any tile of the chess board, and multiple pieces of the same type are permitted (e.g., three kings). Any piece can move to any tile. However, some rules utilize the attack patterns of different pieces, for example by counting the number of attacks possible on a piece. In Chess, an attack on a piece can be removed by getting rid of the attacking piece, moving the defending piece, or blocking line-of-attack between the two pieces (except for knights), meaning there are many ways to add or remove attacks on a Chess board.

In the rest of this section, we describe the generation of a random game board and then the details of each rule possibly used in a PassGame password.

B. Random Board Generation

Since PassGame authentication starts with a challenge of a random board, the generation of the random board is important for both the security and usability of PassGame. On each tile, there are 13 possibilities: the tile is empty, or it is occupied by a king, queen, bishop, knight, rook, or pawn in either black or white.

PassGame randomly selects one from the 13 possibilities for each tile. Pieces appear with the same frequency as they typically appear in midgame chess. That is, empty tiles are most common, pawns are more common than knights, bishops and rooks, and kings and queens occur least frequently. Because the board is randomly generated, it is also possible to get boards which are almost completely empty or completely full. The design is to ensure most boards have enough pieces so that there are many ways to satisfy the rules of a PassGame password, and that many different kinds of PassGame passwords will be satisfiable within any sample of a few random boards.

We allow a user to request a new random board at any time during authentication. A user may request a random board for several possible reasons: (1) The user's password cannot be completed on the given random board (e.g., remove 3 black pieces from the board on a board with less than 3 black pieces), (2) The user wants a board where the password can be input more easily, (3) The user wants to find a game board where shoulder-surfing is less likely, or (4) The user has modified the random board unsuccessfully and does not remember what it initially looked like. A random board sometimes partially or completely satisfies some of a user's rules without any modifications. Thus, a shoulder-surfer may not necessarily see the user inputting all the rules that comprise the user's password, forcing them to guess remaining rules from the contents of the random board.

C. PassGame Rules

In our current design, a PassGame password can be formed with 12 rules. We present the details of the rules below. Users can, and should, pick multiple rules at the same time. In general, rules ask users to pick numerical values, locations, piece types, or color. When choosing color, a user can choose not to pick a color and instead answer "either", meaning the rule can be satisfied with a combination of black and white pieces.

To better understand the effective password space, at the end of our user study, we asked participants to tell us what was the maximum number of pieces they would use for each rule in practice. We present the average response along with the description of each rule.

The first 6 rules do not require any chess knowledge. So, any user should be able to use these rules.

Rule R1: Number of Tiles Moved in Total: The parameter of this rule is the number of tiles moved. To satisfy this rule, a user must make adjustments to a game board so that the number of tiles moved in total should be equal to a predefined number n_{tile}. The board can be considered as a numbered grid from 1 to 64, where the bottom left corner is 1, and the top right is 64. Moving a game piece to the right or to the left by one tile adds or decreases the number of tiles moved in total by one respectively. Similarly, moving a game piece up or down by one row adds or decreases the number of tiles moved in total by 8 respectively. Adding a game piece to a tile adds to the number of tiles moved in total by the number associated with that tile. On the contrary, removing a game piece from a tile decreases the number of tiles moved in total by the number associated with that tile.

For example, if a user sets $n_{tile} = 8$ in the password setup phase, the user can satisfy this rule by adding a piece to tile 8 if the tile is not occupied, or by moving a piece on tile 12 to tile 20 if the destination tile is not occupied. To mitigate shoulder-surfing attacks, a user can also combine multiple adjustments together to achieve the number of tiles in total. For example, if $n_{tile} = 8$, a user can move one piece forward by 20 tiles, move another piece backwards by 10 tiles, add a piece to tile 28, and remove a piece from tile 30 to make the number of total tiles moved be 8. In theory, the range of n_{tile} is $[-2080, 2080]$ as $\sum_{i=1}^{64} i = 2080$.

In practice, according to our user study, users would use a maximum of 17 tile moves for this rule.

Rule R2: Number of Pieces in a Row: The parameters of this rule are color, row index, and number of pieces of the selected color that must exist in the selected row. To satisfy this rule, a user must adjust a game board so that the selected row has the chosen number of pieces in it of the chosen color. This can be done adding pieces or removing pieces from the row, as a randomly generated row may have more pieces than are needed. The number of possible combinations of the parameters is $3 \times 8 \times 8 = 192$ as (1) color can be black, white, or either, and (2) a chess board has 8 rows and columns. According to our user study, users would use up to 5 pieces.

Rule R3: Number of Pieces in a Column: This rule is similar to Rule R2 and the only difference is that R3 is defined on a column. So the number of possible combinations of the parameters is also 192.

Rule R4: Number of Pieces on a Board: This rule is similar as Rule R2 and the only difference is that R4 is defined on a game board. The parameters of this rule are color and number of pieces on the board, so the number of possible combinations of the parameters is $3 \times 64 = 192$ as (1) color can be black, white, or either and (2) a board can hold up to 64 game pieces. According to our user study, a maximum of 22 pieces would be used in this rule.

Rule R5: More or Less Pieces: The parameters of this rule are color and the number of pieces added or removed from a board. To satisfy this rule, a user must add or remove the specified number of pieces in the chosen color. To further mitigate shoulder-surfing attacks, a user may want to add and remove pieces several times. As long as the final number of pieces added or removed from a board totals the specified number, the rule is satisfied. The number of possible combinations of the parameters is $3 \times 64 \times 2 = 384$ because (1) color can be black, white, or either, (2) at most 64 pieces can be added or removed from the board. According to our user study, users indicated they would use a maximum of 15 more pieces, and a maximum of 6 less pieces.

Rule R6: Specific Tile: The parameters of this rule are piece type, color, row index, and column index. The rule is satisfied when the specified piece of the chosen color is at the chosen row and column location. The number of possible combinations of the parameters is $6 \times 3 \times 8 \times 8 = 1152$ as (1) the piece type can be king, queen, bishop, knight, rook, or pawn, (2) the color can be black, white, or either color, and (3) the board has 8 rows and 8 columns. This rule is not shoulder-surfing resistant by itself. But, the rule can be used to form a shoulder-surfing resistant password by including unrelated adjustments such as placing random pieces onto randomly-selected tiles or simply being used with other rules.

