Picture Gesture Authentication: Empirical Analysis, Automated Attacks, and Scheme Evaluation

ZIMING ZHAO and GAIL-JOON AHN, Arizona State University
HONGXIN HU, Clemson University

Picture gesture authentication has been recently introduced as an alternative login experience to text-based password on touch-screen devices. In particular, the newly on market Microsoft Windows 8^{TM} operating system adopts such an alternative authentication to complement its traditional text-based authentication. We present an empirical analysis of picture gesture authentication on more than 10,000 picture passwords collected from more than 800 subjects through online user studies. Based on the findings of our user studies, we 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' thought processes in selecting picture passwords. Our evaluation results show the proposed approach could crack a considerable portion of picture passwords under different settings. Based on the empirical analysis and attack results, we comparatively evaluate picture gesture authentication using a set of criteria for a better understanding of its advantages and limitations.

Categories and Subject Descriptors: D.4.6 [**Operating Systems**]: Security and Protection

General Terms: Security

Additional Key Words and Phrases: Picture gesture authentication, empirical analysis, automated attacks, scheme evaluation

ACM Reference Format:
Ziming Zhao, Gail-Joon Ahn, and Hongxin Hu. 2015. Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation. ACM Trans. Info. Syst. Sec. 17, 4, Article 14 (April 2015), 37 pages.
DOI: http://dx.doi.org/10.1145/2701423

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 Personal Identification Number (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 [Bonneau et al. 2012d].

The work of Ziming Zhao and Gail-Joon Ahn was partially supported by grants from the Global Research Laboratory Project through the National Research Foundation (NRF-2014K1A1A2043029).
Authors' addresses: Z. Zhao, Arizona State University, Tempe, AZ, USA 85281; email: ziming.zhao@asu.edu; G.-J. Ahn (corresponding author), Arizona State University, Tempe, AZ, USA 85281; email: gahn@asu.edu; H. Hu, Clemson University, Clemson, USA 29634; email: hongxih@clemson.edu.
A preliminary version of this paper appears in *Proceedings of the 22nd Usenix Security Symposium, 2013*.

ACM 1094-9224/2015/04-ART14 $15.00
DOI: http://dx.doi.org/10.1145/2701423

Many graphical password schemes—including DAS [Jermyn et al. 1999], Face [Brostoff and Sasse 2000], Story [Davis et al. 2004], PassPoints [Wiedenbeck et al. 2005a], and BDAS [Dunphy and Yan 2007]—have been proposed in the past decade (for more, please refer to Dhamija and Perrig [2000], Thorpe and Van Oorschot [2004], Suo et al. [2005], Chiasson et al. [2007], Gao et al. [2008], Bicakci et al. [2009], Biddle et al. [2011], and Chiasson et al. [2012]). As an outcome of these research efforts, the Windows 8^{TM} operating system comes with a picture password authentication system, namely Picture Gesture Authentication (PGA) [Johnson et al. 2012], which is an instance of Background Draw-a-Secret (BDAS) schemes [Dunphy and Yan 2007]. This new authentication mechanism hit the market with miscellaneous computing devices including personal computers and tablets [Microsoft 2013]. Consequently, it is imperative to examine the user experiences with and potential attacks on this new scheme to understand its advantages and limitations.

To understand user experiences in PGA, we collected more than 10,000 PGA passwords from more than 800 subjects through online user studies within a span of several months. Here, we provide an empirical analysis of the collected passwords. In particular, we are interested in how subjects choose background pictures, where they prefer to draw gestures, and what gesture orders and types they like to use. Our findings from user-chosen passwords show interesting patterns that are consistent with previous research investments on click-based password schemes [Chiasson et al. 2009; Van Oorschot et al. 2010; van Oorschot and Thorpe 2011], in which password composition patterns and predictable characteristics were found. In addition, we present memorability analysis results on passwords that were collected over months.

Harvesting characteristics from passwords of a target picture and exploiting hotspots and geometric patterns on the target picture have proved effective for attacking click-based schemes [Dirik et al. 2007; Thorpe and Van Oorschot 2007; Salehi-Abari et al. 2008]. 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 possible) to work on previously unseen target pictures.

To attack PGA passwords, 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, we build generic knowledge of user choices 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 designed 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.

To further examine the benefits and limitations of PGA, we evaluate if it also provides benefits that other authentication schemes offer based on results from user experience studies and attack evaluations. We consider four categories of criteria: Usability, Deployability, Security, and Privacy (UDSP). Our evaluation criteria are extended from the Usability-Deployability-Security (UDS) evaluation framework [Bonneau et al. 2012b],

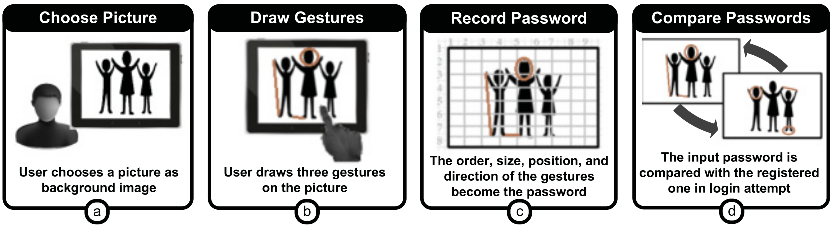

Choose Picture
User chooses a picture as background image
a
Draw Gestures
User draws three gestures on the picture
b
Record Password
The order, size, position, and direction of the gestures become the password
c
Compare Passwords
The input password is compared with the registered one in login attempt
d

Fig. 1. Key steps in picture gesture authentication.

which was designed to evaluate web authentication schemes. To explain the newly introduced benefits, we evaluate four legacy authentication schemes: text-based passwords, Persuasive Cued Click-Points (PCCP) [Chiasson et al. 2012], Fingerprint, and RSA SecurID. We also evaluate and compare two other popular authentication schemes on touch-screens, namely, draw pattern and PIN, using our extended evaluation framework.

The contributions of this article are summarized as follows:

—We compile two datasets of PGA usage from user studies[1] 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.
—We propose and implement a novel attack framework based on selection functions. We evaluate our attack framework using two attack models: namely, nontargeted attack and targeted attack.
—We comparatively evaluate PGA using a new UDSP evaluation framework that is extended from the UDS authentication evaluation framework by considering more usability, security, and privacy benefits.

The remainder of the article is organized as follows. Section 2 gives an overview of picture gesture authentication. Section 3 discusses our empirical analysis on passwords of PGA that were collected from two online studies. In Section 4, we illustrate the idea of using selection functions to model users' password creation processes and build an attack framework based on it. Section 5 presents the implementation details of our proposed attack framework. Section 6 presents the evaluation results of nontargeted attacks. Section 7 presents the evaluation results of targeted attacks. Section 8 presents a UDSP framework and comparative evaluation results of PGA. We discuss several research issues in Section 9, followed by the related work in Section 10. Section 11 concludes the article.

2. AN OVERVIEW OF PICTURE GESTURE AUTHENTICATION

Figure 1 shows the key steps of using PGA. Like other login systems, Windows 8^{TM} PGA has two independent phases: registration and authentication. In the registration stage, a user chooses a picture from his or her local storage as the background, as shown in Figure 1(a). 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^{TM}, the probability of different users choosing 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

[1] These datasets with detailed information are available at http://sefcom.asu.edu/pga/.

Table I. Password Space Comparison with Different Schemes

Length	Draw Pattern*	4-digit PINs$^{*\star\bullet}$	Text-based Password$^{*\star\bullet}$
1	9	10	90
2	56	100	8,100
3	360	1,000	729,000
4	2,280	10,000	65,610,000
5	14,544	100,000	5,904,900,000

Used in * Android, * iOS, $^{\bullet}$ Windows 8.

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, as illustrated in Figure 1(b). 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) [Pace 2011a]. 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, as shown in Figure 1(c).

Once a picture password is successfully registered, the user may login to the system by drawing corresponding gestures instead of typing his or her text-based password. PGA first brings the background image on the screen that the user chose in the registration stage. Then, the user reproduces 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, as shown in Figure 1(d). 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 $12 - d^2 \geq 0$ is satisfied, where d denotes the distance between the tap coordinates and the stored coordinates. The starting and ending points of line gestures and the center of circle gestures are measured with the same predicate [Pace 2011a].

The differences between PGA and the first BDAS scheme proposed in Dunphy and Yan [2007] 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, whereas 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.

Accurate estimation of the PGA password space needs some detailed information, such as the circle radius tolerance, that is not disclosed. Therefore, the password space calculation presented here is taken from Pace [2011a], where Pace et al. quantified the size of theoretical password space of PGA and compared it with other password schemes. As shown in Table I, the password space for PGA is much bigger than other schemes, given the same password length. Pace et al. also considered those cases in which users only draw on some point-of-interests in the picture. Table II shows the password space with different numbers of point-of-interests. If a picture has

Table II. Password Space Comparison with PGA

Length	Picture Password$^{\bullet}$ [Pace 2011a]			
	tap	line	circle	combined
1	270	335	1,949	2,554
2	23,535	34,001	846,183	1,581,773
3	2,743,206	4,509,567	412,096,718	**1,155,509,083=$2^{30.1}$**
4	178,832,265	381,311,037	156,687,051,477	612,157,353,732
5	15,344,276,658	44,084,945,533	70,441,983,603,740	398,046,621,309,172

20 point-of-interests, its password space is $2^{27.7}$ which is larger than text-based passwords with a length *four*.

3. AN EMPIRICAL ANALYSIS OF PICTURE GESTURE AUTHENTICATION PASSWORDS

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 that would be similar to those in text-based passwords [Yuille 1983].

