On the Security of Picture Gesture Authentication

Ziming Zhao and Gail-Joon Ahn, *Arizona State University and GFS Technology, Inc.;*
Jeong-Jin Seo, *Arizona State University;* Hongxin Hu, *Delaware State University*

This paper is included in the Proceedings of the
22nd USENIX Security Symposium.
August 14–16, 2013 • Washington, D.C., USA
ISBN 978-1-931971-03-4

On the Security of Picture Gesture Authentication

Ziming Zhao[†‡] Gail-Joon Ahn[†‡] Jeong-Jin Seo[†] Hongxin Hu[§]
[†]*Arizona State University* [‡]*GFS Technology, Inc.* [§]*Delaware State University*
{*zzhao30,gahn,jseo15*}*@asu.edu* *hhu@desu.edu*

Abstract

Computing devices with touch-screens have experienced unprecedented growth in recent years. Such an evolutionary advance has been facilitated by various applications that are heavily relying on multi-touch gestures. In addition, picture gesture authentication has been recently introduced as an alternative login experience to text-based password on such devices. In particular, the new Microsoft Windows 8^{TM} operating system adopts such an alternative authentication to complement traditional text-based authentication. In this paper, we present an empirical analysis of picture gesture authentication on more than 10,000 picture passwords collected from over 800 subjects through online user studies. Based on the findings of our user studies, we also propose a novel attack framework that is capable of cracking passwords on previously unseen pictures in a picture gesture authentication system. Our approach is based on the concept of selection function that models users' password selection processes. Our evaluation results show the proposed approach could crack a considerable portion of collected picture passwords under different settings.

1 Introduction

Using text-based passwords that include alphanumerics and symbols on touch-screen devices is unwieldy and time-consuming due to small-sized screens and the absence of physical keyboards. Consequently, mobile operating systems, such as iOS and Android, integrate a numeric PIN and a draw pattern as alternative authentication schemes to provide user-friendly login services. However, the password spaces of these schemes are significantly smaller than text-based passwords, rendering them less secure and easy to break with some knowledge of device owners [8].

To bring a fast and fluid login experience on touch-screen devices, the Windows 8^{TM} operating system comes with a picture password authentication system, namely picture gesture authentication (PGA) [25], which is also an instance of background draw-a-secret (BDAS) schemes [18]. This new authentication mechanism hit the market with miscellaneous computing devices including personal computers and tablets. At the time of writing, over 60 million Windows 8^{TM} licenses have been sold [21] and it is estimated that 400 million computers and tablets will run Windows 8^{TM} with this newly introduced authentication scheme in one year [28]. Consequently, it is imperative to examine and explore potential attacks on picture gesture authentication in such a prevalent operating system for further understanding user experiences and enhancing this commercially popular picture password system.

Many graphical password schemes–including DAS [24], Face [9], Story [15], PassPoints [41] and BDAS [18]–have been proposed in the past decade (for more, please refer to [6, 7, 13, 14, 16, 23, 34, 37]). Amongst these schemes, click-based schemes, such as PassPoints, have attracted considerable attention and some research has analyzed the patterns and predictable characteristics shown in their passwords [12, 39]. Furthermore, harvesting characteristics from passwords of a target picture and exploiting hot-spots and geometric patterns on the target picture have been proven effective for attacking click-based schemes [17, 32, 38]. However, PGA allows complex gestures other than a simple click. Moreover, a new feature in PGA, autonomous picture selection by users, makes it unrealistic to harvest passwords from the target pictures for learning. In other words, the target picture is previously *unseen* to any attack models. All existing attack approaches lack a generic knowledge representation of user choice in password selection that should be abstracted from specific pictures. The absence of this abstraction makes existing attack approaches impossible or abysmal (if

All correspondences should be addressed to Dr. Gail-Joon Ahn at gahn@asu.edu.

possible) to work on previously unseen target pictures.

In this paper, we provide an empirical analysis of user choice in PGA based on real-world usage data, showing interesting findings on user choice in selecting background picture, gesture location, gesture order, and gesture type. In addition, we propose a new attack framework that represents and learns users' password selection patterns from training datasets and generates ranked password dictionaries for previously unseen target pictures. To achieve this, it is imperative to build generic knowledge of user choice from the abstraction of hotspots in pictures. The core of our framework is the concept of a selection function that simulates users' selection processes in choosing their picture passwords. Our approach is not coupled with any specific pictures. Hence, the generation of a ranked password list is then transformed into the generation of a ranked selection function list which is then executed on the target pictures. We present two algorithms for generating the selection function list: one algorithm is to appropriately develop an optimal guessing strategy for a large-scale training dataset and the other deals with the construction of high-quality dictionaries even when the size of the training dataset is small. We also discuss the implementation of our attack framework over PGA, and evaluate the efficacy of our proposed approach with the collected datasets.

The contributions of this paper are summarized as follows:

- We compile two datasets of PGA usage from user studies[2] and perform an empirical analysis on collected data to understand user choice in background picture, gesture location, gesture order, and gesture type;
- We introduce the concept of a selection function that abstracts and models users' selection processes when selecting their picture passwords. We demonstrate how selection functions can be automatically identified from training datasets; and
- We propose and implement a novel attack framework which could be potentially redesigned as a picture-password-strength meter for PGA. Our evaluation results show that our approach cracked 48.8% passwords for previously unseen pictures in one of our datasets and 24.0% in the other within fewer than 2^{19} guesses (the entire password space is $2^{30.1}$).

The rest of this paper is organized as follows. Section 2 gives an overview of picture gesture authentication. Section 3 discusses our empirical analysis on picture gesture authentication. In Section 4, we illustrate our attack framework. Section 5 presents the implementation details and evaluation results of the proposed attack framework. We discuss several research issues in Section 6 followed by the related work in Section 7. Section 8 concludes the paper.

2 Picture Gesture Authentication: An Overview

Like other login systems, Windows 8™ PGA has two independent phases, namely registration and authentication. In the registration stage, a user chooses a picture from his or her local storage as the background. PGA does not force users to choose pictures from a predefined repository. Even though users may choose pictures from common folders, such as the `Picture Library` folder in Windows 8™, the probability for different users to choose an identical picture as the background for their passwords is low. This phenomenon requires potential attack approaches to have the ability to perform attacks on previously unseen pictures. PGA then asks the user to draw exactly three gestures on the picture with his or her finger, mouse, stylus, or other input devices depending on the equipment he or she is using. A gesture could be viewed as the cursor movements between a pair of 'finger-down' and 'finger-up' events. PGA does not allow free-style gestures, but only accepts tap (indicating a location), line (connecting areas or highlighting paths), and circle (enclosing areas) [29]. If the user draws a free-style gesture, PGA will convert it to one of the three recognized gestures. For instance, a curve would be converted to a line and a triangle or oval will be stored as a circle. To record these gestures, PGA divides the longest dimension of the background image into 100 segments and the short dimension on the same scale to create a grid, then stores the coordinates of the gestures. The line and circle gestures are also associated with additional information such as directions of the finger movements.

Once a picture password is successfully registered, the user may login the system by drawing corresponding gestures instead of typing his or her text-based password. In other words, PGA first brings the background image on the screen that the user chose in the registration stage. Then, the user should reproduce the drawings he or she set up as his or her password. PGA compares the input gestures with the previously stored ones from the registration stage. The comparison is not strictly rigid but shows tolerance to some extent. If any of gesture type, ordering, or directionality is wrong, the authentication fails. When they are all correct, an operation is further taken to measure the distance between the input password and the stored one. For tapping, the gesture passes authentication if the predicate $12 - d^2 \geq 0$ satisfies, where d denotes the distance between the tap coordi-

[2] These datasets with the detailed information will be available at http://sefcom.asu.edu/pga/.

nates and the stored coordinates. The starting and ending points of line gestures and the center of circle gestures are measured with the same predicate [29].

The differences between PGA and the first BDAS scheme proposed in [18] include: i) in PGA, a user uploads his or her picture as the background instead of choosing one from a predefined picture repository; ii) a user is only allowed to draw three specific types of gestures in PGA, while BDAS takes any form of strokes. The first difference makes PGA more secure than the previous scheme, because a password dictionary could only be generated after the background picture is acquired. However, the second characteristic reduces the theoretical password space from its counterpart. Pace et al. [29] quantified the size of the theoretical password space of PGA which is $2^{30.1}$ with current length-three configuration in Windows 8^{TM}. For more details, please refer to [29].

3 An Empirical Analysis of Picture Gesture Authentication

In this section, we present an empirical analysis on user choice in PGA by analyzing data collected from our user studies. Our empirical study is based on human cognitive capabilities. Since human cognition of pictures is limited in a similar way to their cognition of texts, the picture passwords selected by users are probably constrained by human cognitive limits which would be similar to the ones in text-based passwords [42].

3.1 Experiment Design

For the empirical study, we developed a web-based PGA system for conducting user studies. The developed system resembles Windows 8^{TM} PGA in terms of its workflow and appearance. The differences between our implementation and Windows 8^{TM} PGA include: i) our system works with major browsers in desktop PCs and tablets whereas Windows 8^{TM} PGA is a stand-alone program; ii) some information, such as the criterion for circle radius comparison, is not disclosed. In other words, our implementation and Windows 8^{TM} PGA differ in some criteria (we regard radiuses the same if their difference is smaller than 6 segments in grid). In addition, our developed system has a tutorial page that includes a video clip educating how to use the system and a test page on which users can practice gesture drawings.

Our study protocol, including the type of data we plan to collect and the questionnaire we plan to use, was reviewed by our institution's IRB. The questionnaire consisted of four sections: i) general information of the subject (gender, age, level of education received, and race); ii) general feeling toward PGA (is it easier to remember, faster to input, harder to guess, and easier to observe than text-based password); iii) selection of background picture (preferred picture type); and iv) selection of password (preferred gesture location and type).

We started user studies after receiving the IRB approval letter in August 2012 and compiled two datasets from August 2012 to January 2013 using this system. *Dataset-1* was acquired from a testbed of picture password used by an undergraduate computer science class. *Dataset-2* was produced by advertising our studies in schools of engineering and business in two universities and Amazon's Mechanical Turk crowdsourcing service that has been used in security-related research work [26]. Turkers who had finished more than 50 tasks and had an approval rate greater than 60% were qualified for our user study.