The next 6 rules require only basic knowledge of attacks in chess. To add more attacks, a user can add game pieces under attack, attack existing pieces, or both. Attacks can also be added by removing pieces blocking attack paths of other game pieces. Similarly, attacks can be reduced by adding blocking pieces, removing attacking pieces, or removing the pieces under attack.

Rule R7: Number of Attacks on a Piece: The parameters of this rule are piece type, piece color, and number of attacks. This rule is satisfied when a game piece of the type and color selected is attacked by the chosen number of attackers. One example is that a bishop of either color is under attack by five pieces. If there is no such piece on a random board, a user can add it to the board. If there are multiple such pieces a board, then only one of them is required to be under attack by the specified number of pieces. The number of possible combinations of the parameters is approximately $6 \times 3 \times 16 = 288$ as (1) the piece type can be king, queen, bishop, knight, rook, or pawn, (2) the color can be black, white, or either color, and (3) the maximum number of attacks to one tile is 16 (4 diagonal attacks, 2 horizontal attacks, 2 vertical attacks, and 8 attacks by knights). Note that not every tile can have 16 attackers (e.g corner tiles can have a maximum of 5 attackers), so it may be necessary to move a piece or place a new one in order to satisfy larger numbers of attacks. Users indicated they would use a maximum of 4 attacking pieces.

Rule R8: Number of Attacks by Pieces: The parameters of this rule are piece type, piece color, and number of attacks. The rule is satisfied when a game piece of the selected type and color is attacking the chosen number of game pieces. For a king, a queen, or a knight, there are $3 \times 8 = 24$ combinations because (1) color can be black, white, or either and (2) a king, a queen, or a knight can attack a maximum of 8 pieces. For a bishop or a rook, there are $3 \times 4 = 12$ combinations because a bishop or a rook can attack 4 pieces at most. For a pawn, there are only $3 \times 2 = 6$ combinations because a pawn can only attack two pieces at most. So the total number of possible combinations is $3 \times 24 + 2 \times 12 + 6 = 102$. In our user study, users indicated they would use a maximum of 5 attacks.

Rule R9: Number of Pieces under Attack: The parameters of this are piece color and number of pieces under attack. The rule is satisfied when the selected number of game pieces of the chosen color are under attack. Since (1) the maximum number of attacks is 64 when a board is filled and every game piece is under attack, and (2) color can be black, white, or either, the number of possible combinations is $3 \times 64 = 192$. Users indicated they would use a maximum of 3 attacks.

Rule R10: More or Less Attacks on A Piece: The parameters of this rule are piece type, piece color, and number of attacks to add or remove. The rule is satisfied when the selected number of attacks are added or removed from a game piece of the chosen type and color. If there is no such piece on the board, a user can add it. As described in Rule R7, the maximum number of attacks on one tile is 16. Since (1) color can be black, white, or either and (2) the piece type can be king, queen, bishop, knight, rook, or pawn, the number of possible combinations is $3 \times 6 \times 32 = 576$. In our user studies, users indicated they would add a maximum of 4 attackers and remove a maximum of 2 attackers.

Rule R11: More or Less Attacks by A Piece: The parameters of this rule are piece type, piece color, and number of attacks to add. The rule is satisfied when the selected number of attacks are added or removed from a piece of the chosen color and type. A king, queen, or knight can attack 8 pieces at most. In other words, a user can select any of the 16 possible values between -8 and 8. The number of possible combinations for a king, queen, or knight is $3 \times 16 = 48$ since color can be black, white, or either. A bishop or rook can attack a maximum of 4 pieces, so the number of possible combinations for a bishop or a rook is $3 \times 8 = 24$. A pawn can attack up to 2 pieces, so the number of possible combinations for a pawn is $3 \times 4 = 12$. The total number of combinations is 204. Users indicated they would add a maximum of 4 attacks and remove a maximum of 2 attacks.

Rule R12: More or Less Pieces under Attack: The rule parameters are piece color and number of attacks to add or remove. This rule is satisfied when a user adds or removes the selected number of attacks to game pieces in the chosen color. A user can add or remove up to 64 attacks. The number of possible combinations of the parameters is $3 \times 128 = 384$ since color can be black, white, or either. In our user study, users indicated they would add up to 5 pieces under attack and remove up to 4 pieces.

D. Additional rules

PassGame supports only the rules above, however it is theoretically possible to come up with a near-infinite number of rules. For example, we can generate rules based on arbitrary criteria, for example "Knights which are 3 tiles left or right away from a bishop". We can also split existing rules into more detailed versions, for example "Knights in row 4", versus a more general rule such as rule 2, and similarly we can create less detailed rules such as "Pieces in rows 1-4". There is also more room for rules based on Chess, for example "Kings in check", and we can create rules which are boolean, for example "True/False there are no pieces in Row 3".

Increasing the number of available rules can make it more difficult for the attacker to iterate through all the rules and determine which are in use, potentially requiring them to obtain more password entries in order to make a successful guess. Furthermore, adding or varying rules in use can confound attackers who program tools to examine password entries, forcing them to constantly update these tools. However, including more rules may impact usability; users may feel overwhelmed when confronted with a list of hundreds of rules, even though reading through all of them is not strictly necessary as the user can simply pick a few arbitrarily.

V. Security Analysis

Our security analysis of PassGame focuses on shoulder-surfing resistance and password space. One of the major design goals is to mitigate shoulder-surfing attacks. We propose an information-theoretical measure of shoulder-surfing resistance and compare PassGame to other shoulder-surfing resistant schemes with the measure. An authentication scheme also needs a large password space to defeat brute force attacks by significantly increasing the cost of brute force attacks.