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 standalone program; and (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 teaching users 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 [Kelley et al. 2012]. Turkers who had finished more than 50 tasks and had an approval rate of 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

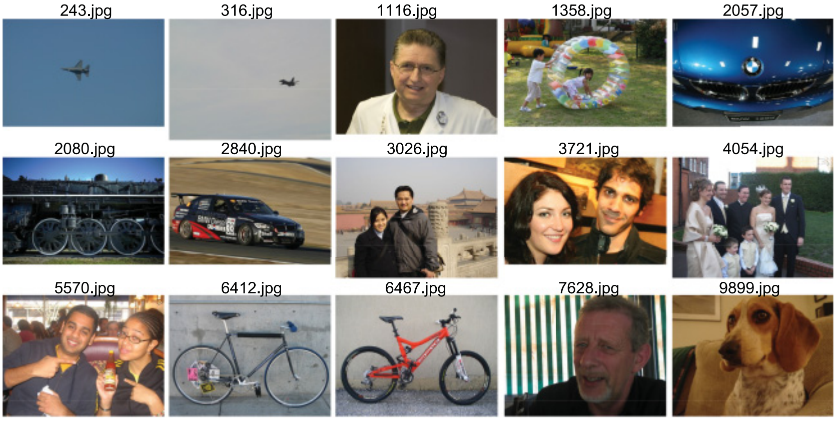


Fig. 2. Background pictures used in *Dataset-2*.

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 in advance 15 pictures, as shown in Figure 2, from the PASCAL Visual Object Classes Challenge 2007 dataset.[2] 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 on picture presentation order. A total of 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 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. Participants in *Dataset-2* are more diverse with 64.4% male, 37.2% 18–24 years old, 45.4% 25–34 years old, and 15.0% 35–50 years old. Even though the subjects in our studies do not represent all possible demographics, the data collected from them represent the most comprehensive PGA usage so far. Their tendencies could provide us with significant insights into user choice in PGA.

3.2. Findings

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

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

[2] http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

We analyzed all unique pictures[3] in *Dataset-1*, and the background pictures chosen by subjects range from celebrities to system screenshots. We categorized them into six classes: people (27/58), civilization (7/58), landscape (3/58), computer-generated picture (14/58), animals (6/58), and 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 toward using picture of people. The advocates for such pictures considered it more friendly (e.g., "*The image was special to me so I enjoy seeing it when I log in*"), easier for remembering passwords (e.g., "*Marking points on a person is easier to remember*"), and as making passwords more secure (e.g., "*The picture is personal so it should be much harder for someone to guess the password*"). However, other participants believed their choice could leak their identity or privacy (e.g., "*revealing myself or my family to anyone who picks up the device*"). They preferred other types of pictures because it was "*less personal if someone gets my picture*" and "*landscape usually doesn't have any information about who you are*."

Fourteen 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 that 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 that the population characteristics (male, aged 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 that may reveal the identity or persona of a device owner. Traditional text-based passwords do 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.

Finding 2. Gestures on Points of Interest.

The security of BDAS schemes mostly relies on the location distribution of users' gestures. It is 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 to users. By analyzing the collected passwords, we noticed that subjects frequently chose standout regions (Points of Interest, PoIs) on which to draw. As shown in Table III, 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 larger than text-based passwords of 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.,

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

Table III. Survey Question: Which of the Following Best Describes What You Are Considering When You Choose Locations to Perform Gestures?

Multi-choice Answers	Dataset *1*	Dataset *2*	Dataset Overall
I try to find locations where special objects are, such as head, eye, clock, car, badge, etc.	24 (72.7%)	389 (59.6%)	413 (60.3%)
I try to find locations where some special shapes are, such as circle and line, etc.	8 (24.2%)	143 (21.9%)	151 (22.1%)
I try to find locations where colors are different from their surroundings, such as a red apple in a green lemon pile, etc.	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%)

Table IV. 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

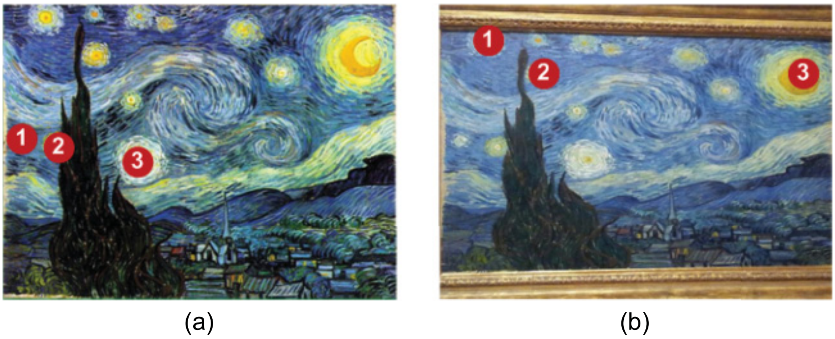


Fig. 3. Two versions of *Starry Night* and corresponding passwords.

"*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 III: 60.3% of subjects prefer to find locations where special objects catch their eyes, whereas 22.1% of subjects would rather draw on some special shapes.

Finding 3. Similarities Across Points of Interest.

We analyzed the attributes of PoIs that users preferred to draw on. We paid more attention to pictures of people because this 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 IV, 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, whereas 10 subjects (45.4%) performed gestures on noses. The tendencies among different subjects to choose similar PoIs are common in other picture categories as well. Figure 3 shows another example where two subjects

Table V. 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%

Table VI. Numbers of Gesture Type Combinations and Average Time Spent on Creating Them

		3×t	3×l	3×c	2×t+l	2×t+c
Dataset-1	#	60	3	0	9	1
	Average Time (Seconds)	5.74	12.39	N/A	10.12	21.56
Dataset-2	#	3438	1447	253	1211	380
	Average Time (Seconds)	4.33	7.11	9.96	6.02	6.14

		2×l+t	2×l+c	2×c+t	2×c+l	t+l+c
Dataset-1	#	7	1	0	0	5
	Average Time (Seconds)	11.17	17.51	N/A	N/A	11.22
Dataset-2	#	1000	622	192	442	1054
	Average Time (Seconds)	7.72	9.98	8.78	10.19	9.37

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.

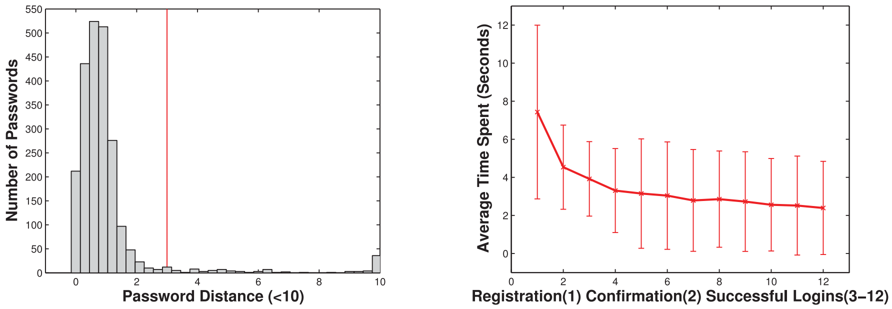
Finding 4. Directional Patterns in PGA Password.

Salehi-Abari et al. [2008] suggest that 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: H+, denoting left-to-right ($x_i \leq x_{i+1}$); H-, denoting right-to-left ($x_i \geq x_{i+1}$); V+, denoting top-to-bottom ($y_i \leq y_{i+1}$); and 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 V, more than half of all 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, with 50.0% and 31.3% occurrences in *Dataset-1* and *Dataset-2*, respectively. V+ (drawing from top to bottom) was the second most popular, with 18.6% and 14.7% occurrences in the two datasets, respectively. This finding shows that it is reasonable to use gesture-order patterns as one heuristic factor to prioritize generated passwords.

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 did subjects in *Dataset-1*. As shown in Table VI, 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



(a) Password Distance Histogram

(b) Average Time Spent (Seconds)

Fig. 4. Memorability and usability.

datasets with the lowest average time (5.74 seconds and 4.33 seconds in corresponding datasets). On the other hand, 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 passwords by pretending they were protecting their bank information. However, subjects in *Dataset-1* actually used these passwords to access class materials that they accessed more than four times a week on average. This may be a reason why subjects in *Dataset-1* preferred passwords with simpler gesture type combinations that are easier to reproduce quickly.

3.3. Memorability and Usability Analysis

The tolerance introduced in PGA is a tradeoff between security and usability. 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 is defined as the average distance of all gestures.

In the 2,536 login attempts collected in *Dataset-1*, 422 were unsuccessful, in which 146 are type or direction errors and 276 are distance errors. Figure 4(a) shows the distance distribution for those passwords whose distance is less than 10, and the red line denotes the threshold for being classified as successful. The result shows that the current setup in our system is quite reasonable to capture most closely presented passwords.

Figure 4(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. Note that the average time for registration is 7.43 seconds, whereas 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.

4. ATTACK FRAMEWORK

In this section, we present an attack framework on Windows 8^{TM} PGA, 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, ranked in descending order of password probabilities.

Next, we 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. In *Pure Brute-force Attack*, an attacker blindly guesses the picture password without knowing any information about the background picture and the users' tendencies. The password space in this model is $2^{30.1}$ in PGA [Pace 2011a]. In the *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 $2^{27.7}$ [Pace 2011a]. Salehi-Abari et al. [2008] designed an approach to automatically identify hotspots in a picture and generate passwords on them. In the *Knowledge-based PoI-assisted Attack*, in addition to the assumption used 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 that in PoI-assisted brute-force attack. However, the generated dictionaries in this model are ranked with the higher possibility passwords at 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 Salehi-Abari et al. [2008] is able to attack previously unseen pictures for click-based graphical passwords. It uses click-order heuristics to generate partially ranked dictionaries. However, this approach cannot be applied directly to BDAS 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.