For registration, subjects in *Dataset-1* were asked to provide their student IDs for a simple verification after which they were guided to upload a picture, register a password and then use the password to access class materials including slides, homework, assignments, and projects. Subjects used this system for the Fall 2012 semester which lasted three and a half months at our university. If subjects forgot their passwords during the semester, they would inform the teaching assistant who reset their passwords. Subjects were allowed to change their passwords by clicking a change password link after login. There were 56 subjects involved in *Dataset-1* resulting in 58 unique pictures, 86 registered passwords, and 2,536 login attempts.

Instead of asking subjects to upload pictures for *Dataset-2*, we chose 15 pictures (please refer to Appendix B for the pictures) in advance from the PASCAL Visual Object Classes Challenge 2007 dataset [19]. We chose these pictures because they represent a diverse range of pictures in terms of category (portrait, wedding, party, bicycle, train, airplane and car) and complexity (pictures with few and plentiful stand-out regions). Subjects were asked to choose one password for each picture by pretending that it was protecting their bank information. The 15 pictures were presented to subjects in a random order to reduce the dependency of password selection upon the picture presentation order. 762 subjects participated in the *Dataset-2* collection resulting in 10,039 passwords. The number of passwords for each picture in the *Dataset-2* varies slightly, with an average of 669, because some subjects quit the study without setting up passwords for all pictures.

For both datasets, subjects were asked to finish the aforementioned questionnaire to help us understand their experiences. We collected 685 (33 for *Dataset-1*, 652 for *Dataset-2*) copies of survey answers in total. According to the demographic-related inquiries in the exit survey, 81.8% subjects in *Dataset-1* are self-reported male and 63.6% are between 18 and 24 years old. While partic-

Table 1: Survey Question: Which of the following best describes what you are considering when you choose locations to perform gestures?

Multi-choice Answers	Dataset		
	1	*2*	Overall
I try to find locations where special objects are.	24 (72.7%)	389 (59.6%)	413 (60.3%)
I try to find locations where some special shapes are.	8 (24.2%)	143 (21.9%)	151 (22.1%)
I try to find locations where colors are different from their surroundings.	0 (0%)	57 (8.7%)	57 (8.3%)
I randomly choose a location to draw without thinking about the background picture.	1 (3.0%)	66 (10.1%)	67 (9.8%)

ipants in *Dataset-2* are more diverse with 64.4% male, 37.2% among 18 to 24 years old, 45.4% among 25 - 34, and 15.0% among 35 - 50. Even though the subjects in our studies do not represent all possible demographics, the data collected from them represents the most comprehensive PGA usage so far. Their tendencies could provide us with significant insights into the user choice in PGA.

3.2 Results

This section summarizes our empirical analysis on the above-mentioned datasets by presenting five findings.

3.2.1 Finding 1: Relationship Between Background Picture and User's Identity, Personality, or Interests

We analyzed all unique pictures[3] in *Dataset-1*, and the background pictures chosen by subjects range from celebrity to system screenshot. We categorize them into six classes: i) people (27/58), ii) civilization (7/58), iii) landscape (3/58), iv) computer-generated picture (14/58), v) animals (6/58), and vi) others (1/58).

For the category of ‘people’, 6 pictures were categorized as ‘me’; 12 pictures were subjects’ families; 4 were pictures of subjects’ friends; and 5 were celebrities. The analysis of answers to the survey question “*Could you explain why you choose such types of pictures?*” revealed two opposite attitudes towards using picture of people. The advocates for such pictures considered: i) it is more friendly. e.g. “*The image was special to me so I enjoy seeing it when I log in*”; ii) it is easier for remembering passwords. e.g. “*Marking points on a person is easier to remember*”; and iii) it makes password more secure. e.g. “*The picture is personal so it should be much harder for someone to guess the password*”. However, other participants believed it may leak his or her identity or privacy. e.g. “*revealing myself or my family to anyone who picks up the device*”. They preferred other types of pictures because “*less personal if someone gets my picture*” and “*landscape usually doesn’t have any information about who you are*”.

Table 2: Attributes of Most Frequently Used PoIs

Attributes	# Gesture	# Password	# Subject
Eye	36	20	19
Nose	21	13	10
Hand/Finger	6	5	4
Jaw	5	3	3
Face (Head)	4	2	2

14 pictures in *Dataset-1* could be categorized as computer-generated pictures including computer game posters, cartoons, and some geometrical graphs. 24.1% (14/58) of such pictures were observed in *Dataset-1* but the survey results indicated 6.4% (42/652) of participants were in such a usage pattern in *Dataset-2* based on the following survey question: “*Please indicate the type of pictures you prefer to use as the background*”. We concluded the population characteristics (male, age 18-24, college students) in *Dataset-1* were the major reason behind this phenomenon. The answers to “*Could you explain why you choose such types of pictures?*” in *Dataset-1* supported this conjecture: “*computer game is something I am interested [in] it*” and “*computer games picture is personalized to my interests and enjoyable to look at*”.

It is obvious that pictures with personally identifiable information may leak personal information. However, it is less obvious that even pictures with no personally identifiable information may provide some clues which may reveal the identity or persona of a device owner. Traditional text-based password does not have this concern as long as the password is kept secure. Previous graphical password schemes, such as Face and PassPoints, do not have this concern either because pictures are selected from a predefined repository.

3.2.2 Finding 2: Gestures on Points of Interest

The security of background draw-a-secret schemes mostly relies on the location distribution of users’ gestures. It is the most secure if the locations of users’ gestures follow a uniform distribution on any picture. However, such passwords would be difficult to remember and may not be preferable by users. By analyzing the collected passwords, we notice that subjects frequently chose standout regions (points of interest, PoIs) on which to draw. As shown in Table 1, only 9.8% subjects claimed to choose locations randomly without caring about the background picture. The observation is supported by survey answers to “*Could you explain the way you choose locations to perform gestures?*”: “*If I have to remember it; it [would] better stand out.*” and “*Something that would make it easier to remember*”.

Even though the theoretical password space of PGA is

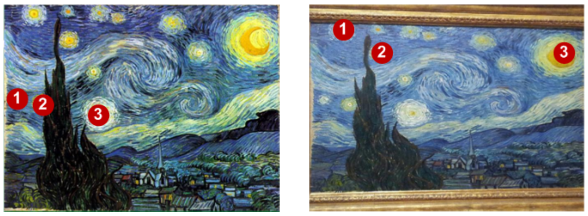
[3] Due to the confidentiality agreement with the subjects, we are not able to share pictures that are marked having personally identifiable information.

Table 3: Numbers of Gesture Type Combinations and Average Time Spent on Creating Them

		3×t	3×l	3×c	2×t+l	2×t+c	2×l+t	2×l+c	2×c+t	2×c+l	t+l+c
Dataset-1	#	60	3	0	9	1	7	1	0	0	5
	Average Time (Seconds)	5.74	12.39	N/A	10.12	21.56	11.17	17.51	N/A	N/A	11.22
Dataset-2	#	3438	1447	253	1211	380	1000	622	192	442	1054
	Average Time (Seconds)	4.33	7.11	9.96	6.02	6.14	7.72	9.98	8.78	10.19	9.37

Table 4: Numbers of Gesture-order Patterns

	H+	H-	V+	V-	DIAG	Others
Dataset-1	43 50.0%	5 5.8%	16 18.6%	4 4.6%	22 25.5%	18 20.9%
Dataset-2	3144 31.3%	1303 12.9%	1479 14.7%	887 8.8%	2621 26.1%	3326 33.1%



(a) (b)

Figure 1: Two Versions of *Starry Night* and Corresponding Passwords

larger than text-based passwords with the same length, a background picture affects user choice in gesture location, reducing the feasible password space tremendously. We summarize three popular ways that subjects used to identify standout regions: i) finding regions with objects. e.g. "*I chose eyes and other notable features*" and "*I chose locations such as nose, mouth or whole face*"; ii) finding regions with remarkable shapes. e.g. "*if there is a circle there I would draw a circle around that*"; and iii) finding regions with outstanding colors. The detailed distribution of these selection processes is shown in Table 1. 60.3% of subjects prefer to find locations where special objects catch their eyes while 22.1% of subjects would rather draw on some special shapes.

3.2.3 Finding 3: Similarities Across Points of Interest

We analyzed the attributes of PoIs that users preferred to draw on. We paid more attention to the pictures of people because it was the most popular category. In the 31 registered passwords for the 27 pictures of people uploaded by 22 subjects in *Dataset-1*, we analyzed the patterns of PoI choice. As shown in Table 2, 36 gestures were drawn on eyes and 21 gestures were drawn on noses. Other locations that attracted subjects to draw included hand/finger, jaw, face (head), and ear. Interestingly, 19 subjects out of 22 (86.3%) drew on eyes at least once, while 10 subjects (45.4%) performed gestures on noses. The tendencies to choose similar PoIs by different subjects are common in other picture categories as well. Figure 1 shows another example where two subjects uploaded two versions of *Starry Night* in *Dataset-1*. The passwords they chose show strikingly similar patterns with three taps on stars, even if there is no single gesture location overlap.

3.2.4 Finding 4: Directional Patterns in PGA Password

Salehi-Abari et al. [32] suggest many passwords in click-based systems follow some directional patterns. We are interested in whether PGA passwords show similar characteristics. For simplicity, we consider the coordinates of tap and circle gestures as their locations and the middle point of the starting and ending points of line as its location. If the x or y coordinate of a gesture sequence follows a consistent direction regardless of the other coordinate, we say the sequence follows a LINE pattern. We divide LINE patterns into four categories: i) H+, denoting left-to-right ($x_i \leq x_{i+1}$); ii) H-, denoting right-to-left ($x_i \geq x_{i+1}$); iii) V+, denoting top-to-bottom ($y_i \leq y_{i+1}$); and iv) V-, denoting bottom-to-top ($y_i \geq y_{i+1}$). If a sequence of gestures follows a horizontal pattern and a vertical pattern at the same time, we say it follows a DIAG pattern.