A. Shoulder-Surfing Resistance

The security of shoulder-surfing resistant schemes relies on the mapping between challenges and responses. If we denote the challenges and responses as C and R respectively, the mapping is $M : C \rightarrow R$, and M associates challenges with their valid responses. So, M is essentially the secret that a user has to memorize for authentication. A shoulder-surfer is able to observe a number of challenges and their corresponding valid responses. Based on the observation, the shoulder-surfer attempts to recover M so that the shoulder-surfer can break in by applying M to a future challenge given by a shoulder-surfing resistant scheme. So the dependency between challenges and responses indicates how a scheme is resistant to shoulder-surfing. A scheme with valid responses highly dependent on a challenge obviously is very vulnerable to shoulder-surfing attacks.

To reduce dependency, most shoulder-surfing resistant schemes mitigate shoulder-surfing attacks by allowing multiple responses to satisfy one challenge. For example, in CHC [19],

a user can click any place within a convex hull formed by preselected pass-icons for a correct response. Similarly in Rule R1 of PassGame, if $n_{tile} = 10$, a user can satisfy this rule in many ways. A user can simply move 10 existing game pieces to the right by one tile, add a piece to tile 10 if not occupied, or a combination of right and left moves of existing pieces, piece additions, and piece removals as long as the total number of tiles moved is 10.

The dependency can be measured by mutual information, an information-theoretical measure of dependency between two random variables. For a shoulder-surfing resistant scheme, the dependency can be represented by $I(C; R)$, meaning the mutual information between challenge C and response R. According to information theory, $I(C; R) = H(C) - H(C|R)$ where $H(C)$ denotes the entropy of the possible challenges and $H(C|R)$ denotes the conditional entropy of challenge C given response R. If a scheme generates challenges with a uniform distribution, the entropy $H(C)$ is a constant dependent on the number of possible challenges. So, to reduce the mutual information $I(X; Y)$, i.e., the dependency between challenges and responses, we need to increase $H(C|R)$. Since the conditional entropy $H(C|R)$ measures the uncertainty of challenges given a response, it is better to make a response to be valid to as many challenges as possible to reduce the dependency.

PassGame is designed to reduce the dependency in this way. PassGame allows a user to make adjustments that are unrelated to rules used to form a PassGame password. The adjustments can be moving existing pieces, adding new pieces, and removing existing pieces. These unrelated adjustments make the corresponding response valid to other challenges as well. So the unrelated adjustments can further reduce dependency and in turn make PassGame more resistant to shoulder-surfing. According to our knowledge, PassGame is the first attempt to include unrelated adjustments to an authentication scheme for mitigating shoulder-surfing attacks. We do not quantitatively compare PassGame with other shoulder-surfing resistance schemes according to the metric $I(C; R)$ as $H(C)$ depends on the number of possible challenges and the number can be very different for different shoulder-surfing resistant schemes. A fair comparison with the information-theoretical metric will be one of our future tasks.

B. Password Space

A PassGame password can be formed with the 12 rules described in the previous section. If only one rule is used, the number of possible passwords is essentially the sum of the possible combinations of parameters in each rule. So the number of possible one-rule passwords is 5938. Among the 5938 one-rule passwords, some will not be frequently used. For example, in Rule R1, the number of tiles moved in total can be up to 2080. However, a password with $n_{tile} = 2080$ is not usable as it would require 64 gestures and a completely empty random board to satisfy. So, we also calculate the *usable password space* of PassGame based on data from our user study. We asked participants in our user studies to tell us the maximum number of pieces they would use for each feature (e.g., what is the max number of pieces in a row you would use if you picked this rule). We then took the average of these values to calculate the parameter ranges. Using the responses provided by our users, we obtain the usable range of the parameters of each rule and then calculate the usable password space to be 1931.

The size of the one-rule password space can be enlarged since more rules can be added to the current design of PassGame. For example, we can add rules like number of pieces which are two tiles apart. In theory, one-rule PassGame can have a large password space to counter brute force attacks.

The password space can be enlarged exponentially when a combination of rules are used to form a PassGame password. The number of two-rule passwords is approximately $5938^2 = 35,259,844$. But, there are certain impossible passwords included in the calculation. For example, we cannot form a password by using R2 and R4 if R2 requires more pieces in a row than R4 requires on the board. So, to calculate the lower bound on the password space, we remove rules or portions of rules that can cause contradictions. Omitting these potentially contradictory features, we find the two-rule password space is $5,585,124$ passwords. For comparison, a 4-digit PIN has a password space of $10^4 = 10,000$, and Android pattern unlock has a total password space of 389,112 when using a 3x3 grid [31].

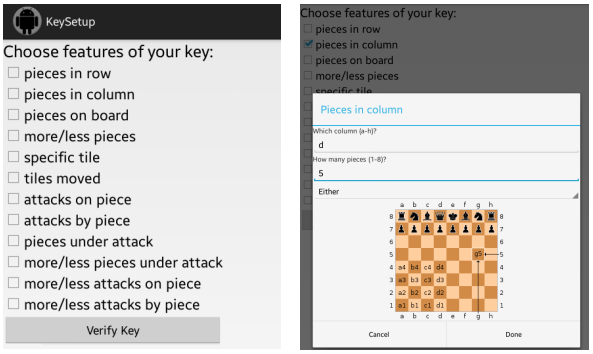
The size of the usable password space for two rules, based on responses from our users, is approximately $1931^2 = 3,728,761$ passwords. With four rules, PassGame reaches a password space of over $1931^4 = 10^{13}$, approximately the strength of an 8 character alphanumeric password without symbols. If we only include rules that can not cause conflicts in the calculation, the lower bound of the usable two-rule password space is $3,119,262$. With four rules, the lower bound is still over 10^{12}. The lower bounds calculated above are not tight lower bounds, but we calculate them to show that two rule PassGame passwords already have a password space much larger than current mobile authentications schemes.

VI. Implementation

We implemented PassGame on the Android operating system. A screenshot of the implementation is shown in Figure 2. The implementation allows a user to set a PassGame password with the rules described in the previous section. When setting a password, a user is required to select rules and set the corresponding rule parameters. The user is also required to verify a new password on a game board before finishing the password setup. The verification asks the user to confirm the password. During an authentication, a user is shown the graphical user interface as in Figure 2. A user can request a new random board by tapping the "New Board" button. There is no penalty for requesting a new random board.