Based on the data origin the attacker harvests from users, we categorize two attack mode: in *nontargeted attack* mode, the training dataset does not consist of any picture and password pair from the target user. The guessing path carried out for a nontargeted attack is not contingent on the knowledge of target user's individual tendencies, but insteadis based on the habits of users in a training dataset. In *targeted attack* mode, the attacker has possession of some picture and password pairs collected from the target user. Hence, the guessing path is more specific to the target user. Our algorithms support both attack modes.

4.2. Password and PoI Representations

We first formalize the representation of a password in PGA with the definition of a Location-Dependent Gesture (LdG) that 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; it 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, and the like. 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 $D \subseteq 2^{\mathcal{D}}$ is a set of attributes that describe this PoI. $\mathcal{D}$ has three subcategories $\mathcal{D}_o, \mathcal{D}_s$, and $\mathcal{D}_c$ and four wildcards $*_o, *_s, *_c$, and $*$, where $\mathcal{D}_o = \{\mathsf{head}, \mathsf{eye}, \mathsf{nose}, \ldots\}$, $\mathcal{D}_s = \{\mathsf{line}, \mathsf{rectangle}, \mathsf{circle}, \ldots\}$, and $\mathcal{D}_c = \{\mathsf{red}, \mathsf{blue}, \mathsf{yellow}, \ldots\}$. Wildcards are used when no specific information is available. For example, if a PoI is identified with an objectness measure [Alexe et al. 2012] that gives no semantics about the identified region, we mark the PoI's describing attribute as $*$.

4.3. Location-Dependent Gesture Selection Functions

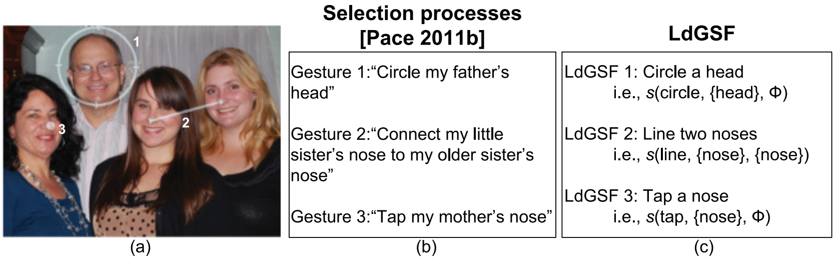
A key concept in our framework is the LdG Gesture Selection Function (LdGSF), which models and simulates the ways of thinking that users go through when they select a gesture on a picture. The motivation behind this abstraction is that the set of PoIs and their locations differ from picture to picture, 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. An LdG Gesture Selection Function (LdGSF) is a mapping $s : G \times 2^{\mathcal{D}} \times 2^{\mathcal{D}} \times \Theta \to 2^{\Pi}$ that takes a gesture, two sets of PoI attributes, and a set of PoIs in the learning picture as input to produce a set of LdGs.

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 $\vec{S}$. We use Θ to denote the universal set of image PoIs and θ_k to denote the PoIs of picture p_k. $s(\mathsf{tap}, \{\mathsf{red}, \mathsf{apple}\}, \varnothing, \theta_k)$ 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 of actual locations of objects in the picture.

One challenge we face is that 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 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." Whereas $s(\mathsf{line}, \{\mathsf{mouth}\}, \{\mathsf{mouth}\}, \theta_k)$ means "connect two different mouths."

Figure 5 shows an example demonstrating how LdGSF simulates a user's selection processes that were taken from Pace [2011b]. In reality, a user's selection process on a PoI and gesture selection may be determined by some subjective knowledge and



Selection processes [Pace 2011b]

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"

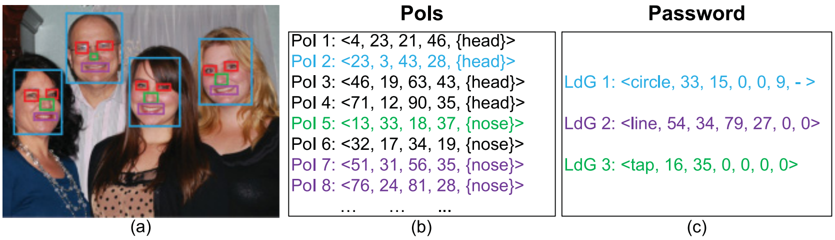
LdGSF

LdGSF 1: Circle a head
i.e., $s(\text{circle}, \{\text{head}\}, \Phi)$

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

LdGSF 3: Tap a nose
i.e., $s(\text{tap}, \{\text{nose}\}, \Phi)$

Fig. 5. (a) Background picture and password. (b) User's selection processes that were taken from (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>

Fig. 6. (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)).

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 overfitting problems into the learning stage. Instead, we choose to ignore subjective information by abstracting "circle my father's head" to "circle a head." A drawback to this abstraction is that an LdGSF may return more than one LdG, and we have no knowledge to rank them directly because they come from the same LdGSF. Using Figure 5(a) as an example, "circle a head" outputs four different LdGs on each head in the picture. The LdGSF sequence shown in Figure 5(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, which 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.

Definition 3. LdGSF identification is a process that can be denoted as a function $e : \Theta \times \Pi \rightarrow 2^S$, where $\forall s \in e(\theta_k, \pi)$, $\pi \in s(\theta_k)$. The function takes the PoIs of a picture and one LdG in its corresponding password as input and generates a set of LdGSFs that could reproduce the same LdG on the picture.

Figure 6 shows an example demonstrating 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 the PoI's attributes. For instance, the location and size of LdG 1 in Figure 6(c) matches PoI 2 in Figure 6(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 the PoI's height and width). Then, an LdGSF $s(\mathsf{circle}, \{\mathsf{head}\}, \varnothing)$ is created that is equivalent to the LdG shown in Figure 5(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 $\mathsf{order}\colon \vec{S} \rightarrow \{1..|\vec{S}|\}$ be a bijection that indicates the order in which LdGSF sequences should be performed. The objective of generating a ranked LdGSF sequence list is to find such a bijection.

4.4. LdGSF Sequence List Generation and Ordering

Now, we present our approach to finding the aforementioned bijection that indicates the order in which the LdGSF sequences should be performed on a target picture to generate 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 $\mathcal{B}_{\mathsf{mssc}}$ [Feige et al. 2004] and $\mathcal{B}_{\mathsf{emts}}$ [Zhang et al. 2010]. 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 $\vec{s_1}, \ldots, \vec{s_n}$ and corresponding picture password $\vec{\pi}_1, \ldots, \vec{\pi}_n$, for which $\vec{s_i}(\theta_i) \ni \vec{\pi}_i, i \in \{1..n\}$ and $\theta_1, \ldots, \theta_n$ are the sets of PoIs in pictures $p_1, .., p_n$. *Question*: **E**xpected **M**in **S**election **S**earch (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 $r \leftarrow \{1..n\}$. We use $\mathsf{cover}_{emss}(k) = \min_{\vec{s}:\vec{s}(\theta_k)\ni\vec{\pi}_k}(\mathsf{order}_{emss}(\vec{s}))$ to compute the number of required guesses to break $\vec{\pi}_k$. Therefore, $\mathbb{E}(\min\{i : \vec{s_i}(\theta_r) \ni \vec{\pi}_r\}$ is equivalent to $\mathbb{E}(\mathsf{cover}_{emss}(r))$.

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" produce 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 mssc [Feige et al. 2004; Zhang et al. 2010]. Min Sum Set Cover (mssc) is formalized as follows: Given a set U and a collection $\mathcal{C}$ of subsets of U where $\bigcup_{C\in\mathcal{C}} = U$, let $\mathsf{order}_{mssc} : \mathcal{C} \rightarrow \{1..|\mathcal{C}|\}$ be a bijection, and let $\mathsf{cover}_{mssc} : U \rightarrow \{1..|\mathcal{C}|\}$ be defined by $\mathsf{cover}_{mssc}(j) = \min_{C\ni j}(\mathsf{order}_{mssc}(C))$. The problem is called min sum because the object is to minimize $\sum_{j\in U} \mathsf{cover}_{mssc}(j)$.

Given any instance $(U, \mathcal{C})$ of mssc, denote $U = \{1..n\}$. We create a set of PoIs θ_j and a picture password $\vec{\pi}_j$ for each $j \in U$. θ_j and $\vec{\pi}_j$ must be different from θ_k and $\vec{\pi}_k$, respectively, for any $k \neq j$. For each $C \in \mathcal{C}$, we create an LdGSF sequence $\vec{s_C}$ such that $\vec{s_C}(\theta_j) \ni \vec{\pi}_j$ if $j \in C$ and such that $\vec{s_C}(\theta_j) = \phi$ if $j \notin C$. We can always construct such an LdGSF sequence for each C by combining all $\theta_j, j \in C$ as a new PoI type in a wildcard representation. The set $\vec{S}$ consists of the set of $\vec{s_C}$ for different C. Set

$\mathsf{order}_{mssc}(C) \leftarrow \mathsf{order}_{emss}(\vec{s_C})$, then

$$\begin{aligned}
&\mathbb{E}(\mathsf{cover}_{emss}(r)) \\
&= \sum_{i=1}^{n} i \times Pr(\mathsf{cover}_{emss}(r) = i) \\
&= \sum_{i=1}^{n} i \times \frac{|k \in \{1..n\} : \mathsf{cover}_{emss}(k) = i|}{n} \\
&= \sum_{i=1}^{n} i \times \frac{|j \in U : \mathsf{cover}_{mssc}(j) = i|}{n} \\
&= \sum_{j \in U} \frac{\mathsf{cover}_{mssc}(j)}{n}
\end{aligned}$$

The number of picture passwords that are cracked for the first time at the ith guess divided by the total number of picture passwords

Therefore, order_{emss} minimizes $\mathbb{E}(\mathsf{cover}_{emss}(r))$ if and only if order_{mssc} minimizes $\sum_{j \in U} \mathsf{cover}_{mssc}(j)$. We give an approximation algorithm for emss in Algorithm 1 that is a modification from $\mathcal{B}_{\mathsf{mssc}}$ [Feige et al. 2004] and $\mathcal{B}_{\mathsf{emts}}$ [Zhang et al. 2010]. The time complexity of BestCover is $O(n^2 + |\vec{S'}|log(|\vec{S'}|))$.