We examined the occurrence of each LINE and DIAG pattern in the collected data. As shown in Table 4, more than half passwords in both datasets exhibited some LINE patterns, and a quarter of them exhibited some DIAG patterns. Among four LINE patterns, H+ (drawing from left to right) was the most popular one with 50.0% and 31.3% occurrences in *Dataset-1* and *Dataset-2*, respectively. And, V+ (drawing from top to bottom) was the second most popular with 18.6% and 14.7% occurrences in two datasets, respectively. This finding shows it is reasonable to use gesture-order patterns as one heuristic factor to prioritize generated passwords.

3.2.5 Finding 5: Time Disparity among Different Combinations of Gesture Types

We analyzed all registered passwords to understand the gesture patterns and the relationship between gesture type and input time. For 86 registered passwords (258 gestures) in *Dataset-1*, 212 (82.1%) gesture types were taps, 39 (15.1%) were lines, and only 7 (2.7%) were circles. However, the corresponding occurrences for 10,039 registered passwords (30,117 gestures) in *Dataset-2* were 15,742 (52.2%), 10,292 (34.2%), and 4,083 (13.5%), respectively. Obviously, subjects in *Dataset-2* chose more diverse gesture types than subjects in *Dataset-1*. As shown in Table 3, there was a strong connection between the time subjects spent on reproducing passwords and the gesture types they chose. Three taps, the most common gesture combination, appeared in both datasets with the lowest average time (5.74 seconds and 4.33 seconds in corresponding dataset). On the other hand, the passwords with two circles and one line took the longest average input time (10.19 seconds in *Dataset-2*). In the user studies, subjects in *Dataset-2* were asked to set up the passwords by pretending they were protecting their bank information. However, subjects in *Dataset-1* actually used these passwords to access the class materials which they accessed more than four times a week on average. This may be a reason why subjects in *Dataset-1* prefer passwords with simpler gesture type combinations that are easier to reproduce in a timely manner.

4 Attack Framework

In this section, we present an attack framework on Windows 8$^{\text{TM}}$ picture gesture authentication, leveraging the findings addressed in Section 3. Our attack framework takes the target picture's PoIs, a set of learning pictures' PoIs and corresponding password pairs as input, and produces a list of possible passwords, which is ranked in the descending order of the password probabilities.

Next, we first discuss the attack models followed by the representations of picture password and PoI. We then illustrate the idea of a selection function and its automatic identification. We also present two algorithms for generating a selection function sequence list and describe how it can generate picture password dictionaries for previously unseen target pictures.

4.1 Attack Models

Depending on the resources an attacker possesses, we articulate three different attack models: i) *Pure Brute-force Attack*: an attacker blindly guesses the picture password without knowing any information of the background picture and the users' tendencies. The password space in this model is $2^{30.1}$ in PGA [29]. ii) *PoI-assisted Brute-force Attack*: an attacker assumes the user only performs drawings on PoIs of the background picture and this model randomly guesses passwords on identified PoIs. The password space for a picture with 20 PoIs in this model is $2^{27.7}$ [29]. Salehi-Abari et al. [32] designed an approach to automatically identify hot-spots in a picture and generate passwords on them. iii) *Knowledge-based PoI-assisted Attack*: in addition to the assumption for PoI-assisted brute-force attack, an attacker ought to have some knowledge about the password patterns learned from collected picture and password pairs (not necessarily from the target user or picture). The guessing space in this model is the same as the one in PoI-assisted brute-force attack. However, the generated dictionaries in this model are ranked with the higher possibility passwords on the top of the list.

Attack schemes could also be divided into two categories based on whether or not an attacker has the ability to attack previously unseen pictures. The method presented in [32] is able to attack previously unseen pictures for click-based graphical password. It uses click-order heuristics to generate partially ranked dictionaries. However, this approach cannot be applied directly to background draw-a-secret schemes because the gestures allowed in such schemes are much more complex and the order-based heuristics could not capture users' selection processes accurately. In contrast, our attack framework could abstract generic knowledge of user choice in picture password schemes. In addition, as a working *knowledge-based PoI-assisted* model, it is able to generate ranked dictionaries for previously unseen pictures.

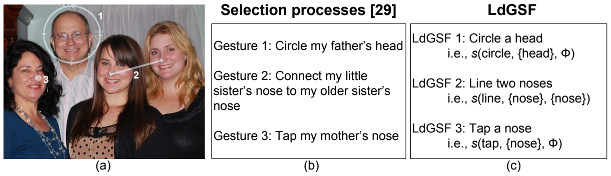
4.2 Password and PoI Representations

We first formalize the representation of a password in PGA with the definition of a location-dependent gesture which represents a single gesture on some locations in a picture.

Definition 1 *A location-dependent gesture (LdG) denoted as π is a 7-tuple $\langle g, x_1, y_1, x_2, y_2, r, d\rangle$ that consists of gesture's type, location, and other attributes.*

In this definition, g denotes the type of LdG that must be one of tap, line, and circle. A tap LdG is further represented by the coordinates of a gesture $\langle x_1, y_1\rangle$. A line LdG is denoted by the coordinates of the starting and ending points of a gesture $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$. A circle LdG is denoted by the coordinates of its center $\langle x_1, y_1\rangle$, radius r, and direction $d \in \{+, -\}$ (clockwise or not). We define the password space of location-dependent gesture as $\Pi = \Pi_{\mathsf{tap}} \bigcup \Pi_{\mathsf{line}} \bigcup \Pi_{\mathsf{circle}}$. A valid PGA password is a length-three sequence of LdGs denoted as $\vec{\pi}$, and the PGA password space could be denoted as $\vec{\Pi}$.

A point of interest is a standout region in a picture. PoIs could be regions with semantic-rich meanings, such as face (head), eye, car, clock, etc. Also, they could stand out in terms of their shapes (line, rectangle, circle, etc.) or colors (red, green, blue, etc.). We denote a PoI by the coordinates of its circumscribed rectangle and some describing attributes. A PoI is a 5-tuple $\langle x_1, y_1, x_2, y_2, D\rangle$, where $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$ are the coordinates of the top-left and bottom-right points of the circumscribed rectangle, and $D \subseteq 2^{\mathscr{D}}$ is a set of attributes that describe this PoI. $\mathscr{D}$ has three sub-categories $\mathscr{D}_o, \mathscr{D}_s$ and $\mathscr{D}_c$ and four wildcards $*_o, *_s, *_c$, and $*$, where $\mathscr{D}_o = \{\mathsf{head, eye, nose, ...}\}$, $\mathscr{D}_s = \{\mathsf{line, rectangle, circle, ...}\}$, and $\mathscr{D}_c = \{\mathsf{red, blue, yellow, ...}\}$. Wildcards are used when no specific information is available. For example, if a PoI is identified with objectness measure [3] that gives



Selection processes [29]

Gesture 1: Circle my father's head

Gesture 2: Connect my little sister's nose to my older sister's nose

Gesture 3: Tap my mother's nose

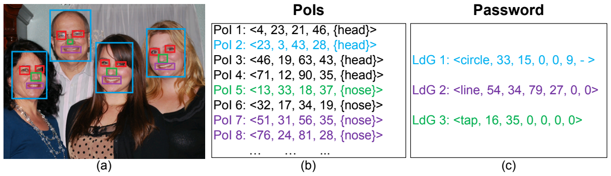
LdGSF

LdGSF 1: Circle a head
i.e., *s*(circle, {head}, Φ)

LdGSF 2: Line two noses
i.e., *s*(line, {nose}, {nose})

LdGSF 3: Tap a nose
i.e., *s*(tap, {nose}, Φ)

Figure 2: (a) Background picture and password (b) User's selection processes that were taken from [30] (c) Corresponding LdGSFs that simulate user's selection processes



PoIs

PoI 1: <4, 23, 21, 46, {head}>
PoI 2: <23, 3, 43, 28, {head}>
PoI 3: <46, 19, 63, 43, {head}>
PoI 4: <71, 12, 90, 35, {head}>
PoI 5: <13, 33, 18, 37, {nose}>
PoI 6: <32, 17, 34, 19, {nose}>
PoI 7: <51, 31, 56, 35, {nose}>
PoI 8: <76, 24, 81, 28, {nose}>
… … …

Password

LdG 1: <circle, 33, 15, 0, 0, 9, - >
LdG 2: <line, 54, 34, 79, 27, 0, 0>
LdG 3: <tap, 16, 35, 0, 0, 0, 0>

Figure 3: (a) Background picture and identified PoIs (b) Identified PoIs (c) Password representations (Colors are used to indicate the connections between the PoIs in (b) and LdGs in (c))

no semantics about the identified region, we mark the PoI's describing attribute as $*$.

4.3 Location-dependent Gesture Selection Functions

A key concept in our framework is the location-dependent gesture selection function (LdGSF) which models and simulates the ways of thinking that users go through when they select a gesture on a picture. The motivation behind this abstraction is that the set of PoIs and their locations differ from picture to picture, but the ways that users think to choose locations for drawing a gesture exhibit certain patterns. This conjecture is supported by our observations from collected data and surveys discussed in Section 3. With the help of LdGSF, the PoIs and corresponding passwords in training pictures are used to generalize picture-independent knowledge that describes how users choose passwords.

Definition 2 *A location-dependent gesture selection function (LdGSF) is a mapping* $s : G \times 2^{\mathscr{D}} \times 2^{\mathscr{D}} \times \Theta \rightarrow 2^{\Pi}$ *which takes a gesture, two sets of PoI attributes, and a set of PoIs in the learning picture as input to produce a set of location-dependent gestures.*

The universal set of LdGSF is defined as S. A length-three sequence of LdGSF is denoted as $\vec{s}$, and a set of length-three LdGSF sequences is denoted as $\vec{S}$. $s(\mathsf{tap}, \{\mathsf{red}, \mathsf{apple}\}, \varnothing, \theta_k)$ is interpreted as 'tap a red apple in the picture p_k' and $s(\mathsf{circle}, \{\mathsf{head}\}, \varnothing, \theta_k)$ as 'circle a head in p_k'. Note that, no specific information of the locations of 'red apple' and 'head' is provided here which makes the representations independent from actual locations of objects in the picture.