A. Rule Selection

During password setup, the user selects rules from a checklist. When a box is checked, a prompt appears to ask for details, for example the color, number, type, or location of pieces involved in the rule. The prompt also provides brief



(a) Checkbox selection (b) Rule selection prompt

Figure 1. Screenshots of the password setup phase.

Figure 2. A screenshot of the PassGame application.

hints and helpful information for using the rule, for example a diagram indicating how rows and columns are labeled in chess. Figure 1 shows the password setup phase. In Figure 1(a), the user chooses from the list of rules, presented as check boxes, tapping *pieces in column* as one of their selections. The prompt in Figure 1(b) appears, asking the user to specify which column (free typing with the soft keyboard), how many pieces (free typing using the numerical soft keyboard), and which color (a drop down with black, white, and either as options). Columns are typically labeled a-h, and a maximum of 8 pieces can occupy a column. Basic hints are provided for most rules. Here, a hint figure shows how columns in chess are labeled, and hint text tells the user to use letters a-h and numbers 0-8 in their input.

When a user finishes setting the password, the user is taken to a blank board as a final sanity check against redundant passwords. An example of a redundant password could be a password that asks for 5 white pieces in row 2 but only 3 white pieces on the board. At this stage the user may also decide the password is too hard to enter, e.g., has low usability, and go back to make changes. The user has to complete the password on the blank board to finish setup. The participant can view the password during this step at any time time by pressing a "show password" button. Rules that would require removing pieces from a random board are omitted during this phase. Once the password is set, the user authenticates by entering the password on a randomly generated game board rather than a blank one.

VII. User Study

We implemented PassGame on the Android operating system. A screenshot of the implementation is shown in Figure 2. To evaluate PassGame, we conducted user studies with participants recruited from two university communities. We used a Samsumg Galaxy Tab 3 with a 7 inch 1024×600 display and the Samsung S4 with a 5 inch 1920×1080 display.

Procedure: On the first day, participants are invited to come to our controlled laboratory environment to fill out demographic information and learn how to generate a PassGame password. Participants are shown a fifteen-minute series of videos that covers the basics of PassGame and shows them how to use all the available rules. Questions about the technicalities of PassGame or the different rules are encouraged, but most participants were able to use PassGame with little to no further guidance. After learning how to use the scheme, participants are asked to generate their own PassGame passwords using one of the mobile devices. Before they leave the laboratory, participants must successfully authenticate themselves twice on two different random boards.

Similar to previous studies [32], we asked participants to use PassGame during the one-week-long user study to simulate regular use of the authentication scheme. We sent an email to participants 3-4 days after the first session then again 5-6 days after the first session. The email contains a link to an emulated version of the PassGame application hosted on the internet. The emulated version uses the same code and behaves in the same way as the version that participants used during the first session, and can be completed on any internet accessible device including a PC. We use an emulated version rather than asking participants to return to the laboratory to use the device because it is more convenient for participants and this portion of the experiment is designed solely to simulate regular use of the scheme in order to stimulate memorability. Use of the emulator is encouraged but not mandatory because (1) email responses are not reliable because of various reasons such as junk mail filtering, (2) we want to investigate the effect of regular use on the memorability of PassGame. Each participant had at most two successful authentications on the emulator and the attempts on the emulator happened within 36 hours from the sending time of the reminder emails.

One week after the first session, participants are invited back to the controlled laboratory environment for the second

session. Participants are given the mobile device that they used during the first session and are asked to recall their passwords. If a participant fails to recall his or her password, the participant may try as many times as they would like for up to five minutes. At the end of the second session, participants are asked to fill out a survey rating the usability of PassGame and their favorite mobile authentication scheme.

Conditions: To evaluate the usability of PassGame with different security strengths, participants were randomly grouped into one of three conditions: (1) 1R: Participants in this condition were asked to make a password using a single rule. They were not allowed to use Rule R6 because it is not shoulder surfing resistant on its own, but otherwise had no limitations on which rules they could select. (2) 2R: Participants in this condition were asked to make a password with two rules. (3) 4R: Participants in this condition were asked to make a password with four rules.

We limit the experiment to 4 rules because in practice, we found that passwords with 5 rules or more were too difficult to create and use. This is due to the difficulty in satisfying each rule individually without contradicting others. The task would not be difficult if the same, simple rule could be used multiple times, for example Rule 2, but we wanted to see the impact of choosing different rules. Rules such as 2, 3, and 5 can be difficult to satisfy simultaneously. In our future work, we plan to study PassGame with no limits on the number of rules that can be selected and no limits on repeating the same rule multiple times.

Participants: We recruited participants for the user studies by distributing fliers and leaflet style advertisements. A $10 cash incentive was offered for completing both sessions of the user study. Thirty seven participants were recruited for the user studies and 36 successfully finished both sessions. Of those who finished, 23 participants were male and 13 were female. There were 7 participants aged 20 or younger, 22 participants aged between 21 and 25, 4 participants aged 26-30, and 3 participants over the age of 30. Participants were asked "Are you skilled at using smartphones or mobile devices." On a scale from "Strongly Disagree" (1) to "Strongly Agree" (5), participants rated their skill an average of 4.28, with 32 rating their skill at 4 or higher.

Statistical Testing: We use a significance level of .05 for our hypothesis testing in this paper. For omnibus comparisons on categorical and quantitative data, we use Chi-squared and Kruskal-Wallis respectively. If the omnibus test is significant, we perform pairwise tests with Chi-squared for categorical data and Mann-Whitney for quantitative data.

A. Memorability Results

As a PassGame password formed with more rules requires more rule selections and rule parameters to be memorized, we hypothesize that the recall rate of PassGame passwords decreases when the number of rules used to form PassGame passwords increases.