ALGORITHM 1: BestCover($(\vec{s_1}, \ldots, \vec{s_n})$,$(\vec{\pi_1}, \ldots, \vec{\pi_n})$)

```
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
    $\mathsf{order}^{-1}(i) \leftarrow \vec{s_k}$, that $T_{\vec{s_k}}$ has most elements that are not included in $\bigcup_{i'<i} \mathsf{order}^{-1}(i')$;
end
return order
```

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

4.4.2. Unbiased *LdGSF Sequence List Generation.* The overfitting 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 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 the 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 selects 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 that 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:

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

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 those scenarios in which 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$^{-1}(i)$← $\vec{s_k}$, 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 click-order patterns as the heuristics for dictionary generation [Salehi-Abari et al. 2008], 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 LDGSF IDENTIFICATION

We adopted several computer vision techniques to identify sophisticated and salient objects in the images. Some techniques were implemented in Matlab code, whereas some were implemented under the OpenCV[4] framework. The computer vision techniques we adopted include:

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. We used the method described in object detection with discriminatively trained part-based models [Felzenszwalb et al. 2010]. The latest version of its Matlab code release at the time of writing [Girshick et al. 2010] includes trained models for 21 object classes that are commonly found in the images of the PASCAL Visual Object Classes Challenge 2007 dataset. These models are trained to detect airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and tv monitor objects. The actual detection is performed by calling function imgdetect of the code release with parameters threshold=-0.6. To complement this effort, we also used the Viola-Jones object detection framework [Viola and Jones 2004]. In the Viola-Jones object detection framework, each learned classifier is represented as a haar cascade. We collected 30 proven haar cascades[5] for eight different object classes including face (head), eye, nose, mouth, ear, forehead, body, and clock. The actual detection is performed by calling OpenCV API cvHaarDetectObjects with parameters scaleFactor=1.1 and minNeighbors=4.

Salient object detection and objectness measure: Since the PoIs in our attack framework could be any local image regions that could be of interest to human users, we also resort to visual saliency and salient object detection techniques, for which the analyses and comparisons of models can be found in Borji et al. [2012] and Borji et al. [2013]. We used the discriminative regional feature integration approach described in Jiang et al. [2013] for its ease of use and outstanding performance recorded in the aforementioned two analyses reports. For a complementary approach, we used objectness measure [Alexe et al. 2012], which deals with class-generic object detection. We used an objectness measure library[6] that is able to locate objects and give numerical confidence values with its results.

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 Suzuki [1985]. For circle detection, we used Canny edge detector [Canny 1986] and Hough transform algorithms [Ballard 1981].

Figure 7 displays the PoI detection results on four example pictures in *Dataset-2*. As we can see in Figure 7(b), circle detection could identify both bicycle wheels and a car badge, but its false-positive rate is a little high. Contour detection is the most robust algorithm with a low false-positive rate that could locate regions whose colors are different, as shown in Figure 7(c). The objectness measure, shown in Figure 7(d), could also identify regions whose colors and textures are different from their

[4] http://opencv.willowgarage.com.

[5] http://alereimondo.no-ip.org/OpenCV/34.

[6] http://groups.inf.ed.ac.uk/calvin/objectness/.

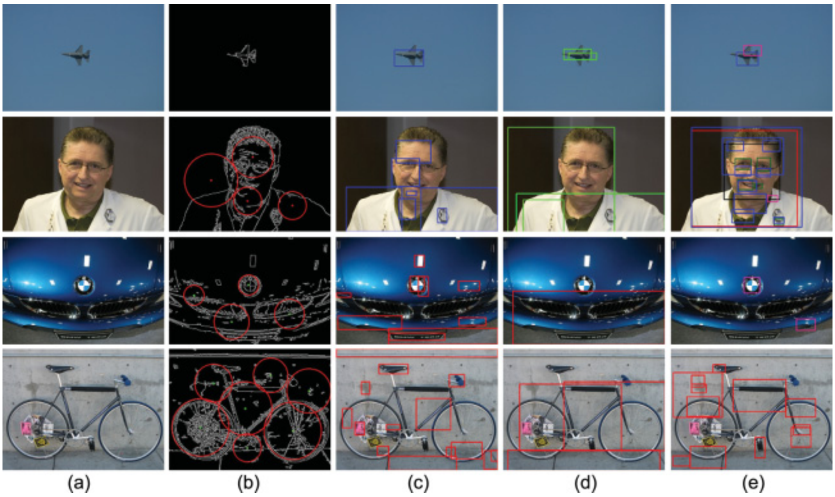


Fig. 7. PoI identification on example pictures in *Dataset-2*. (a) Original pictures. (b) Circle detection with Hough transform. (c) Contour detection. (d) Objectness measure. (e) Object detection.

surroundings. Since most haar cascades we used are designed for facial landmarks, they work smoothly on portraits, as shown in the second picture in Figure 7(e). However, the results show relatively high false-positive rates on pictures from other categories. In order to identify more PoIs as accurately 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 $P^1_{A\text{-}50}$ and $P^2_{A\text{-}50}$ for each picture in *Dataset-1* and *Dataset-2*, respectively. Those PoI sets consisted of at most 50 PoIs with the highest confidences. Compared with our previous work [Zhao et al. 2013], we identified nine more PoIs on average for each image in *Dataset-1* and four more PoIs on average for each image in *Dataset-2*.

Since our attack algorithms are independent of the PoI identification algorithms, we are also interested in examining how our attack framework performs with ideal PoI annotations for pictures. In addition to 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 15 most appealing regions in the pictures without referring to any password in the collected dataset. We call this annotated PoI set $P^2_{L\text{-}15}$.

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 $P^2_{A\text{-}50}$ using our LdGSF identification algorithm discussed in Section 4.3. The results from P_L are closer to users' actual selection processes, whereas 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.

The top 10 identified LdGSFs using $P^2_{L\text{-}15}$ are shown in Table VII, 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, whereas "tap a nose" is the most popular one when there is a nose in the picture. The result seems unreasonable at first glance since there is always a nose in a head. Actually, if the head in the picture is really small, we simply annotate the circumscribed rectangle as a head instead of marking the inner rectangles with more specific attributes. Table VII indicates that gestures on human facial landmarks are the most popular selection functions adopted by subjects.

Table VII. Top 10 Identified LdGSFs Using $P^2_{L\text{-}15}$

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

Table VIII. Top 10 Identified LdGSFs Using $P^2_{A\text{-}50}$

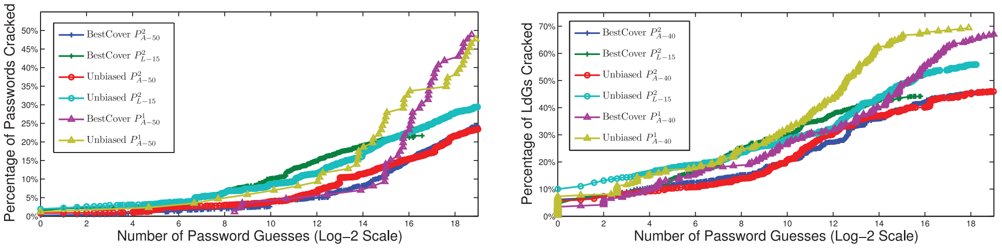
Rank	$Pr(s_k)$	$Pr(s_k\|D_{s_k} \subseteq \theta)$
1	(tap, {body}, ∅)	(tap, {clock}, ∅)
2	(tap, {circle}, ∅)	(circle, {clock}, ∅)
3	(tap, {mouth}, ∅)	(tap, {shoulder}, ∅)
4	(tap, {eye}, ∅)	(tap, {eye}, ∅)
5	(tap, {$*_c$}, ∅)	(tap, {head}, ∅)
6	(tap, {head}, ∅)	(tap, {car}, ∅)
7	(tap, {$*$}, ∅)	(tap,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

The top 10 identified LdGSFs using $P^2_{A\text{-}50}$ are shown in Table VIII. By comparing Table VII and Table VIII, we notice differences caused by using an annotated PoI set and an automated detected PoI set. The fact that $s(\text{tap}, \{*\}, \varnothing)$ is among the top 10 LdGSFs is an indicator that the automatic PoI identification could not classify many PoIs and simply marked them as $*$. It is at first surprising to find that there are three LdGs on clock in the top 10 ordered by $Pr(s_k|D_{s_k} \subseteq \theta)$ because there is no clock in any picture in *Dataset-2*. The closest guess is OpenCV, which falsely identified some circle-shape objects as clocks.