One challenge we face is some PoIs may be big enough to take several unique gestures. Let us consider a picture with a big car image in it. Simply saying 'tap a car' could result in lots of distinct tap gestures in the circumscribed rectangle of the car. One solution to this problem is to divide the circumscribed rectangle into a grid with the scale of toleration threshold. However, this solution would result in too many password entries in the generated dictionary. For simplicity, we introduce five inner points for one PoI, namely center, top, bottom, left, and right that denote the center of the PoI and four points of the center of two consecutive corners. Any gesture that falls into the proximities of these five points of a PoI would be considered as an action on this PoI. For some PoIs that are big enough to take an inner line gesture, we put $\varnothing$ as the input of the second set of PoI attributes. $s(\mathsf{line}, \{\mathsf{mouth}\}, \varnothing, \theta_k)$ denotes 'line from the left(right) to the right(left) on the same mouth'. While, $s(\mathsf{line}, \{\mathsf{mouth}\}, \{\mathsf{mouth}\}, \theta_k)$ means 'connect two different mouths'.

Figure 2 shows an example demonstrating how LdGSF simulates a user's selection processes that were taken from [30]. In reality, a user's selection process on a PoI and gesture selection may be determined by some subjective knowledge and cognition. For example, 'circle my father's head' and 'tap my mother's nose' may involve some undecidable computing problems. One solution to handle this issue is to approximate subjective selection processes in objective ways by including some modifiers. 'circle my father's head' may be transformed into 'circle the *uppermost* head' or 'circle the *biggest* head'. However, it is extremely difficult, if not impossible, to accurately approximate subjective selection processes in this way, and it may bring serious over-fitting problems in the learning stage. Instead, we choose to ignore subjective information by abstracting 'circle my father's head' to 'circle a head'. A drawback of this abstraction is that an LdGSF may return more than one LdG and we have no knowledge to rank them directly, as they come from the same LdGSF. Using Figure 2(a) as an example, 'circle a head' outputs four different LdGs on each head in the picture. The LdGSF sequence shown in Figure 2(c) generates $4 \times (4 \times 3) \times 4 = 192$ passwords. To cope with this issue, we use gesture-order to rank the passwords generated by the same LdGSF sequence that will be detailed in Section 4.5. Next, we present an automated approach to extract users' selection processes from the collected data and represent them with LdGSFs.

Figure 3 shows an example demonstrating that how to extract users' selection processes from PoIs automatically. First, PoIs in the background picture are identified using mature computer vision techniques such as object detection, feature detection and objectness measure. Then, each LdG in a password is compared with

PoIs based on their coordinates and sizes. If a match between PoIs and LdGs is found, a new LdGSF is created as the combination of the LdG's gesture type and PoI's attributes. For instance, the location and size of LdG 1 in Figure 3(c) matches PoI 2 in Figure 3(b) (the locations of the circle gesture and PoI center are compared first; then, the radius of the circle is compared with 1/2 of PoI's height and width). Then, an LdGSF $s(\mathsf{circle}, \{\mathsf{head}\}, \varnothing)$ is created which is equivalent to the LdG shown in Figure 2(c).

To choose a password in PGA, the user *selects* a length-three LdGSF sequence. With the definition of LdGSF, the generation of ranked password list is simplified into the generation of the ranked LdGSF sequence list. Let order: $\vec{S} \to \{1..|\vec{S}|\}$ be a bijection which indicates the order LdGSF sequences should be performed. The objective of generating ranked LdGSF sequence list is to find such a bijection.

4.4 LdGSF Sequence List Generation and Ordering

Now we present our approach to find the aforementioned bijection that indicates the order that the LdGSF sequences should be performed on a target picture for generating the password dictionary. Our framework is not dependent on certain rules, but is adaptive to the tendencies shown by users who participate in the training set. The characteristic of adaptiveness helps our framework generate dedicated guessing paths for different training data. Next, we present two algorithms for obtaining such a feature.

4.4.1 BestCover LdGSF Sequence List Generation

We first propose an LdGSF sequence list generation algorithm named BestCover that is derived from $\mathscr{B}_{\mathsf{emts}}$ [44]. The objective of BestCover LdGSF sequence list generation is to optimize the guessing order for the sequences in the list by minimizing the expected number of sequences that need to be tested on a random choice of picture in the training dataset.

The problem is formalized as follows: **Instance**: The collection of LdGSF sequences $\vec{s}_1, ..., \vec{s}_n$ and corresponding picture password $\vec{\pi}_1$,...,$\vec{\pi}_n$, for which $\vec{s}_i(\theta_i) \ni \vec{\pi}_i, i \in \{1..n\}$ and $\theta_1, .., \theta_n$ are the sets of PoIs in pictures $p_1, .., p_n$. **Question**: Expected Min Selection Search (emss): The objective is to find order so as to minimize $\mathbb{E}(\min\{i : \vec{s}_i(\theta_r) \ni \vec{\pi}_r\}$, where $\vec{s}_i = \mathsf{order}^{-1}(i)$ and the expectation is taken with respect to a random choice of $r \leftarrow \{1..n\}$.

The hardness of this problem is that different LdGSFs and LdGSF sequences may generate the same list of LdGs and passwords. For instance, 'tap a red object' and 'tap an apple' turn out the same result on a picture in which there is a red apple. An overlap in different LdGSF results is similar to the coverage characteristics in the set cover problem. We can prove the NP-hardness of emss by reducing from emts [44]. Due to space limitations, we omit the corresponding proof. We give an approximation algorithm for emss in Algorithm 1 that is a modification from $\mathscr{B}_{\mathsf{mssc}}$ [20]. The time complexity of BestCover is $O(n^2 + |\vec{S}'|log(|\vec{S}'|))$.

Algorithm 1: BestCover($(\vec{s}_1, .., \vec{s}_n)$,$(\vec{\pi}_1$,...,$\vec{\pi}_n)$)

```
for $i = 1..n$ do
    $T_{\vec{s}_i} \leftarrow \{k : \vec{s}_i(\theta_k) \ni \vec{\pi}_k\}$;
end
$\vec{S}' \leftarrow \{\vec{s} : |T_{\vec{s}}| > 0\}$;
for $i = 1..|\vec{S}'|$ do
    order$^{-1}(i) \leftarrow \vec{s}_k$, that $T_{\vec{s}_k}$ has most elements that are not
    included in $\bigcup_{i'<i}$order$^{-1}(i')$;
end
return order
```

BestCover is good for a training dataset that consists of comprehensive and large scale password samples, because it assumes the target passwords exhibit same or at least very similar distributions to the training data. However, if the training dataset is small and biased, the results from BestCover may over-fit the training data and fail in testing data.

4.4.2 Unbiased LdGSF Sequence List Generation

The over-fitting problem in BestCover is brought about by the biased PoI attribute distributions in training data. For example, we have a training set with 9 pictures of apples and 1 picture of a car, and 5 corresponding passwords have circles on apples and 1 has a circle on car. In the generated LdGSF sequence list, BestCover will put sequences with 'circle an apple' prior to the ones with 'circle a car', because the former ones have an LdGSF that was used in more passwords. However, we can see the probability for users to circle car (1/1) is higher than apples (5/9) if we consider the occurrences of apple and car in pictures.

Unbiased LdGSF sequence list generation copes with this issue by considering the PoI attribute distributions. It removes the biases from the training dataset by normalizing the occurrences of LdGSFs with the occurrences of their corresponding PoIs. Let $D_{\vec{s}_k} \subseteq \theta$ denote the event that θ contains enough PoIs that have attributes specified in $\vec{s}_k$. If a PoI with a specific type of attributes does not exist in a picture, the probability that a user select the PoI with such an attribute on this picture to draw a password is 0, denoted as $Pr(\vec{s}_k|\overline{D_{\vec{s}_k} \subseteq \theta}) = 0$, e.g. a user would not think and perform 'tap a red apple' on a picture without the existence of the red apple. We assume each LdGSF in a sequence is independent of each other and approximately compute $Pr(\vec{s}_k|D_{\vec{s}_k} \subseteq \theta)$ with Equation 1.

$$
\begin{aligned}
&Pr(\vec{s_k}|D_{\vec{s_k}} \subseteq \theta) \\
&= Pr(s_1 s_2 s_3|D_{s_1} \subseteq \theta \wedge D_{s_2} \subseteq \theta \wedge D_{s_3} \subseteq \theta) \\
&= Pr(s_1|D_{s_1} \subseteq \theta) \times Pr(s_2|D_{s_2} \subseteq \theta) \times Pr(s_3|D_{s_3} \subseteq \theta)
\end{aligned}
\tag{1}
$$

For each $s_i \in S$, we compute $Pr(s_i|D_{s_i} \subseteq \theta)$ with Equation 2:

$$
Pr(s_i|D_{s_i} \subseteq \theta) = \frac{\sum_{j=1}^{n} count(D_{s_i}, \vec{\pi}_j)}{\sum_{j=1}^{n} count(D_{s_i}, \theta_j)} \tag{2}
$$

where $\sum_{j=1}^{n} count(D_{s_i}, \vec{\pi}_j)$ denotes the number of LdGs in passwords of the training set that share the same attributes with s_i, and $\sum_{j=1}^{n} count(D_{s_i}, \theta_j)$ denotes the number of PoIs in the training set that share the same attributes with s_i. $Pr(s_i|D_{s_i} \subseteq \theta)$ describes the probability of using a certain LdGSF when there are enough PoIs with the required attributes.

The Unbiased algorithm generates an LdGSF sequence list by ranking $Pr(\vec{s_k}|D_{\vec{s_k}} \subseteq \theta)$ instead of $Pr(\vec{s_k})$ in descending order as shown in Algorithm 2. The time complexity of Unbiased is $O(n|S| + |\vec{S}|log(|\vec{S}|))$. The Unbiased algorithm would be better for the scenarios where fewer samples are available or samples are highly biased.