The recall results of the user study are shown in Table I. The results show that none of our participants had any trouble in remembering 1R or 2R passwords. The recall rate of 4R passwords is 30% lower than the rates of 1R and 2R passwords, but most participants were still able to remember their 4R passwords as well. We perform an omnibus chi-squared test on the three conditions and find a significant difference between the memorability of the conditions ($\chi^2 = 8.51$, $p = .014$). The hypothesis is supported by the data of PassGame passwords formed by 4 or less rules. We believe that the statistical difference will become more significant when the number of rules used to form a PassGame password is larger. We restrict our user study on PassGame to passwords formed with no more than 4 rules because (1) a two-rule password already has more password strength than 4-digit PIN, and (2) PassGame passwords formed with more than 4 rules are less usable.

TABLE I. PASSGAME RECALL RATES BY CONDITION

Conditions	Participants	Recall	Recall Rate
$1R$	12	12	100%
$2R$	14	14	100%
$4R$	10	7	70%

We examine the effect of the reminder emails on memorability. We hypothesize that using the emulator during the week will make participants more likely to remember their passwords at the end of the week. Five participants used the emulator only after receiving the first reminder email, 2 used the emulator only after receiving the second reminder email, 24 used the emulator both times, and 5 did not use the emulator at all. The omnibus chi-squared test reveals no significance ($\chi^2 = 1.64$, $p = .651$). All three participants who forget their passwords used the emulator both times, and were unable to finish authentication successfully either time. The results suggest that PassGame passwords are memorable after one week even with no reminders.

We hypothesize that chess knowledge has an impact on memorability. Thirty-one participants indicated that they knew how to play chess, while 5 indicated they did not know how to play chess. Among the 3 participants that forgot their passwords, 2 knew how to play chess and 1 did not. Our omnibus chi-squared test reveals that there is no significant difference ($\chi^2 = 1.04$, $p = .309$). The results are not compliant with our expectation. But, the results also indicate that the scheme is memorable even by persons who have no knowledge of chess.

B. Password Entry Time

Our implementation records the time users spend attempting to enter their passwords. In this section, we analyze the timing data from the final session of the user study.

TABLE II. AVERAGE ENTRY TIMES, AVERAGE NEW BOARDS AND ATTEMPTS PER SUCCESSFULL AUTHENTICATION

Conditions	Total (s)	Correct (s)	Boards	Attempts
1R	33	23	1.6	1.22
2R	110	44	1.9	2.07
4R	143	49	2.1	2.63

Table I shows the average total entry time, average entry time for successful attempts, and average attempts per successfully attempt. Users in the 1R, 2R, and 4R conditions

required 33, 110, and 143 seconds respectively to authenticate themselves from the moment they started the application, including time spent thinking, requesting new boards, and making incorrect attempts. A Kruskal Wallis test between the three conditions finds no significant difference (H=4.996, p=.082). On average, users required 1.6, 1.9, and 2.1 new randomly generated boards for the 1R, 2R, and 4R conditions respectively before successfully entering their passwords. Additionally, users required an average of 1.22, 2.07, and 2.63 authentication attempts before a success for 1R, 2R, and 4R respectively. Correct attempts, measuring time from application load or the end of the previous unsuccessful attempt until the last touch in a successful attempt, required on average 23, 44, and 49 seconds for 1R, 2R, and 4R respectively. The best 4 users in 1R required less than 7s to authenticate themselves. We perform a Kruskal Wallis test on the timings for the first correct attempt and find that there is not a significant difference in the timings (H=3.741, p=.154).

We believe that these statistics will improve as users gain experience with the scheme. PassGame is effectively a short puzzle solving task. Once users become familiar with the puzzle, entry times should improve. Password entry times for a single correct attempt are already very similar between the conditions. The entry times for correct attempts are in line with other schemes such as Deja Vu (32s) [9], Delayed Oracle Choice PIN entry (25s) [13], or CDS (20s) [20] and superior to other shoulder-surfing resistant schemes like Convex Hull Click (72s) [19].

SwiPin [14], ColorPIN [15], The Phone Lock [16], and other schemes that improve on PIN or pattern unlock offer short login times, but at the cost of weak password strength and limited shoulder-surfing resistance. PassGame can be used as a supplementary high-security scheme in environments where the user is afraid of shoulder-surfing. The user may be willing to trade off entry time in exchange for security in these situations.

C. User Perception

TABLE III. USABILITY SURVEY RATINGS

Scheme	Ratings	Conve.	Speed
PassGame-1R	4	4.50	4.25
PassGame-2R	7	4.29	3.29
PassGame-4R	7	3.75	2.57
PassGame-all	18	4.06	3.22
4-digit PIN	10	5	5

At the end of the user study we asked participants to fill out a survey regarding the usability of PassGame and their current favorite authentication scheme. Participants were asked to rate the following statements (once for PassGame, and once for their favorite scheme) on a scale from "Strongly Disagree" (1) to "Strongly Agree" (5): (a) It is convenient to enter a password using this scheme. (b) The speed of entering a password with this scheme is fast. Additionally, we provide participants with the following definitions as a guideline: (a) Convenience: The scheme does not restrict you or take too much attention, (b) Speed: You can finish the scheme quickly. It usually does not need too many tries. For their favorite scheme, 10 participants chose 4-digit PIN, 2 participants chose Google's pattern unlock scheme, 3 chose fingerprint scanner. We sorted the usability results for PassGame based on which condition users were assigned to. The results of the usability survey are shown in Figure 3. The average usability rating is shown in Table III. For statistical analysis, we sort the usability ratings into the categories agree (4 or higher) or do not agree (3 or lower). We hypothesize that most users will think that PassGame is roughly as convenient as the 4-digit PIN or Google's pattern unlock scheme. We also hypothesize that the speed rating will decline as more rules are used. A chi-squared omnibus test on the three conditions of PassGame plus 4-digit PIN shows no significant difference in convenience ($\chi^2 = 4.11$, $p = .25$), however there is a significant different in speed ($\chi^2 = 11.04$, $p = .01$). Pairwise testing reveals the results are significant between 2R and 4-digit PIN ($\chi^2 = 7.47$, $p < .01$) and between 4R and 4-digit PIN ($\chi^2 = 10.12$, $p < .01$). At 2 rules and up, users perceive PassGame to be a slower scheme than the 4-digit PIN. We believe the difference is mainly caused by the shoulder-surfing resistance. A user usually repeats a 4-digit PIN without any thinking. But a user of shoulder-surfing resistant schemes needs to think out a valid response to a random challenge. Another possible reason is the difference in the familiarity to the scheme, as participants may be using 4-digit PINs on their mobile devices every day, and they only used PassGame a few times.