6. NONTARGETED ATTACK EVALUATION

In this section, we present the evaluation results of our framework for nontargeted attacks. 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 bits 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.

Nontargeted attacks also require that the training dataset does not include previous passwords from the targeted user. However, it turns out to be very time-consuming to perform strict nontargeted attacks on our *Dataset-2*. Instead, in our analyses, training password datasets include a very small number of passwords from the targeted subject. More specifically, in our experiment, there were around 9,400 training passwords, of which only 14 came from the targeted user. Even though this may affect the results,



(a) (b)

Fig. 8. Offline attacks with all available passwords. For *Dataset-1*, there are 86 passwords that include 258 LdGs. For *Dataset-2*, there are 10,039 passwords that have 30,117 LdGs. (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Percentage of LdGs cracked vs. number of password guesses, per condition.

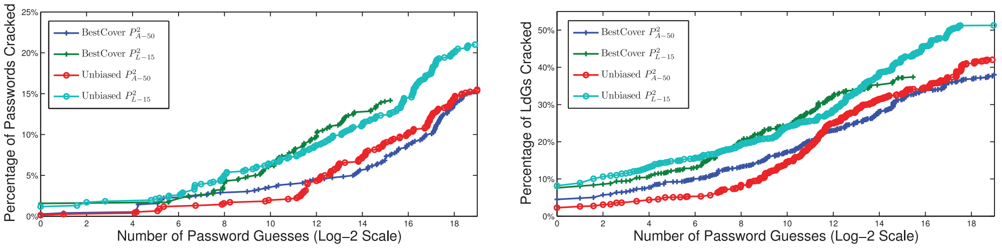
we believe it is less influential. Since all training passwords were treated equally, the influence brought by the 0.14% training data is low.

6.1. Offline Attacks

Picture passwords may be hashable using discretization methods [Jean-Camille Birget and Memon 2006]. Even though the approach that Windows 8^{TM} is adopting to store picture passwords remains undisclosed, we could consider two attack scenarios in which picture passwords are prone to offline attacks. In the first scenario, all passwords that 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, such as cracking text-based password systems. In the second scenario, picture passwords could be used for other purposes besides logging into Windows 8^{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 likely perform an offline attack.

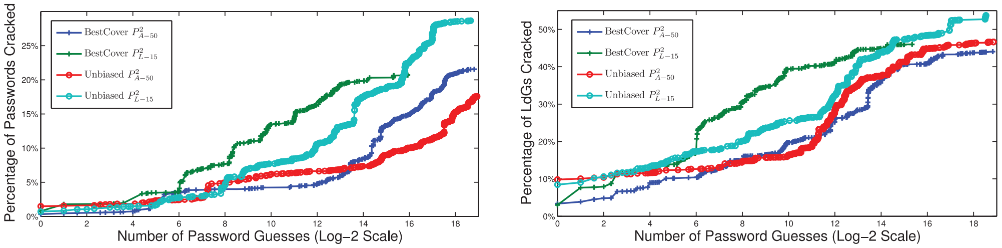
The offline attack results within 2^{19} guesses in different settings are shown in Figure 8. There are 86 passwords in *Dataset-1*, which have a total of 258 LdGs. And 10,039 passwords were collected in *Dataset-2*, containing a total of 30,117 LdGs. For *Dataset-1*, BestCover cracks 42 (48.8%) passwords out of 86, whereas Unbiased cracks 41 (47.7%) passwords for the same dataset with $P^1_{A\text{-}50}$. For *Dataset-1*, 179 LdGs (69.3%) out of 258 are cracked with Unbiased, and 173 (67.0%) 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 that BestCover with $P^2_{A\text{-}50}$ cracking 2,434 passwords (24.2%) is the best result for all purely automated attacks on *Dataset-2*. As Figure 8 suggests, BestCover outperforms Unbiased slightly when ample training data are available. The better performance of both algorithms on *Dataset-1* is because the password gesture combinations in *Dataset-1* are relatively simpler than those in *Dataset-2,* as we discussed in Section 3.2.

In *Dataset-2*, subjects may not have chosen all 15 passwords with the same care because they were eager to finish the process. To reduce this effect, we ran another analysis in which only the first password chosen by each subject was considered. There are 762 passwords that have 2,286 LdGs. Like the previous analysis, the training dataset excluded passwords from the target picture. As shown in Figure 9, 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 $P^2_{A\text{-}50}$ cracked 118 passwords (15.5%). BestCover cracks 108 (14.2%) and 116 (15.2%) with $P^2_{L\text{-}15}$ and $P^2_{A\text{-}50}$, respectively.



(a) (b)

Fig. 9. Offline attacks with only the first chosen password by each subject in *Dataset-2*. There are 762 passwords that have 2,286 LdGs. (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Percentage of LdGs cracked vs. number of password guesses, per condition.



(a) (b)

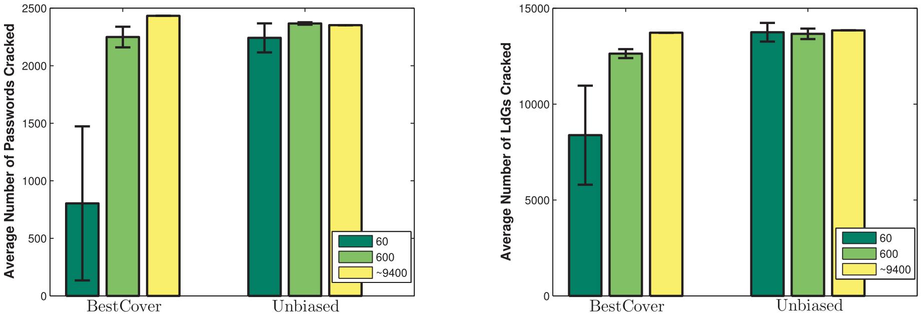
Fig. 10. Offline attacks with only passwords for pictures 243, 1116, 2057, 4054, 6467, and 9899. There are 4,003 passwords that have 12,009 LdGs. (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Percentage of LdGs cracked vs. number of password guesses, per condition.

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%), whereas 703 passwords (17.6%) are cracked by Unbiased with $P^2_{A\text{-}50}$. BestCover cracks 829 (20.7%) and 863 (21.6%) with $P^2_{L\text{-}15}$ and $P^2_{A\text{-}50}$ respectively. Results of this analysis are not as good as results with passwords from all pictures.

6.2. Effects of Training Data Size

In Figure 11, we show the password and LdG cracking results with different sizes of training datasets. For each algorithm, we used $P^2_{A\text{-}50}$ 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 in the experiments using all available training data, the training passwords in these experiments are from different pictures than the target passwords. Therefore, there is no overlap between the training passwords and the target passwords.

As Figure 11 shows, BestCover with 60 training samples could only break an average of 803 passwords (8.0%) out of 10,039; the standard deviation is as strong as 669. While Unbiased with 60 training samples can crack 2,242 passwords (22.3%), which is almost the same as the results generated from all available training samples. Also,



(a) (b)

Fig. 11. Effects of training data size. (a) Average number of passwords cracked vs. different training data sizes. (b) Average number of LdGs cracked vs. different training data sizes. $P^2_{A\text{-}50}$ is used for this analysis. Average over three analyses, with one standard deviation shown.

the standard deviation for three trials is as low as 125. The results from BestCover with 600 training samples are much better than its counterparts with 60 training samples. All these observations are expected because 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.

6.3. Effects on Different Picture Categories

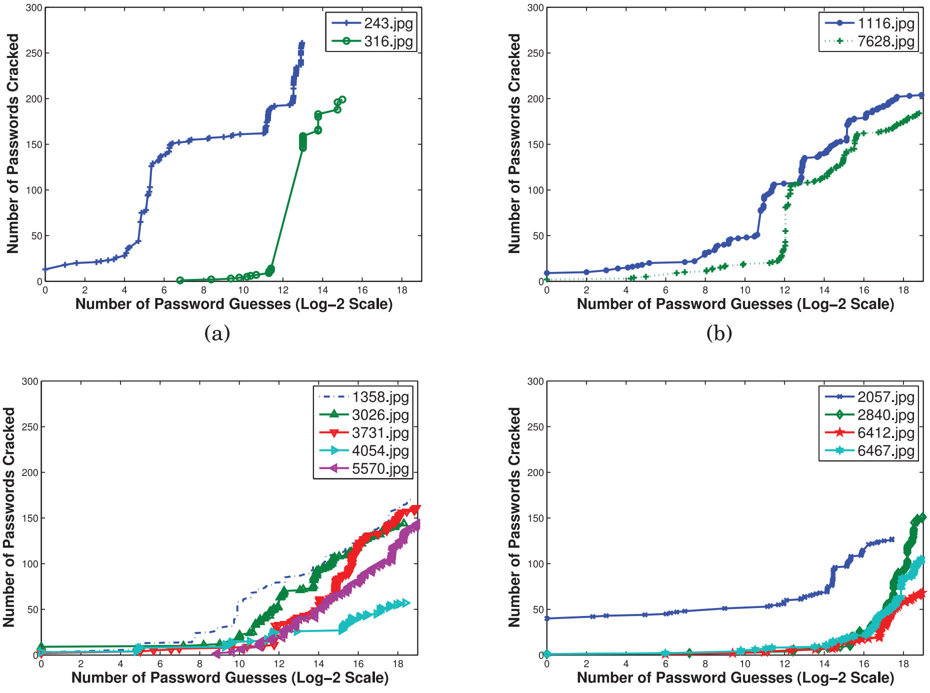
We measured the attack results on different picture categories, as shown in Figure 12, where each subfigure depicts the number of passwords cracked versus the number of password guesses. Each curve in a subfigure corresponds to a picture, as shown in the legend. Our approach cracks more passwords for 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 12(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 $P^2_{A\text{-}50}$ breaks 261 passwords (39.0%) for picture 243 and 199 for picture 316. The cracking success rates are much higher than the average success rate in *Dataset-2* under the same condition. Note that the sizes of generated dictionaries for these two pictures are smaller than 2^{19} due to the number of available PoIs.