Algorithm 2: Unbiased(S)

```
for $s \in S$ do
    Compute $Pr(s|D_s \subseteq \theta)$ with Equation 2;
end
for $\vec{s} \in \vec{S}$ do
    Compute $Pr(\vec{s}|D_{\vec{s}} \subseteq \theta)$ with Equation 1;
end
for $i = 1..|\vec{S}|$ do
    order$^{-1}(i) \leftarrow \vec{s_k}$, that $Pr(\vec{s_k}|D_{\vec{s_k}} \subseteq \theta)$ holds the $i$-th position
    in the descending ordered $Pr(\vec{s}|D_{\vec{s}} \subseteq \theta)$ list;
end
return order
```

4.5 Password Dictionary Generation

The last step in our attack framework is to generate the password dictionary for a previously unseen target picture. First, the PoIs in the previously unseen picture are identified. Then, a dictionary is acquired by applying the LdGSF sequences on the PoIs, following the order created by the BestCover or Unbiased algorithm. Obviously, the passwords generated by an LdGSF sequence that holds a higher position in the LdGSF sequence list will also be in higher positions in the dictionary. However, as addressed earlier, BestCover and Unbiased algorithms do not provide extra information to rank the passwords generated by the same LdGSF sequence. Inspired by using the click-order patterns as the heuristics for dictionary generation [32], we propose to rank such passwords generated by the same LdGSF sequence with gesture-orders. In the training stage, we record the gesture-order occurrence of each LINE and DIAG pattern and rank the patterns in descending order. In the attack stage, for the passwords generated by the same LdGSF sequence, we reorder them with their gesture-orders in the order of LINE and DIAG patterns. Passwords that do not belong to any LINE or DIAG pattern hold lower positions.

5 Implementation and Evaluation

5.1 PoI Identification

We chose OpenCV [1] as the computer vision framework for our implementation and collected several feature detection tools for automatically identifying PoIs in background pictures. The computer vision techniques we adopted include: i) object detection: the goal of object detection is to find the locations and sizes of semantic objects of a certain class in a digital image. Viola-Jones object detection framework [40] is the first computationally affordable online object detection framework that utilizes Haar-like features instead of image intensities. Each learned classifier is represented and stored as a haar cascade. We collected 30 proven haar cascades from [31] for 8 different object classes including face (head), eye, nose, mouth, ear, head, body, and clock. ii) low-level feature detection: due to the high positive and high negative rates of object detection, we also resorted to some low-level feature detection algorithms that identify standout regions without extracting semantics. To identify regions whose colors are different from their surroundings, we first converted the color pictures to black and white, then found the contours using algorithms in [35]. For the circle detection, we used Canny edge detector [10] and Hough transform algorithms [5]. iii) objectness measure: objectness measure [3] deals with class-generic object detection. Different from detecting objects in a specific class, the objectness measure finds the locations and sizes of class-generic objects whose colors and textures are opposed to the background images. Objectness measure could be considered as a technique combining several low-level feature detectors together. We used an objectness measure library from [2] that is able to locate objects and give numerical confidence values with its results.

Figure 4 displays the PoI detection results on four example pictures in *Dataset-2*. As we can see in Figure 4(b), circle detection could identify both bicycle wheels and car badge, but its false positive rate is a little high. Contour detection is the most robust algorithm with a low false positive rate which could locate regions whose colors are different as shown in Figure 4(c). Objectness measure shown in Figure 4(d) could also iden-

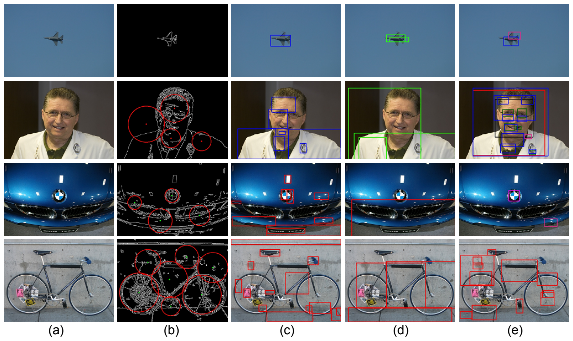


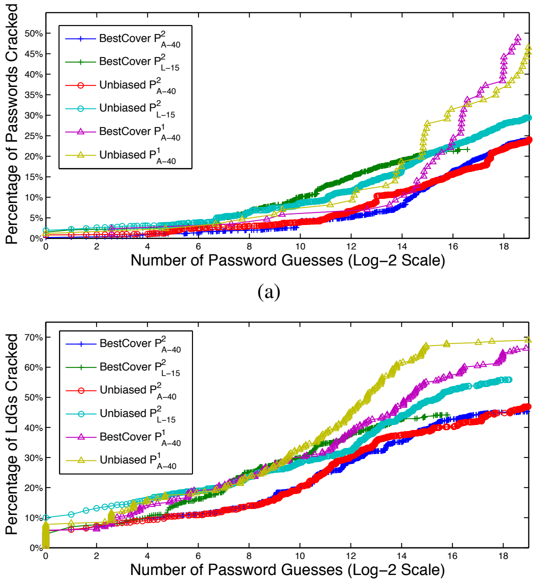
Figure 4: PoI Identification on Example Pictures in *Dataset-2*: (a) Original pictures (b) Circle detection with Hough transform (c) Contour detection (d) Objectness measure (e) Object detection

tify regions whose colors and textures are different from their surroundings. Since most haar cascades we used are designed for facial landmarks, they work smoothly on portraits as does the second picture in Figure 4(e). However, the results show relatively high false positive rates on pictures from other categories. In order to identify more PoIs as accurate as possible, our approach in PoI identification leveraged two steps. In the first step, all possible PoIs were identified using different kinds of tools. In the second step, we examined all identified PoIs and removed duplicates by comparing their locations, sizes and attributes. Then, our approach generated a PoI set called $P^1_{A\text{-}40}$ and $P^2_{A\text{-}40}$ for each picture in *Dataset-1* and *Dataset-2*, respectively. Those PoI sets consisted of at most 40 PoIs with the highest confidences.

Since our attack algorithms are independent from the PoI identification algorithms, we are also interested in examining how our attack framework performs with ideal PoI annotations for pictures. Besides using the automated PoI identification techniques, we manually annotated pictures in *Dataset-2* for some outstanding PoIs as well. To annotate the pictures, we simply recorded the locations and attributes of at most fifteen most appealing regions in the pictures without referring to any password in the collected dataset. We call this annotated PoI set $P^2_{L\text{-}15}$.

5.2 Attack Evaluation

Offline Attacks. Due to the introduction of a tolerance threshold, picture passwords may be more difficult to store securely compared with text-based passwords that are normally saved after salted hashing. Even though the approach that Windows 8$^{\text{TM}}$ is adopting to store picture passwords remains undisclosed, we could consider two attack scenarios where picture passwords are prone to offline attacks. In the first scenario, all passwords which fall into the vicinity (defined by the threshold) of chosen passwords could be stored in a file with salted hashes for comparison. An attacker who has access to this file could perform offline dictionary attacks like cracking text-based password systems. In the second scenario, picture passwords could be used for other purposes besides logging into Windows 8$^{\text{TM}}$, where no constraint on the number of attempts is enforced. For example, a registered picture password could be transformed and used as a key to encrypt a file. An attacker who acquires the encrypted file would like to perform an offline attack.



(a)

(b)

Figure 5: (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Percentage of LdGs cracked vs. number of password guesses, per condition. For *Dataset-1*, there are 86 passwords that include 258 LdGs. For *Dataset-2*, there are 10,039 passwords that have 30,117 LdGs.

In order to attack passwords from a previously unseen picture, the training dataset excluded passwords from the target picture. More specifically, to evaluate *Dataset-1* (58 unique pictures), we used passwords from 57 pictures as the training data and attacked the passwords for the last picture. To evaluate *Dataset-2* (15 unique pictures), we used passwords for 14 pictures as training data, learned the patterns exhibited in the training data, and generated a password dictionary for the last picture. The same process was carried out 58 and 15 times for *Dataset-1* and *Dataset-2*, respectively, in which the target picture was different in each round. The size of the dictionary was set as 2^{19} which is 11-bit smaller than the theoretical password space. We compared all collected passwords for the target picture with the generated dictionary for the picture, and recorded the number of password guesses.

The offline attack results within 2^{19} guesses in different settings are shown in Figure 5. There are 86 passwords in *Dataset-1*, which have a total of 258 LdGs.

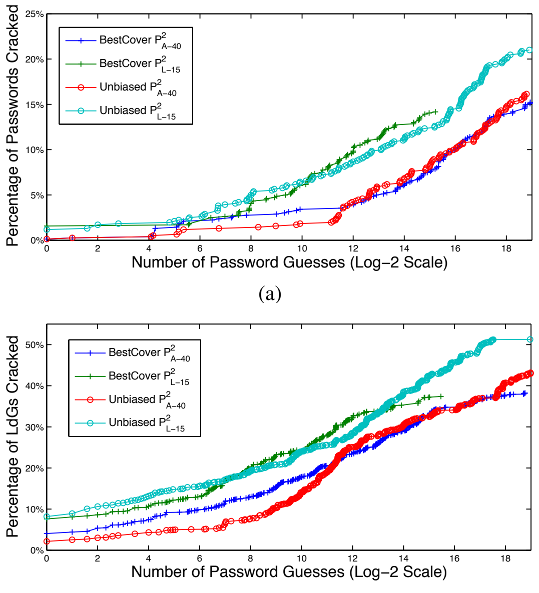


Figure 6: (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Percentage of LdGs cracked vs. number of password guesses, per condition. Only the first chosen password by each subject in *Dataset-2* was considered. There are 762 passwords that have 2,286 LdGs.

And 10,039 passwords were collected in *Dataset-2*, containing a total of 30,117 LdGs. For *Dataset-1*, BestCover cracks 42 (48.8%) passwords out of 86 while Unbiased cracks 40 (46.5%) passwords for the same dataset with $P^1_{A\text{-}40}$. For *Dataset-1*, 178 LdGs (68.9%) out of 258 are cracked with Unbiased and 171 (66.2%) are broken with BestCover. On the other hand, Unbiased with $P^2_{L\text{-}15}$ breaks 2,953 passwords (29.4%) out of 10,039 for *Dataset-2*. This implies Unbiased with $P^2_{A\text{-}40}$ cracking 2,418 passwords (24.0%) is the best result for all purely automated attacks on *Dataset-2*. As Figure 5 suggests, BestCover outperforms Unbiased slightly when ample training data is available. The better performance of both algorithms on *Dataset-1* is because the password gesture combinations in *Dataset-1* are relatively simpler than the ones in *Dataset-2* as we discussed in Section 3.2.5.