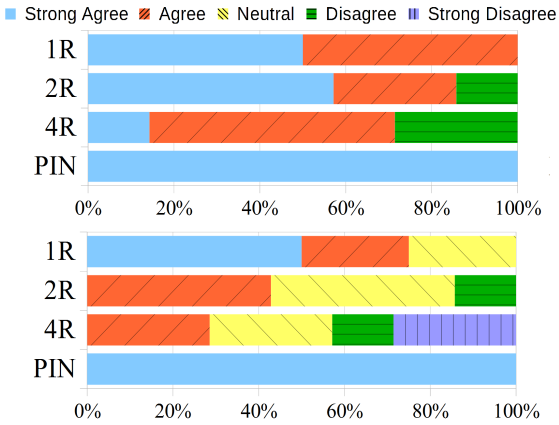


Figure 3. Usability Survey for Convenience (top), Speed (bottom).

D. User Choice in Passwords

We hypothesize that there will be hotspots in feature selection, e.g. that some features will be more common than others. Additionally. we believe that certain pieces, colors, and numbers will be more popular than others. Analyzing the data from our user study reveals several hotspots.

A total of 74 rules were selected by the 36 users in our study. A user can chose each rule only once, so the maximum number of times a rule could appear is 36 times. Figure 4 shows the number of times each rule was selected, demonstrating that

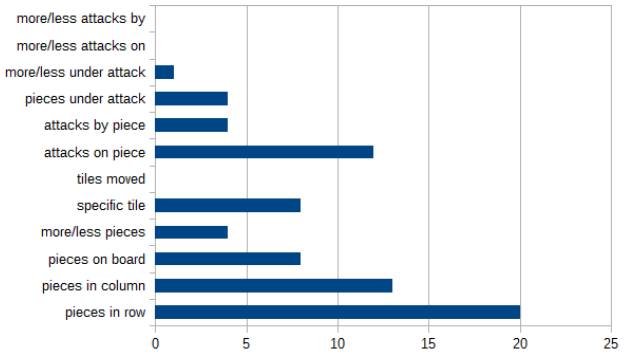


Figure 4. Number of times each feature was selected in our user study.

hotspots do exist in rule selection. For pieces in a row or column, the majority of users (85%) elected to use less than four pieces. When a piece was required to be chosen for some rule, e.g., for the specific tile rule or the attacks on piece rule, users chose the king (46%) and queen (29%) over the rook (13%), bishop (0%), knight (8%), and pawn (4%). We hypothesize that knowledge of chess leads users to prefer the most "powerful" piece, and plan to investigate the effect in games where pieces are equally balanced, such as Monopoly, in our future work.

E. Shoulder-Surfing User Study

TABLE IV. SUCCESSFULL SHOULDER-SURFING ATTEMPTS BY CATEGORY

Strength	1 Viewing	5 Viewings	Unlimited (¡1 hour)
Easy	0	5	15
Medium	0	0	3
Hard	0	0	0

We invited participants back after the first user study for a second user study on the shoulder-surfing resistance of PassGame. To ensure maximum consistency, we recorded the entry of three different PassGame passwords formed with 2 rules, 3 rules, and 4 rules. Only the 4-rule password was formed with rules requiring basic chess knowledge. Participants were told that the passwords were "easy," "medium," and "hard" respectively, that each password had between 2 and 4 rules, and that only the hard password involved chess knowledge. Five successful entries on five different random boards are recorded for each password. Participants are informed that there are no moves made during password entry that are not related to the password, i.e., no unrelated adjustments, every move made is significant to authentication and all authentication attempts are done in a natural and efficient manner.

Participants view the recordings on the same device that they used in the first user study, with no obstructions to their vision, simulating a worst case scenario for shoulder-surfing. Moves are displayed on the screen as a highly visible purple cursor that is transparent enough not to block vision of the board. Moves are executed at a relatively slower speed to allow better observation. Participants in this experiment have already been familiarized with PassGame, so only a brief recap of the rules is provided. As an additional aid, participants are provided with a sheet of paper listing all of the rules along with a brief description, and printouts of blank boards as scratch paper. If a participant thinks they have cracked the password, they can try it on the device with unlimited attempts, simulating a worst case scenario where attempts are not limited. A $100 prize pool is used to encourage participants to recover the PassGame password successfully. Participants who recover PassGame passwords successfully can split the prize, where participants who cracked the hardest password receive the majority of the pool.

Table IV shows our shoulder-surfing study results. Initially, we limit participants to a single viewing of each password entry as in [13] and [22], simulating a realistic shoulder-surfing attack by an observer. Note that in [22], participants were able to view only a single password entry, whereas we allow participants to view five. In [13], ten successful entries are shown. Zero out of fifteen participants were able to recover any of the passwords.

We investigate the effectiveness of PassGame against repeated observation, as in [33], by allowing participants 5 additional sequential viewings of each of the 5 password entries. Shi et al. [29] show that the probability of a shoulder surfer correctly guess the password in their scheme with just 2 recordings is rated at 20-25%. Chameleon [30] is considered secure against 3 or fewer captured login sessions. Our experiment allows for 5 recordings and unlimited attempts on the actual device, so the probability of a successful guess should be much higher. If an attacker has many recordings of a PassGame password, they can crack it by studying the intersection of information between the recorded entries. The number of recordings required and the probability to crack a password with a given number of recordings depends on how much intersection exists between recorded passwords. For example, an attacker could rule out the "tiles moved" rule by counting the number of tiles moved in several recordings. If the number of tiles moved does not match in just one successful authentication attempt, the attacker knows to discard this rule.