In Figure 12(b), we show the results on two *portrait* pictures where Unbiased with $P^2_{A\text{-}50}$ cracks 388 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 to draw on these features are high. The results show that passwords on simple pictures with fewer PoIs or portraits, for which state-of-the-art computer vision techniques could detect PoIs with high accuracy, are easier for attackers to break.

Figure 12(c) shows the attack results on five 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 portraits. Unbiased with $P^2_{A\text{-}50}$ cracks 677 passwords (20.2%) for these five pictures in total, which is lower than the average success rate in *Dataset-2*.

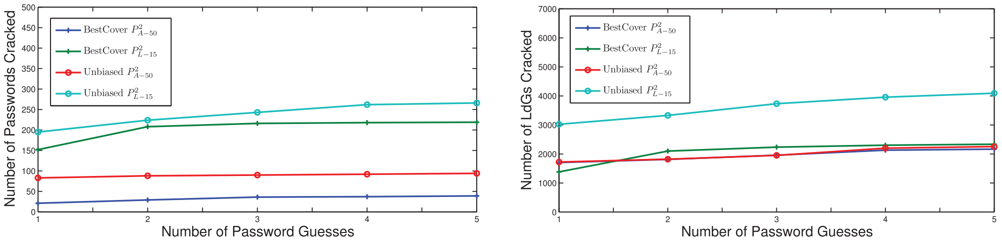
Figure 12(d) shows the attack results on four miscellaneous pictures, two of which are bicycle pictures and the other two car pictures. The picture, 6412.jpg, has a bicycle leaning against the wall. Different colors on the bicycle and wall in this picture make it cluttered and produce lots of PoIs. Unbiased with $P^2_{A\text{-}50}$ cracked 451 (17.0%) for all four pictures.



(a) (b)

(c) (d)

Fig. 12. Effects on different picture categories. (a) Pictures with fewer PoIs. (b) Portraits. (c) Pictures with people in them. (d) Pictures with lots of PoIs. Unbiased algorithm on $P^2_{A\text{-}50}$ is used for this analysis. (Please refer to Figure 2 for the pictures.)

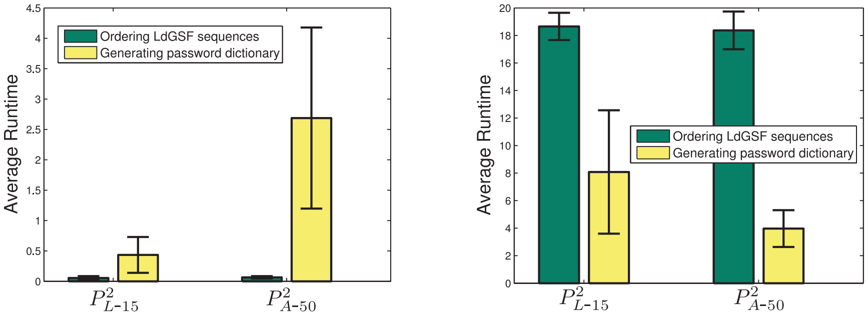


(a) (b)

Fig. 13. Online attacks with all available passwords. There are 10,039 passwords that have 30,117 LdGs in *Dataset-2*. (a) Number of passwords cracked within five guesses, per condition. (b) Number of LdGs cracked within five guesses, per condition.

6.4. Online Attacks

The current Windows 8^{TM} allows five failure attempts before it forces users to enter their text-based passwords. Therefore, breaking a password in under five guesses implies the feasibility for launching an online attack. Figure 13 shows a refined view of the number of passwords and LdGs cracked with the first five guesses per condition. Purely automated attack Unbiased with $P^2_{A\text{-}50}$ breaks 83 passwords (0.8%) with the first guess and cracks 94 passwords (0.9%) within the first five guesses, whereas BestCover with $P^2_{A\text{-}50}$ cracked 20 passwords (0.2%) with the first guess and 38 passwords (0.4%) within five guesses. Additionally, Unbiased with $P^2_{A\text{-}50}$ breaks 1,723 LdGs (5.7%) with



(a) BestCover algorithm

(b) Unbiased algorithm

Fig. 14. Average runtime in seconds to order LdGSF sequences using BestCover and Unbiased. Average over 15 pictures in *Dataset-2* with one standard deviation shown.

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%) with the first guess and 266 (2.6%) within the first five guesses. In the meantime, Unbiased 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.

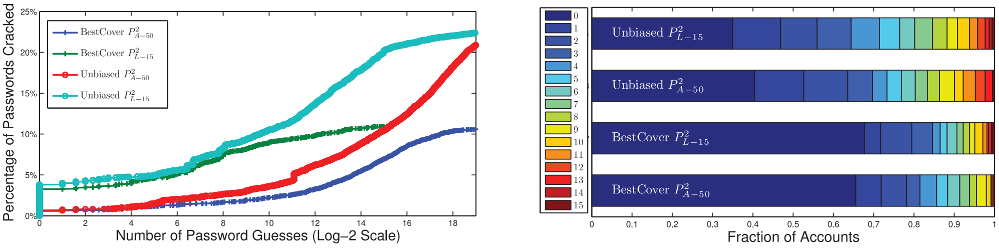
6.5. Performance

Our analyses were carried out on a computer with a dual-core processor and 4GB of RAM. In Figure 14, we show the average runtime for our algorithms to order the LdGSF sequences and generate a 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 $P^2_{A\text{-}50}$ to order LdGSF sequences is only 0.06 seconds and to generate a dictionary is 2.68 seconds, whereas 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.

7. TARGETED ATTACK EVALUATION

In this section, we present the evaluation results of our framework for targeted attacks. In targeted attacks, an attacker has possession of some picture and password pairs collected from the target user. Hence, the guessing path is more specific to the targeted user. This attack model is realistic when the user is using different passwords and background images on multiple devices. The attacker may have acquired some of the user's passwords by shoulder-surfing or using a password logger and wants to accelerate the guessing of the user's passwords for other devices by taking acquired passwords into account.

Because most subjects in *Dataset-1* only chose one password, *Dataset-1* was excluded from these experiments. We only use the passwords of those subjects who chose two or more passwords in *Dataset-2* in these experiments. There are 697 subjects who fall into this pattern, resulting in 9,974 passwords. For each of the 697 subjects, we use one of her passwords as the target and the rest of her passwords as a training dataset to build the model. The average size of training datasets is around 13, which is significantly smaller than the size used (around 9,400) in nontargeted attacks. A dictionary is generated in this way for each target password per user. Since each subject only chose one password for each picture, a training dataset does not include passwords for the target picture. We recorded the number of password guesses when a password is cracked. Then, we



(a) (b)

Fig. 15. Offline attacks. There are 9,974 passwords from 697 accounts in this experiment. The average size of training datasets is around 13. (a) Percentage of passwords cracked vs. number of password guesses, per condition. (b) Passwords cracked per account. Each horizontal bar represents a condition. Regions within each bar show the fraction of accounts for which the indicated number of passwords were cracked.

cumulated the results for each user and each target password together in a single figure, as illustrated in Figure 15.

The offline attack results within 2^{19} guesses in different settings are shown in Figure 15(a). Unbiased with $P^2_{L\text{-}15}$ breaks 2,233 passwords (22.4%) out of 9,974. Unbiased with $P^2_{A\text{-}50}$ breaks 2,083 passwords (20.9%) out of 9,974. Even though the results are a little bit lower than nontargeted attacks, we should take the significantly smaller training dataset sizes into account. In a nontargeted attack, the training data size is around 9,400 passwords. However, in a targeted attack, the training data sizes range from at least 1 password to at most 14 passwords with an average of 13. In other words, targeted attacks using Unbiased algorithms with an around 100 times smaller training data set could achieve almost the same results as nontargeted attacks. BestCover with $P^2_{L\text{-}15}$ and $P^2_{A\text{-}50}$ breaks 1,096 (10.9%) and 1,057 (10.6%) passwords, respectively. Due to the small training data size, the results from BestCover for nontargeted attacks are quite lower than its counterparts for targeted attacks.

For online attacks within five guesses (shown in the left-lower corner of Figure 15(a)), Unbiased with $P^2_{L\text{-}15}$ breaks 434 passwords (4.4%) out of 9,974, and the first guesses could even break 380 (3.8%) passwords. Unbiased with $P^2_{A\text{-}50}$ breaks 77 passwords (0.7%) out of 9,974. BestCover with $P^2_{L\text{-}15}$ breaks 351 passwords (3.5%), and BestCover with $P^2_{A\text{-}50}$ breaks 70 passwords (0.7%).

Figure 15(b) shows the fractions of the accounts for which the indicated number of passwords were cracked. Each bar represents one condition. Unbiased with $P^2_{A\text{-}50}$ cracks at least one password for 60.5% of accounts, whereas Unbiased with $P^2_{L\text{-}15}$ could crack 65.0%. Even though BestCover with $P^2_{L\text{-}15}$ cracks more passwords in total than BestCover with $P^2_{L\text{-}15}$, BestCover with $P^2_{L\text{-}15}$ breaks more accounts at least once. Both Unbiased and BestCover with $P^2_{L\text{-}15}$ crack all 15 passwords in 4 (5.7%) out of 697 accounts.