In *Dataset-2*, subjects may not choose all 15 passwords with the same care as they were eager to finish the process. To reduce this effect, we ran another analysis in which only the first chosen password by each subject was considered. There are 762 passwords that have 2,286 LdGs. Like previous analysis, the training dataset excluded passwords from the target picture. As shown in Figure 6, results of this analysis are not as good as previous ones. Unbiased with $P^2_{L\text{-}15}$ breaks 160 passwords (21.0%) out of 762. Unbiased with $P^2_{A\text{-}40}$ cracking 123 passwords (16.1%). BestCover cracks 108 (14.2%) and 116 (15.2%) with $P^2_{L\text{-}15}$ and $P^2_{A\text{-}40}$, respectively.

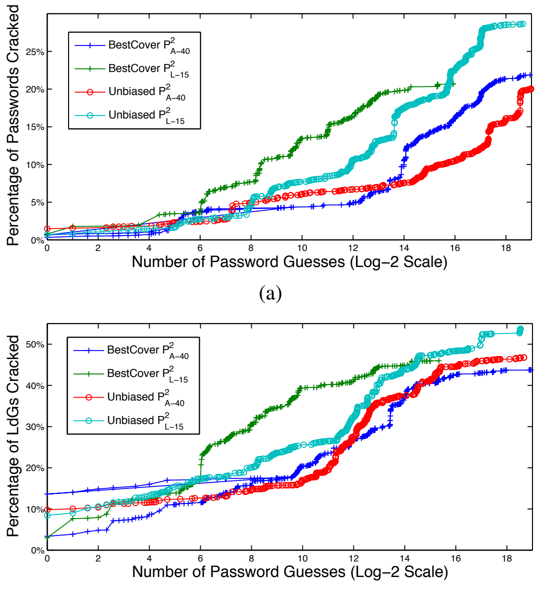
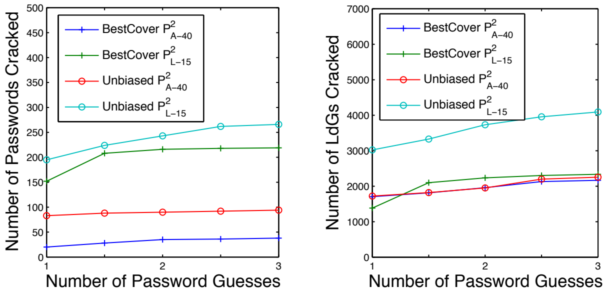


Figure 7: (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Percentage of LdGs cracked vs. number of password guesses, per condition. Only passwords for pictures 243, 1116, 2057, 4054, 6467, and 9899 were considered. There are 4,003 passwords that have 12,009 LdGs.

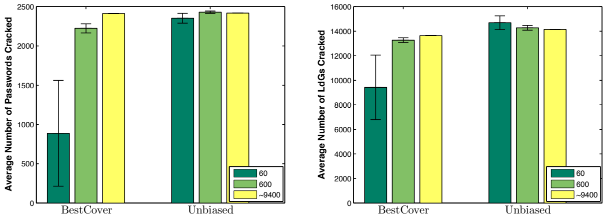
Since some pictures in *Dataset-2* are similar, we ran an additional analysis in which only passwords for pictures 243 (airplane), 1116 (portrait), 2057 (car), 4054 (wedding), 6467 (bicycle), and 9899 (dog) were considered. There are 4,003 passwords that have 12,009 LdGs. Unbiased with $P^2_{L\text{-}15}$ breaks 1,147 passwords (28.6%) while 803 passwords (20.1%) are cracked by Unbiased with $P^2_{A\text{-}40}$. BestCover cracks 829 (20.7%) and 875 (21.8%) with $P^2_{L\text{-}15}$ and $P^2_{A\text{-}40}$ respectively. Results of this analysis are not as good as results with passwords from all pictures.

Online Attacks. The current Windows 8™ allows five failure attempts before it forces users to enter their text-based passwords. Therefore, breaking a password under five guesses implies the feasibility for launching an online attack. Figure 8 shows a refined view of the number of passwords and LdGs cracked with the first five guesses per condition. Purely automated attack Unbiased with $P^2_{A\text{-}40}$ breaks 83 passwords (0.8%) with the first guess and cracks 94 passwords (0.9%) within the first five guesses, while BestCover with $P^2_{A\text{-}40}$ cracked 20 passwords (0.2%) for the first guess and 38 passwords (0.4%) within five guesses. Additionally, Unbiased with $P^2_{A\text{-}40}$ breaks 1,723 LdGs (5.7%) with the first guess. With the help of manually labeled PoI set $P^2_{L\text{-}15}$, the results are even better. For example, Unbiased breaks 195 passwords (1.9%) for the first guess and 266 (2.6%) within the first five guesses. In the meantime, Unbi-



(a) (b)

Figure 8: (a) Number of passwords cracked within five guesses, per condition. (b) Number of LdGs cracked within five guesses, per condition.



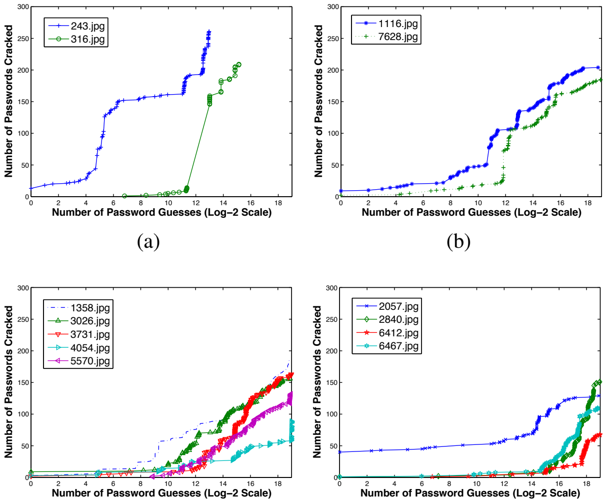
(a) (b)

Figure 9: (a) Average number of passwords cracked vs. different training data sizes. (b) Average number of LdGs cracked vs. different training data sizes. $P^2_{A\text{-}40}$ is used for this analysis. Average over 3 analyses, with one standard deviation shown.

ased with $P^2_{L\text{-}15}$ breaks 3,022 LdGs (10.0%) with the first guess and 4,090 LdGs (13.5%) with five guesses.

Effects of Training Data Size. In Figure 9, we show the password and LdG cracking results with different sizes of training datasets. For each algorithm, we used $P^2_{A\text{-}40}$ as the PoI set and performed three analyses with 60, 600, and all available passwords (about 9,400) as training data, respectively. The sizes of 60 and 600 represent two cases: i) a training set (60) is ten times smaller than the target set (about 669); and ii) a training set (600) is almost the same size as the target set (about 669). For training datasets with the sizes of 60 and 600, we randomly selected these training passwords and performed each analysis three times to get the averages and standard deviations.

As Figure 9 shows, BestCover with 60 training samples could only break an average of 888 passwords (8.8%) out of 10,039. And the standard deviation is as strong as 673. While Unbiased with 60 training samples can crack 2,352 passwords (23.4%) that is almost the same as the results generated from all available training samples. Also, the standard deviation for three trials is as low as 62. The results from BestCover with 600 training samples are much better than the counterparts with 60 training samples. All these observations are expected as Unbiased could eliminate the biases considered in BestCover. The results clearly demonstrate the benefit of using the Unbiased algorithm when a training dataset is small.



(a) (b)

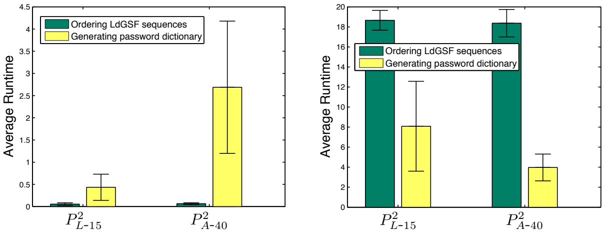
(c) (d)

Figure 10: (a) pictures with fewer PoIs (b) portraits (c) pictures with people in them (d) pictures with lots of PoIs. Unbiased algorithm on $P^2_{A\text{-}40}$ is used for this analysis. (Please refer to Appendix B for the pictures).

Effects on Different Picture Categories. We measured the attack results on different picture categories as shown in Figure 10 where each subfigure depicts the number of passwords cracked versus the number of password guesses. Each curve in a subfigure corresponds to a picture as shown in the legend. Our approach cracks more passwords for a picture, if the curve is skewed upward. And the cracking is faster (with fewer guesses), if the curve is leaned toward the left.

Figure 10(a) provides a view of the attack results on target pictures 243 and 316, each of which has only one airplane flying in the sky. Fewer PoIs in these two pictures make subjects choose more similar passwords. Unbiased with $P^2_{A\text{-}40}$ breaks 261 passwords (39.0%) for the picture 243 and 209 (31.2%) for the picture 316. The cracking success rates are much higher than the average success rate in *Dataset-2* under the same condition. Note that the size of generated dictionaries for these two pictures are smaller than 2^{19} due to the number of available PoIs.

In Figure 10(b), we show the results on two *portrait* pictures where Unbiased with $P^2_{A\text{-}40}$ cracks 389 passwords (29.0%) for both in total. The attack success rate is much higher than the average success rate in *Dataset-2*. This is due to the fact that state-of-the-art computer vision algorithms work well on facial landmarks and subjects' tendencies of drawing on these features are high. The results show that passwords on simple pictures with fewer PoIs or portraits, for which state-of-the-art com-



(a) BestCover algorithm (b) Unbiased algorithm

Figure 11: Average runtime in seconds to order LdGSF sequences using BestCover and Unbiased. Average over 15 pictures in *Dataset-2* with one standard deviation shown.

puter vision techniques could detect PoIs with high accuracy, are easier for attackers to break.

Figure 10(c) shows the attack results on 5 pictures of people. Some of these pictures only have very small figures of people and others have larger figures but not big enough to be considered as a portrait. Unbiased with $P^2_{A\text{-}40}$ cracks 726 passwords (21.7%) for these 5 pictures in total, which is lower than the average success rate in *Dataset-2*.