Participants were allowed to view all 5 entries an additional 5 times (a total of 6 including the previous experiment). Thus in total, participants witnessed 30 successful authentication attempts of each password. Entries were shown in sequence, that is participants saw all 5 entries, then were given time to think or take notes, then shown all 5 again. Participants chose for themselves when to move on to the next viewing, typically after a few seconds.

The easy password was shown first. After the additional viewings, 5 participants (33%) cracked the easy password with one attempt on the device.

All 15 participants moved on to the medium password. After the additional viewings, no participants were able to crack the medium password. Some participants were able to partially guess 1-2 rules (based on verbal confirmation), but none were able to crack the password entirely. We did not confirm or deny if users guessed any rules successfully during the experiment.

No participants were confident enough to opt to try inputting the password on the device.

Only 5 participants opted to try the hard password. All 5 failed to crack the password after the additional viewings. Several participants described it as "impossible" and that they felt "nobody would be able to get that."

Lastly, we allowed participants unlimited viewing of the recordings, including pause, fast forward, and rewind, along with unlimited guessing attempts on the device. Participants were also allowed to work in teams if they wished, and about 3 groups of 2 were formed. This is to simulate a worst-case situation when the attacker has captured recordings of multiple passwords, and they have considerable time and energy on their hands. All 15 participants were able to guess the easy password in this manner, but none were able to guess the medium or hard passwords after 20 minutes each (as previously, only the same 5 participants opted to attempt the hard password). Some participants opted to keep trying, and 3 participants (2 of which were grouped as a team) were able to crack the medium password after an average of 40 minutes. None were able to crack the hard password in under 1 hour, though only 1 participant attempted the hard password beyond the 20 minute mark, with the rest agreeing that it was still too difficult.

Our study shows that even a rudimentary PassGame password has good protection against shoulder surfing, and a more complicated password can be highly resistant to shoulder surfing. With a single viewing of 5 complete successful password entries, even the simple password could not be shoulder-surfed. The hard passwords was resistant to unlimited viewings, simulating a worst-case camera attack. In our future work, we plan to develop a program to crack recorded PassGame passwords to determine how many entries are needed on average to generate enough intersection for a successful guess.

VIII. Discussion

In this section, we discuss extensions of PassGame and discuss the problem of challenge-response authentication in terms of usability.

A. Extension of PassGame with New Games

To foil an attacker who obtains the older password through various means such as password hash cracking, interception, or simply guessing, system owners or administrators prefer expiring old passwords every a few months or weeks and asking system users to generate new passwords. While password expiration policies can possibly help secure the system by reducing the time that an attacker has to access the system, password expiration policies can cause extra burden on system users such as interruption of ongoing work and increase in login errors. Zeng et al. [34] even reported that the knowledge of old passwords can help in breaking new passwords.

PassGame can be extended to reduce or eliminate the side effects of the password expiration policies. The extension is to add a game dimension to PassGame. In other words, when a user is required to change the old password based on one game, the user can select another game and form a new password based on the new game. To better reduce or eliminate the side effects, the systems may use games that are as different as possible. For example, if the old password is based on chess, the system may suggest the user to use Monopoly for the new password.

The game change can help reduce memory interference in long term memory, which is used for continuing storage of information [35], as the new game is completely different from the old game and the passwords formed based on the different games are less likely to cause memory interference.

The addition of the game dimension can also prevent breaking new passwords based on the knowledge of old passwords. PassGame based on different games may have different sets of dimensions so no relationship between the new password and old passwords is available to assist in cracking the new password. For example chess and Monopoly have different sets of game pieces/rules and completely different game boards.

We plan to perform a user study on the extension in our future work. Since passwords usually expire every 3 months or 6 months, the user study may take a long time.

B. Impact of Unrelated Adjustments

As we observed in our user study, shoulder-surfing based on a recording of multiple entries is conducted by attempting to find the intersection of information between recorded passwords. Information can be gathered from 1) the initial random board, 2) the user's adjustments, and 3) success or failure of the authentication attempt. The first and third options are effectively outside the user's control when they are being recorded. To make it more difficult to deduce their password, a user can raise the overall amount of information the attacker has to parse for intersections by making unrelated adjustments. To an attacker these adjustments can be considered noise. As the amount of unrelated adjustments rises, the likelihood of intersections found by an attacker to be noise (i.e. false positives) instead of part of the actual password increases. Thus even a user that knows they are being recorded can use the scheme with some degree of protection, by trading off some usability. The more usability traded off, e.g., the more unrelated adjustments made, the harder it will be for an attacker to extract useful information from the authentication attempt.

C. Cost of Shoulder-Surfing Resistance

In general, shoulder-surfing resistant schemes incur relatively higher usability costs such as longer password entry time, so PassGame is designed to be a supplemental scheme for use in crowded places or places with camera surveillance. Alternatively, a user can set one or two rule PassGame passwords for medium security on Android systems, and passwords with more rules for high security.

We recognize that an inevitable shortcoming of any challenge-response scheme is the requirement of *focus*. To assess the challenge and craft an appropriate response requires intelligence and concentration which may make the scheme less suitable for some situations when users may want to check their phones (e.g., when crossing the street). Sometimes the

tradeoff with usability will not be a big issue, such as when the user is sitting on a bus awaiting some destination. We believe that a scheme like PassGame would work best when used along side a faster and simpler scheme so the user can cater authentication to the situation the user is in. The user may authenticate themselves with the simpler scheme when alone or in a trusted area and defer to PassGame when in public or when accessing more sensitive data. Alternatively, the user may have a 1 or 2 feature PassGame password that is easy to input for low security, and additional features that need to be satisfied to access high security content.

IX. Conclusion

We designed PassGame to mitigate shoulder-surfing attacks on mobile authentication. We implemented PassGame on the Android operating system and conducted a user study on the memorability/usability of PassGame and the shoulder-surfing resistance of PassGame. Our user studies show that PassGame passwords, which greatly exceed the password strength of current mobile authentication schemes and feature robust shoulder-surfing resistance, can still achieve 100% recall rates when recalled one week after password setup. PassGame even offers some resistance against camera attacks.