8. COMPARATIVE EVALUATION OF PGA USING UDSP FRAMEWORK

Bonneau et al. proposed a framework for comparative evaluation of web authentication schemes in Bonneau et al. [2012b]. Their evaluation framework considers scheme benefits from three different categories: usability, deployability, and security. We refer to this as the *UDS framework* for short. Since the UDS framework is designed for web authentication schemes, it does not capture some scheme benefits achieved by new devices. And it does not include comprehensive privacy benefits either. In this section, we extend the UDS framework by introducing three new usability benefits, one new security benefits, and five new privacy benefits. These newly introduced benefits are numbered after the original benefits in the UDS framework, so please refer to Bonneau et al. [2012b] for detailed explanations of the original benefits. We call the extended

Table IX. Comparative Evaluation of Schemes in Terms of Usability Benefits

	Usability benefits										
Scheme	*Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*Assistance-to-Recall**	*Touchscreen-Friendly**	*Enjoyable-to-Use**
Text-based Passwords			●		●	●	○	●			
PCCP			●		●	○	○	●	●	●	
Fingerprint	●	●	●	○	●	○			●		
RSA SecurID					●	○	○			●	
Draw Pattern*			●		●	●	○	●		●	●
4-digit PINs*			●		●	●	●	●		●	
PGA*			●		●	●	○	●	●	●	●

*Represents new benefits in UDSP framework or schemes that were not evaluated earlier.

framework a UDSP framework. As in Bonneau et al. [2012b], if a scheme partially provides a benefit, we use the *Quasi-* prefix to indicate that.

Even though some newly introduced benefits might have some correlations with the original benefits, they describe some features that cannot be covered by the UDS framework. Similar to the UDS framework, in which some benefits are subjective (e.g., *Efficient-to-Use* and *Infrequent-Errors*), some benefits we introduce here are subjective and related to users' individual choices. For instance, we introduce a usability benefit named *Enjoyable-to-Use* for which we conduct an informal survey to get opinions on different schemes. A second example is a privacy benefit named *Resilient-to-Birthday-Disclosure* that describes the possibility that the disclosure of users' credentials leads to the disclosure of their birthdays. Technically, a user could use any meaningless 4-digit number as a PIN. However, it is statistically significant that users' PINs are related to their birthdays [Bonneau et al. 2012d]. So, we do not grant PINs this benefit. Some people may argue that statistical significance has nothing to do with a scheme itself. We do not agree with this argument, and we believe it is the scheme designers' responsibility to flatten the credentials' probability distributions. This is also one of the five password system design principles discussed in Yan et al. [2012]. Some existing schemes have adopted this practice, such as those presented in Schechter et al. [2010] and Chiasson et al. [2012]. Thorpe et al. [2014] also demonstrated that the distribution of user-chosen passwords for PassPoints can be manipulated by merely changing how the background image is presented.

8.1. UDSP Framework

We explain each of the new benefits we consider by giving it a name, an actual definition, and its evaluation on four schemes. These four schemes are text-based passwords, Persuasive Cued Click-Points (PCCP) [Chiasson et al. 2012], Fingerprint, and RSA SecurID. The descriptions of these schemes and detailed evaluations of them on the original UDS framework can be found in Bonneau et al. [2012a] and Bonneau et al. [2012b]. Tables IX through XII list seven schemes, four of which were evaluated in Bonneau et al. [2012a]; the other three are new, and their evaluation is in terms of each benefit in the UDSP framework. The detailed explanations of the evaluation of

Table X. Comparative Evaluation of Schemes in Terms of Deployability Benefits

	Deployability benefits					
Scheme	*Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*
Text-based Passwords	●	●	●	●	●	●
PCCP		●		●		●
Fingerprint	○				○	
RSA SecurID				●	●	
Draw Pattern*		●	●	●	●	
4-digit PINs*	●	●	●	●	●	●
PGA*		●		●	●	

*Represents new benefits in UDSP framework or schemes that were not evaluated earlier.

Table XI. Comparative Evaluation of Schemes in Terms of Security Benefits

	Security benefits											
Scheme	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*	*Resilient-to-Human-Choices-Guided-Guessing**
Text-based Passwords		○						●	●	●	●	
PCCP		●	○			●	●	●	●	●	●	○
Fingerprint	●		●						●	●		●
RSA SecurID	●	●	●	●	●	●	●	●		●	●	○
Draw Pattern*		●						●	●	●	●	
4-digit PINs*								●	●	●	●	
PGA*		○	○				●	●	●	●		

*Represents new benefits in UDSP framework or schemes that were not evaluated earlier.

Table XII. Comparative Evaluation of Schemes in Terms of Privacy Benefits

	Privacy benefits				
Scheme	*Resilient-to-Identity-Disclosure**	*Resilient-to-Image-Disclosure**	*Resilient-to-Interests-Disclosure**	*Resilient-to-Birthday-Disclosure**	*Resilient-to-Physiological-Data-Disclosure**
Text-based Passwords	○	●		●	●
PCCP	●	●		●	●
Fingerprint		●	●	●	
RSA SecurID	○	●	●		●
Draw Pattern*	●	●	●	●	●
4-digit PINs*	●	●	●		●
PGA*				●	●

*Represents new benefits in UDSP framework or schemes that were not evaluated earlier.

those three new schemes—namely, Draw Pattern, PIN, and PGA—in these tables will be discussed in Sections 8.2 and 8.3.

U9. Assistance-to-Recall: Users of the scheme receive implicit or explicit assistance to recall their passwords or credentials at the time of authentication. A scheme is granted *Assistance-to-Recall* if it is *Memorywise-Effortless* or it provides visual, acoustical, or any other built-in-scheme means to help users recall their credentials. Note that *Assistance-to-Recall* is different from *Quasi-Memorywise-Effortless* in UDS, which requires users only to remember one secret for everything.

Text-based passwords are not *Assistance-to-Recall*. Even though text-based password system are often accompanied by password reminders, such reminders are not an inherent part of the authentication scheme. PCCP is *Assistance-to-Recall* because it leverages the human ability to remember images to help users recall their passwords. We consider Fingerprint as *Assistance-to-Recall* because it is *Memorywise-Effortless*. RSA SecurID is not *Assistance-to-Recall*. Even though the users of RSA SecurID can read dynamic 6-digit code from hardware tokens, they still need to memorize and recall their 4-digit PINs without help.

U10. Touchscreen-Friendly: Using the scheme on touch-screen devices is at least as easy as inputting text-based password with keyboards. A scheme is not granted *Touchscreen-Friendly* if its credential input process requires special devices other than touch-screens.

Text-based passwords are not *Touchscreen-Friendly* since users find typing alphanumerics and symbols on touch-screens very difficult. PCCP is *Touchscreen-Friendly* because it only requires users to click points on pictures that could occupy the whole touch-screen. Fingerprint is not *Touchscreen-Friendly* because it requires a fingerprint scanner. We grant RSA SecurID *Touchscreen-Friendly* status since typing only numerical digits is not difficult on touch-screens with virtual numpads.

U11. Enjoyable-to-Use: Users of the scheme authenticate themselves to verifiers in a fluid manner. In the meantime, the user experience of credential input process is enjoyable, which means users consider the scheme fun or exciting to use.

The metric of *Enjoyable-to-Use* is relatively subjective. A user's perception of a scheme may change over time, location, and other factors. For example, a user who is always curious about the new things may enjoy an authentication scheme at first and get bored after using the scheme for a while. Even though it is difficult to accurately model a scheme's enjoyableness, it is necessary to consider this feature as a usability property. To measure this subjective feature, we conducted an informal pilot study to ask our participants' opinions on this property. Our finding on this property is preliminary and only tries to capture users' coarse feelings. Further analysis on this property remains for future work. Twelve graduate and undergraduate students participated in this survey. Each participant was asked to give a binary answer. If a scheme received six positive votes for *Enjoyable-to-Use*, we grant *Enjoyable-to-Use* to the scheme. Based on this survey, none of the text-based passwords, PCCP, Fingerprint, and RSA SecurID is granted *Enjoyable-to-Use*.

S12. Resilient-to-Human-Choices-Guided-Guessing: Users of the scheme do not need to choose secrets as or as part of their credentials. It is statistically proved that human-chosen secrets follow a skewed probability distribution [Bonneau 2012a; Bonneau et al. 2012c]. Many methods have been proposed to use these patterns to guess human-chosen secrets [Thorpe and Van Oorschot 2007; Bonneau 2012b; Uellenbeck et al. 2013]. If human-chosen secrets are only used as part of the credentials in a scheme, or if a scheme has mechanisms to flatten user-chosen patterns, we grant this scheme *Quasi-Resilient-to-Human-Choices-Guided-Guessing*.

Text-based passwords are not granted this benefit. Even if complicated password composition policies are used in real-world websites, new password composition patterns are continuously discovered [Rao et al. 2013]. PCCP and RSA SecurID are granted *Quasi-Resilient-to-Human-Choices-Guided-Guessing* because PCCP requires users to click in a randomly selected portion of each image, which flattens the password distribution, and RSA SecurID has a 6-digit dynamic passcode that is not chosen by humans. We consider Fingerprint to have this benefit.