Figure 10(d) shows the attack results on 4 miscellaneous pictures, two of which are bicycle pictures and the other two are car pictures. The picture, 6412.jpg, has a bicycle leaning against the wall. Different colors on the bicycle and wall in this picture make it cluttered and have lots of PoIs. Unbiased with $P^2_{A\text{-}40}$ only cracks 68 passwords (10.1%) for this picture. However, Unbiased with $P^2_{A\text{-}40}$ cracked 458 (17.1%) for all 4 pictures.

Performance. We also evaluated the performance of our attack approach. Our analyses were carried out on a computer with dual-core processor and 4GB of RAM. In Figure 11, we show the average runtime for our algorithms to order the LdGSF sequences and generate dictionary for a picture in *Dataset-2*. Each bar represents the average time in seconds over 15 pictures with the standard deviation using different algorithms and PoI sets. The results show that BestCover is much faster than Unbiased under the same condition. The average runtime for BestCover on $P^2_{A\text{-}40}$ to order LdGSF sequences is only 0.06 seconds and to generate a dictionary is 2.68 seconds, while Unbiased spends 18.36 and 3.96 seconds, respectively. As we analyzed in Section 4.4, such a difference is caused by the complexity of each algorithm. With such a prompt response, BestCover could be used for online queries.

6 Discussion

6.1 Picture-Password-Strength Meter

Our framework could enhance the security of PGA so it would eventually protect users and their devices by providing a picture-password-strength meter. One way to help users choose secure passwords is to enforce some composition policies, such as 'three taps are not allowed'. However, a recent effort [26] on text-based password found that rule-based password compositions are ineffective because they can allow weak passwords and reject strong ones. The cornerstone of accurate strength measurement is to quantify the strength of a password. With a ranked password dictionary, our framework, as the first potential picture-password-strength meter, is capable of quantifying the strength of selected picture passwords. More intuitively, a user could be informed of the potential number of guesses for breaking a selected password through executing our attack framework.

6.2 Other Attacks on PGA

Besides keyloggers that record users' finger movements, there are some other attack methods that may affect the security of PGA and other background draw-a-secret schemes. Shoulder surfing, an attack where attackers simply observe the user's finger movements, is one of them. In our survey, 54.3% participants believe the picture password scheme is easier for attackers to observe when they are providing their credentials than text-based password. Several new shoulder surfing resistant schemes [22, 43] were proposed recently. However, the usability is always a major concern for these approaches. The smudge attack [4] which recovers passwords from the oily residues on a touch-screen has also been proven feasible to the background draw-a-secret schemes and could pose threats to PGA.

6.3 Limitations of Our Study

While we took great efforts to maintain our studies' validity, some design aspects of our studies and developed system may have caused subjects to behave differently from what they do on Windows 8$^{\text{TM}}$ PGA. Subjects in *Dataset-2* pretended to access their bank information but did not have anything at risk. Schechter et al. [33] suggest that role playing like this affects subjects' security behavior, so passwords in *Dataset-2* may not be representative of real passwords chosen by real users. Besides, we did not record whether a subject used a tablet with touch-screen or a desktop with mouse. The different ways of input may affect the composition of passwords. Moreover, *Dataset-2* includes multiple passwords per user and this may have impacted the results. In our analyses, training password datasets include passwords from the targeted subject. Even though this may have affected the results, we believe it is less influential. Because, for each analysis, there were around 9,400 training passwords for which only 14 came from the targeted user.

Since all training passwords were treated equally, the influence brought by the 0.14% training data is low. As discussed in Section 5.2, even though our online attack results showed the feasibility of our approach, it still requires more realistic and significant attack cases. As part of future work, we plan to integrate smudge attacks [4] into our framework to improve the efficacy of our online attacks.

7 Related Work

The security and vulnerability of text-based password have attracted considerable attention because of several infamous password leakage incidents in recent years. Zhang et al. [44] studied the password choices over time and proposed an approach to attack new passwords from old ones. Castelluccia et al. [11] proposed an adaptive Markov-based password strength meter by estimating the probability of password using training data. Kelley et al. [26] developed a distributed method to calculate how effectively password-guessing algorithms could guess passwords. Even though the attack framework we presented is dedicated to cracking background draw-a-secret passwords, the idea of abstracting users' selection processes of password construction introduced in this paper could also be applicable to cracking and measuring text-based passwords.

The basic idea of attacking graphical password schemes is to generate dictionaries that consist of potential passwords [36]. However, the lack of sophisticated mechanisms for dictionary construction affects the attack capabilities of existing approaches. Thorpe et al. [38] proposed a method to harvest the locations of training subjects' clicks on pictures in click-based passwords to attack other users' passwords on the same pictures. In the same paper [38], they presented another approach which creates dictionaries by predicting hot-spots using image processing methods. Oorschot et al. [27] cracked DAS using some password complexity factors, such as reflective symmetry and stroke-count. Salehi-Abari et al. [32] proposed an automated attack on the PassPoints scheme by ranking passwords with click-order patterns. However, the click-order patterns introduced in their approach could not capture users' selection processes accurately, especially when a background image significantly affects user choice.

8 Conclusion

We have presented a novel attack framework against background draw-a-secret schemes with special attention on picture gesture authentication. We have described an empirical analysis of Windows 8^{TM} picture gesture authentication based on our user studies. Using the proposed attack framework, we have demonstrated that our approach was able to crack a considerable portion of picture passwords in various situations. We believe the findings and attack results discussed in this paper could advance the understanding of background draw-a-secret and its potential attacks.

Acknowledgements

The authors are grateful to Lujo Bauer of Carnegie Mellon University and Sonia Chiasson of Carleton University for useful comments while this work was in progress. The authors also thank the anonymous reviewers whose comments and suggestions have significantly improved the paper.

References

[1] Opencv. http://opencv.willowgarage.com.

[2] Alexe, B., Deselaers, T., and Ferrari, V. Objectness measure v1.5. http://groups.inf.ed.ac.uk/calvin/objectness/objectness-release-v1.5.tar.gz.

[3] Alexe, B., Deselaers, T., and Ferrari, V. Measuring the objectness of image windows. *IEEE Transactions Pattern Analysis and Machine Intelligence* (2012).

[4] Aviv, A., Gibson, K., Mossop, E., Blaze, M., and Smith, J. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX conference on Offensive technologies* (2010), USENIX Association, pp. 1–7.

[5] Ballard, D. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition 13*, 2 (1981), 111–122.

[6] Bicakci, K., Atalay, N., Yuceel, M., Gurbaslar, H., and Erdeniz, B. Towards usable solutions to graphical password hotspot problem. In *Proceedings of the 33rd Annual IEEE International on Computer Software and Applications Conference* (2009), vol. 2, IEEE, pp. 318–323.

[7] Biddle, R., Chiasson, S., and Van Oorschot, P. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys 44*, 4 (2011), 2012.

[8] Bonneau, J., Preibusch, S., and Anderson, R. A birthday present every eleven wallets? the security of customer-chosen banking pins. *Financial Cryptography and Data Security* (2012), 25–40.

[9] Brostoff, S., and Sasse, M. Are passfaces more usable than passwords? a field trial investigation. *People And Computers* (2000), 405–424.

[10] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (1986), 679–698.

[11] Castelluccia, C., Dürmuth, M., and Perito, D. Adaptive password-strength meters from markov models. In *Proceedings of the 19th Network and Distributed System Security Symposium* (2012), vol. 2012.

[12] Chiasson, S., Forget, A., Biddle, R., and van Oorschot, P. User interface design affects security: Patterns in click-based graphical passwords. *International Journal of Information Security 8*, 6 (2009), 387–398.

[13] Chiasson, S., Stobert, E., Forget, A., Biddle, R., and Van Oorschot, P. Persuasive cued click-points: Design, implementation, and evaluation of a knowledge-based authentication mechanism. *IEEE Transactions on Dependable and Secure Computing 9*, 2 (2012), 222–235.

[14] Chiasson, S., van Oorschot, P., and Biddle, R. Graphical password authentication using cued click points. Springer, pp. 359–374.

[15] Davis, D., Monrose, F., and Reiter, M. On user choice in graphical password schemes. In *Proceedings of the 13th conference on USENIX Security Symposium* (2004), USENIX Association, pp. 11–11.

[16] Dhamija, R., and Perrig, A. D*éjà* vu: A user study using images for authentication. In *Proceedings of the 9th conference on USENIX Security Symposium* (2000), USENIX Association.

[17] Dirik, A. E., Memon, N., and Birget, J.-C. Modeling user choice in the passpoints graphical password scheme. In *Proceedings of the 3rd symposium on Usable privacy and security* (2007), ACM, pp. 20–28.

[18] Dunphy, P., and Yan, J. Do background images improve draw a secret graphical passwords? In *Proceedings of the 14th ACM conference on Computer and communications security* (2007), ACM, pp. 36–47.

[19] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[20] Feige, U., Lovász, L., and Tetali, P. Approximating min sum set cover. *Algorithmica 40*, 4 (2004), 219–234.

[21] Foley, M. J. Microsoft: 60 million windows 8 licenses sold to date. http://www.zdnet.com/microsoft-60-million-windows-8-licenses-sold-to-date-7000009549/, 2013.

[22] Forget, A., Chiasson, S., and Biddle, R. Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In *Proceedings of the 28th international conference on Human factors in computing systems* (2010), ACM, pp. 1107–1110.

[23] Gao, H., Guo, X., Chen, X., Wang, L., and Liu, X. Yagp: Yet another graphical password strategy. In *Proceedings of the 24th Annual Computer Security Applications Conference* (2008), IEEE, pp. 121–129.

[24] Jermyn, I., Mayer, A., Monrose, F., Reiter, M., and Rubin, A. The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium* (1999), Washington DC, pp. 1–14.

[25] Johnson, J. Picture gesture authentication, US Patent 163201, 2012.

[26] Kelley, P., Komanduri, S., Mazurek, M., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L., and Lopez, J. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proceedings of the IEEE Symposium on Security and Privacy* (2012), IEEE, pp. 523–537.

[27] Oorschot, P., and Thorpe, J. On predictive models and user-drawn graphical passwords. *ACM Transactions on Information and system Security (TISSEC) 10*, 4 (2008), 5.

[28] Ovide, S. Microsoft's windows 8 test: Courting consumers. http://online.wsj.com/article/SB10001424052970204530504578078743616727514.html.