Acknowledgment

This work was supported in part by the National Science Foundation under Grants CNS-1460897, CNS-1338105, CNS-1343141, and DGE-1623713. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] J. Gurary, Y. Zhu, N. Alnahash, and H. Fu, "Passgame: A shoulder-surfing resistant mobile authentication scheme," in *Advances in Computer-Human Interactions*, Mar. 2017.

[2] X. Suo, Y. Zhu, and G. Owen, "Graphical passwords: a survey," in *21st Annual Computer Security Applications Conference*, Dec. 2005, pp. 462–472.

[3] M. Harbach, E. von Zezschwitz, A. Fichtner, A. D. Luca, and M. Smith, "It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception," in *Symposium On Usable Privacy and Security*, 2014, pp. 213–230.

[4] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work?–a literature review of empirical studies on gamification," in *47th Hawaii International Conference on System Sciences (HICSS)*, 2014, pp. 3025–3034.

[5] C. Kroeze and M. S. Olivier, "Gamifying authentication," in *Information Security for South Africa (ISSA)*, 2012, pp. 1–8.

[6] D. L. Nelson, V. S. Reed, and J. R. Walling, "Pictorial superiority effect," *Journal of Experimental Psychology: Human Learning and Memory*, vol. 2, no. 5, pp. 523–528, Sep. 1976.

[7] G. Blonder, "Graphical password," Sep. 1996, patent 5,559,961.

[8] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *Proceedings of the 8th Conference on USENIX Security Symposium*, 1999, pp. 1–14.

[9] R. Dhamija and A. Perrig, "Deja vu: A user study using images for authentication," in *Proceedings of the 9th Conference on USENIX Security Symposium*, 2000, pp. 1–4.

[10] A. De Angeli, L. Coventry, G. Johnson, and K. Renaud, "Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems," *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 128–152, Jul. 2005.

[11] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: The case of android unlock patterns," in *Proceedings of the ACM SIGSAC Conference on Computer Communications Security*, 2013, pp. 161–172.

[12] F. Tari, A. A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proceedings of the Second Symposium on Usable Privacy and Security*, 2006, pp. 56–66.

[13] V. Roth, K. Richter, and R. Freidinger, "A pin-entry method resilient against shoulder surfing," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 2004, pp. 236–245.

[14] E. von Zezschwitz, A. De Luca, B. Brunkow, and H. Hussmann, "Swipin: Fast and secure pin-entry on smartphones," in *Proceedings of the Conference on Human Factors in Computing Systems*, vol. 15, 2015, pp. 1403–1406.

[15] A. De Luca, K. Hertzschuch, and H. Hussmann, "Colorpin: securing pin entry through indirect input," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 1103–1106.

[16] A. Bianchi, I. Oakley, V. Kostakos, and D. S. Kwon, "The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices," in *Proceedings of the 5th International Conference on Tangible, Embedded, and Embodied Interaction*, 2011, pp. 197–200.

[17] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proceedings of the 7th Symposium on Usable Privacy and Security*, 2011, pp. 1–12.

[18] D. Lin, P. Dunphy, P. Olivier, and J. Yan, "Graphical passwords & qualitative spatial relations," in *Proceedings of the 3rd Symposium on Usable Privacy and Security*, 2007, pp. 161–162.

[19] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, 2006, pp. 177–184.

[20] H. Gao, Z. Ren, X. Chang, X. Liu, and U. Aickelin, "A new graphical password scheme resistant to shoulder-surfing," in *International Conference on Cyberworlds (CW)*, 2010, pp. 194–199.

[21] D. Davis, F. Monrose, and M. K. Reiter, "On user choice in graphical password schemes," in *Proceedings of the 13th Conference on USENIX Security Symposium*, 2004, pp. 1–11.

[22] W. A. van Eekelen, J. van den Elst, and V.-J. Khan, "Picassopass: a password scheme using a dynamically layered combination of graphical elements," in *Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 1857–1862.

[23] J. Gurary, Y. Zhu, G. Corser, J. Oluoch, N. Alnahash, and H. Fu, "Maps: A multi-dimensional password scheme for mobile authentication," in *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, 2015, pp. 409–412.

[24] D. Luca *et al.*, "Back-of-device authentication on smartphones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2389–2398.

[25] C. Winkler *et al.*, "Glass unlock: Enhancing security of smartphone unlocking through leveraging a private near-eye display," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, vol. 15, 2015, pp. 1407–1410.

[26] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proceedings of the 3rd Symposium on Usable Privacy and Security*, 2007, pp. 13–19.

[27] A. De Luca, E. Von Zezschwitz, and H. Hußmann, "Vibrapass: secure authentication based on shared lies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 913–916.

[28] Frank, "Chaos computer club breaks apple touchid,"

http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid, Sep. 2013.

[29] P. Shi, B. Zhu, and A. Youssef, "A pin entry scheme resistant to recording-based shoulder-surfing," in *Third International Conference on Emerging Security Information, Systems and Technologies (Securware)*, 2009, pp. 237–241.

[30] W.-C. Ku, D.-M. Liao, C.-J. Chang, and P.-J. Qiu, "An enhanced capture attacks resistant text-based graphical password scheme," in *International Conference on Communications in China (ICCC)*, 2014, pp. 204–208.

[31] T. Kwon and S. Na, "Tinylock: Affordable defense against smudge attacks on smartphone pattern lock systems," *Computers & Security*, vol. 42, pp. 137–150, 2014.

[32] N. Wright, A. S. Patrick, and R. Biddle, "Do you see your password?: Applying recognition to textual passwords," in *Proceedings of the 8th Symposium on Usable Privacy and Security*, 2012, pp. 1–14.

[33] T. Takada and M. Ishizuka, "Chameleon dial: repeated camera-recording attack resilient pin input scheme," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2015, pp. 365–368.

[34] Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: An algorithmic framework and empirical analysis," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010, pp. 176–186.

[35] R. Atkinson and R. Shiffrin, *The Psychology of Learning and Motivation*. New York: Academic Press, 1968, vol. 2.