[29] Pace, Z. Signing in with a picture password. http://blogs.msdn.com/b/b8/archive/2011/12/16/signing-in-with-a-picture-password.aspx.

[30] Pace, Z. Signing into windows 8 with a picture password. http://www.youtube.com/watch?v=Ek9N2tQzHOA.

[31] Reimondo, A. Haar cascades. http://alereimondo.no-ip.org/OpenCV/34.

[32] Salehi-Abari, A., Thorpe, J., and Van Oorschot, P. On purely automated attacks and click-based graphical passwords. In *Proceedings of the 24th Annual Computer Security Applications Conference* (2008), IEEE, pp. 111–120.

[33] Schechter, S. E., Dhamija, R., Ozment, A., and Fischer, I. The emperor's new security indicators. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy* (2007), IEEE, pp. 51–65.

[34] Suo, X., Zhu, Y., and Owen, G. Graphical passwords: A survey. In *Proceedings of the 21st Annual Computer Security Applications Conference* (2005), IEEE, pp. 10–19.

[35] Suzuki, S., et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing 30*, 1 (1985), 32–46.

[36] Thorpe, J., and Van Oorschot, P. Graphical dictionaries and the memorable space of graphical passwords. In *Proceedings of the 13th conference on USENIX Security Symposium* (2004), USENIX Association, pp. 10–10.

[37] Thorpe, J., and Van Oorschot, P. Towards secure design choices for implementing graphical passwords. In *Proceedings of the 20th Annual Computer Security Applications Conference* (2004), IEEE, pp. 50–60.

[38] Thorpe, J., and van Oorschot, P. Human-seeded attacks and exploiting hot-spots in graphical passwords. In *Proceedings of 16th USENIX Security Symposium* (2007), USENIX Association, p. 8.

[39] van Oorschot, P., and Thorpe, J. Exploiting predictability in click-based graphical passwords. *Journal of Computer Security 19*, 4 (2011), 669–702.

[40] Viola, P., and Jones, M. Robust real-time face detection. *International journal of computer vision 57*, 2 (2004), 137–154.

[41] Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., and Memon, N. Authentication using graphical passwords: effects of tolerance and image choice. In *Proceedings of the Symposium on Usable privacy and security* (2005), ACM, pp. 1–12.

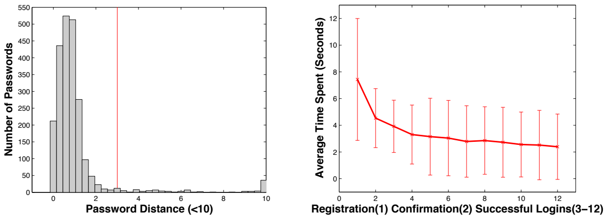
[42] Yuille, J. C. *Imagery, memory, and cognition*. Lawrence Erlbaum Assoc Inc, 1983.

[43] Zakaria, N., Griffiths, D., Brostoff, S., and Yan, J. Shoulder surfing defence for recall-based graphical passwords. In *Proceedings of the 7th Symposium on Usable Privacy and Security* (2011), ACM, p. 6.

[44] Zhang, Y., Monrose, F., and Reiter, M. 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), ACM, pp. 176–186.

A Memorability and Usability Analysis

The tolerance introduced in PGA is a trade-off between security and usability. In order to quantify this tradeoff, we calculate the distance between input PGA passwords with the registered ones. When the types or directions of gestures do not match, we regard input passwords incomparable with the registered ones. Otherwise, the distance



(a) Password Distance Histogram

(b) Average Time Spent (Seconds)

Figure 12: Memorability and Usability

is defined as the average distance of all gestures. We denote the password presented for the i-th attempt $\vec{\pi}^{(i)}$ and $\vec{\pi}^{(0)}$ as the password registered for the same picture.

In the 2,536 login attempts collected in *Dataset-1*, 422 are unsuccessful in which 146 are type or direction errors and 276 are distance errors. Figure 12(a) shows the distance distribution for the password whose distance is less than 10 and the red line denotes the threshold for being classified as successful. The result shows the current setup in our system is quite reasonable to capture most closely presented passwords.

Figure 12(b) shows the average time in seconds that subjects spent on registering, confirming, and reproducing passwords. $x = 1$ denotes the registration, $x = 2$ denotes the conformation, and all others denote the later login attempts. As we can notice, the average time for the registration is 7.43 seconds while 4.53 seconds are taken for the confirmation. With subjects getting used to the picture password system, the average time spent for successful logins is reduced to as low as 2.51 seconds. On the other hand, the average time spent on all unsuccessful login attempts is 5.86 seconds.

B Dataset-2 Pictures

Figure 13 shows 15 images that are used in *Dataset-2* as the background pictures for password selection.

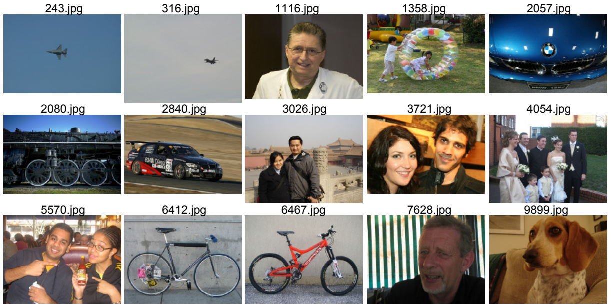


Figure 13: Background Pictures Used in *Dataset-2*

C LdGSF Identification

We discuss the identified LdGSFs by linking PoIs and passwords in *Dataset-2* with the help of two PoI sets $P^2_{L\text{-}15}$ and $P^2_{A\text{-}40}$ using our LdGSF identification algorithm discussed in Section 4.3. The results from P_L are closer to users' actual selection processes, while the results from P_A are the best approximations to users' selection processes we could get in a purely automated way with state-of-the-art computer vision techniques.

Table 5: Top 10 Identified LdGSFs using $P^2_{L\text{-}15}$

Rank	$Pr(s_k)$	$Pr(s_k \mid D_{s_k} \subseteq \theta)$
1	$(\text{tap}, \{\text{head}\}, \varnothing)$	$(\text{tap}, \{\text{nose}\}, \varnothing)$
2	$(\text{tap}, \{*_c\}, \varnothing)$	$(\text{tap}, \{\text{mouth}\}, \varnothing)$
3	$(\text{tap}, \{\text{circle}\}, \varnothing)$	$(\text{tap}, \{\text{circle}\}, \varnothing)$
4	$(\text{tap}, \{\text{eye}\}, \varnothing)$	$(\text{tap}, \{\text{eye}\}, \varnothing)$
5	$(\text{circle}, \{\text{head}\}, \varnothing)$	$(\text{tap}, \{*_c\}, \varnothing)$
6	$(\text{tap}, \{\text{nose}\}, \varnothing)$	$(\text{tap}, \{\text{head}\}, \varnothing)$
7	$(\text{circle}, \{\text{circle}\}, \varnothing)$	$(\text{circle}, \{\text{circle}\}, \varnothing)$
8	$(\text{circle}, \{\text{eye}\}, \varnothing)$	$(\text{tap}, \{\text{ear}\}, \varnothing)$
9	$(\text{line}, \{*_c\}, \{*_c\})$	$(\text{line}, \{\text{mouth}\}, \{\text{mouth}\})$
10	$(\text{line}, \{\text{eye}\}, \{\text{eye}\})$	$(\text{tap}, \{\text{forehead}\}, \varnothing)$

The top ten identified LdGSFs using $P^2_{L\text{-}15}$ are shown in Table 5 ordered by their $Pr(s_k)$ and $Pr(s_k|D_{s_k} \subseteq \theta)$. It also suggests that 'tap a head' is found the most times in the passwords, while 'tap a nose' is the most popular one when there is a nose in the picture. The result seems unreasonable at the first glance since there is always a nose in a head. Actually, it is because if the head in the picture is really small, we simply annotate the circumscribed rectangle as head instead of marking the inner rectangles with more specific attributes. Table 5 indicates that gestures on human organs are the most popular selection functions adopted by subjects.

Table 6: Top 10 Identified LdGSFs using $P^2_{A\text{-}40}$

Rank	$Pr(s_k)$	$Pr(s_k \mid D_{s_k} \subseteq \theta)$
1	$(\text{tap}, \{\text{circle}\}, \varnothing)$	$(\text{tap}, \{\text{clock}\}, \varnothing)$
2	$(\text{tap}, \{\text{mouth}\}, \varnothing)$	$(\text{circle}, \{\text{clock}\}, \varnothing)$
3	$(\text{tap}, \{\text{eye}\}, \varnothing)$	$(\text{tap}, \{\text{shoulder}\}, \varnothing)$
4	$(\text{tap}, \{\text{head}\}, \varnothing)$	$(\text{tap}, \{\text{eye}\}, \varnothing)$
5	$(\text{tap}, \{*_c\}, \varnothing)$	$(\text{tap}, \{\text{head}\}, \varnothing)$
6	$(\text{tap}, \{*\}, \varnothing)$	$(\text{tap}, \{\text{body}\}, \varnothing)$
7	$(\text{circle}, \{\text{eye}\}, \varnothing)$	$(\text{tap}, \{\text{mouth}\}, \varnothing)$
8	$(\text{tap}, \{\text{body}\}, \varnothing)$	$(\text{tap}, \{\text{circle}\}, \varnothing)$
9	$(\text{circle}, \{\text{circle}\}, \varnothing)$	$(\text{tap}, \{*\}, \varnothing)$
10	$(\text{circle}, \{\text{head}\}, \varnothing)$	$(\text{tap}, \{*_c\}, \varnothing)$

The top ten identified LdGSFs using $P^2_{A\text{-}40}$ are shown in Table 6. By comparing Table 5 and Table 6, we could notice differences caused by using annotated PoI set and automated detected PoI set. The fact that $s(\text{tap}, \{*\}, \varnothing)$ is among the top ten LdGSFs is an indicator that the automatic PoI identification could not classify many PoIs and simply mark them as $*$. It is surprising to find out there are two LdGs on clock in top ten ordered by $Pr(s_k|D_{s_k} \subseteq \theta)$ at first, because there is no clock in any picture in *Dataset-2*. The closest guess is OpenCV falsely identified some circle shape objects as clocks, but the number is not very big since there is no LdG on a clock in the top ten ordered by $Pr(s_k)$.