P1. Resilient-to-Identity-Disclosure: Leak of the credentials of the scheme does not disclose the identities of the credentials' owners. In some cases, the credentials have two parts; namely, identifier (username) and secret (password), such as those in web authentication schemes. In other cases, the identifier part is omitted. An authentication scheme of a mobile operating system that supports only a single user does not need a username. Example schemes include Draw Pattern on Android and PIN on iOS. A scheme is granted this benefit if and only if the disclosure of any part of the credential cannot reveal the identity of the user. A scheme is given *Quasi-Resilient-to-Identity-Disclosure* if the disclosure of the secret part would not give away the user's identity.

Text-based passwords and RSA SecurID are granted *Quasi-Resilient-to-Identity-Disclosure* since user-chosen usernames could easily reveal users' identities. PCCP is granted this benefit because it does not require an overt username. Fingerprint is not granted this benefit since the disclosure of a fingerprint, which is important identifiable information, may lead to identity theft.

P2. Resilient-to-Image-Disclosure: Leak of the credentials of the scheme does not disclose the images of the credentials' owners.

Text-based passwords, RSA SecurID, and Fingerprint are granted this benefit since no image is involved in these schemes. PCCP is also granted this benefit because it only uses preselected images.

P3. Resilient-to-Interests-Disclosure: Leak of the credentials of the scheme does not disclose users interests.

Text-based passwords are not granted this benefit because previous research effort discovered that password compositions are related to users' interests [Castelluccia et al. 2012]. Disclosure of a password may reveal the interests of the corresponding username. PCCP is not granted this benefit because users may be more likely to click objects that draw their interest to a PCCP image. RSA SecurID and Fingerprint are granted this benefit.

P4. Resilient-to-Birthday-Disclosure: Leak of the credentials of the scheme does not disclose users' birthdays.

Fingerprint and PCCP are granted this benefit. RSA SecurID is not because it is statistically significant that users' PINs are related to their birthdays [Bonneau et al. 2012d], and RSA SecurID uses a 4-digit PIN as part of the secret. Text-based passwords are granted this benefit.

P5. Resilient-to-Physiological-Data-Disclosure: Leak of the credentials of the scheme does not disclose users' physiological information.

Fingerprint authentication may disclose users' fingerprint information. As an example, the iPhone 5s introduced fingerprint authentication, which triggered discussions about privacy issues in addition to identity theft. Some are worried that Apple may collect every iPhone 5s user's fingerprint, and losing an iPhone 5s may bring the disclosure of one's fingerprint information. Text-based passwords, PCCP, and RSA SecurID are granted this benefits.

8.2. Evaluation of Draw Pattern and PINs

Draw Pattern and PINs are two commercially popular authentication schemes that are used in Android and iOS, respectively. In this section, we evaluate both within the UDSP framework.

8.2.1. Draw Pattern. Please refer to Uellenbeck et al. [2013], if the reader is not familiar with Android Draw Pattern. Draw Pattern is neither *Memorywise-Effortless* nor *Scalable-for-Users*. It is *Nothing-to-Carry*, but not *Physically-Effortless*. However, it offers advantages over text-based passwords because users can use one hand and one finger to unlock the devices. It is *Easy-to-Learn* and *Efficient-to-Use*. It is *Quasi-Infrequent-Errors* due to inconsistent finger movements. It is *Easy-Recovery-from-Loss* and *Touchscreen-Friendly*. It is not *Assistance-to-Recall*. It is *Enjoyable-to-Use* based on our survey.

Draw Pattern is not *Accessible* for blind users and has *Negligible-Cost-per-User*. It is *Server-Compatible* because each dot could be mapped to an alphanumeric, after which a draw pattern could be stored and compared as a text-based password. It is also *Browser-Compatible*. It is *Mature* and used in hundreds of millions of Android devices. It is not *Non-Proprietary*.

Draw Pattern is not *Resilient-to-Physical-Observation* since either shoulder surfing or smudge attack can reveal the secret. It is *Resilient-to-Targeted-Impersonation* since knowing the user's personal details does not help attack the password. It is neither *Resilient-to-Throttled-Guessing* nor *Resilient-to-Unthrottled-Guessing* due to a small password space and the strong patterns shown in a collected password set [Uellenbeck et al. 2013]. It is neither *Resilient-to-Internal-Observation* nor *Resilient-to-Leaks-from-Other-Verifiers* due to password reuse across sites and devices. It is not *Resilient-to-Phishing* but is *Resilient-to-Theft* since no physical token is used. It has *No-Trusted-Third-Party* and *Requiring-Explicit-Consent*. It is *Unlinkable*. It is not *Resilient-to-Human-Choices-Guided-Guessing* since feasible attacks have been presented [Uellenbeck et al. 2013].

This scheme is *Quasi-Resilient-to-Identity-Disclosure* if a username is required. In some contexts of use, such as in Android, Draw Pattern does not need a username; hence it is *Resilient-to-Identity-Disclosure*. It is both *Resilient-to-Image-Disclosure* and *Resilient-to-Interests-Disclosure*. It is both *Resilient-to-Birthday-Disclosure* and *Resilient-to-Physiological-Data-Disclosure* as well.

8.2.2. Personal Identification Numbers. Four-digit PINs are widely used in ATMs and Apple iPhones. An empirical analysis of customer-chosen banking PINs was presented in Bonneau et al. [2012c]. PINs are neither *Memorywise-Effortless* nor *Scalable-for-Users*. However, they are easier to remember and recall than text-based passwords due to their short representations. They are *Nothing-to-Carry* but not *Physically-Effortless*. They are *Easy-to-Learn* and *Efficient-to-Use*. They are *Infrequent-Errors* and *Easy-Recovery-from-Loss*. They are also *Touchscreen-Friendly* because normally a numpad is displayed to assist user input. They are not *Assistance-to-Recall*. They are not granted *Enjoyable-to-Use* based on our survey.

PINs are a simplified version of text-based passwords. Therefore, PINs provide the same deployabilities as text-based passwords.

PINs are not *Resilient-to-Physical-Observation* because shoulder surfing and video recording are typical ways to steal PINs. They are not *Resilient-to-Targeted-Impersonation* since a considerable portion of users use their birthdays as their PINs. They are neither *Resilient-to-Throttled-Guessing* nor *Resilient-to-Unthrottled-Guessing* due to a small password space and strong patterns [Bonneau et al. 2012c]. They are neither *Resilient-to-Internal-Observation* nor *Resilient-to-Leaks-from-Other-Verifiers* due to PIN reuse. They are not *Resilient-to-Phishing* but *Resilient-to-Theft*. They do not use *No-Trusted-Third-Party*, and they are *Requiring-Explicit-Consent*. They are *Unlinkable*. PINs are not *Resilient-to-Human-Choices-Guided-Guessing*.

PINs are *Quasi-Resilient-to-Identity-Disclosure* if usernames are provided. In some contexts of use, such as in iPhone where usernames are not used, PINs are *Resilient-to-Identity-Disclosure*. They are *Resilient-to-Image-Disclosure*, *Resilient-to-Interests-Disclosure,* and *Resilient-to-Physiological-Data-Disclosure*. They are not *Resilient-to-Birthday-Disclosure*.

8.3. Evaluation of PGA

We presented an empirical analysis of human-chosen PGA passwords in Section 3 and of nontargeted and targeted attack results on collected passwords in Section 6 and Section 7, respectively. The results from both empirical analysis and attacks serve as the basis for us to evaluate if PGA provides certain criterion.

PGA is not *Memorywise-Effortless* since the size, location, and ordering of each drawing must be remembered. It is not *Scalable-for-Users* because a password has to be chosen for each site per user. However, it is easier to recall than text-based passwords due to the human ability to remember images. We capture this characteristic by giving it *Assistance-to-Recall*. It is obviously *Nothing-to-Carry*. It is not *Physically-Effortless* because users need to draw gestures on screens. However, it offers advantages over text-based passwords because, in most cases, users could use one hand and one finger to unlock the device quickly. It is *Touchscreen-Friendly* because drawing gestures on touch-screens is much easier than typing alphanumerics. It is *Easy-to-Learn* and *Efficient-to-Use* according to our user studies. It is *Quasi-Infrequent-Errors* and *Easy-Recovery-from-Loss*. It is *Enjoyable-to-Use* based on our survey.

PGA is not *Accessible* for blind users. It has *Negligible-Cost-per-User* because no costly physical token is involved. It is not *Server-Compatible* with text-based passwords because the format and comparison of passwords are not the same as those of text-based passwords. It is *Browser-Compatible* because developers could capture drawings using JavaScript. It is *Mature* and used in hundreds of millions of Windows devices. It is not *Non-Proprietary* because a patent [Johnson et al. 2012] was issued to Microsoft$^{\text{TM}}$.

PGA is not *Resilient-to-Physical-Observation*. In fact, shoulder-surfing PGAs on a large screen would be a serious problem [Honan 2012]. It is at best *Quasi-Resilient-to-Targeted-Impersonation* since knowing the user's personal habits and inclinations could help in guessing her drawings. It is at best *Quasi-Resilient-to-Throttled-Guessing* and not *Resilient-to-Unthrottled-Guessing* based on our attack results in Section 6. It is neither *Resilient-to-Internal-Observation* nor *Resilient-to-Leaks-from-Other-Verifiers* due to password reuse across sites and devices. It is *Resilient-to-Phishing* because users choose their own images as background. It is *Resilient-to-Theft* since no physical token is used. It has *No-Trusted-Third-Party* and *Requiring-Explicit-Consent*. It is not *Unlinkable* because a user may use identical but personal images on different sites and devices. It is not *Resilient-to-Human-Choices-Guided-Guessing*, as we discussed in Sections 3 and 6.

PGA is not *Resilient-to-Identity-Disclosure* since the background image discloses the identity of the user. It is neither *Resilient-to-Image-Disclosure* nor *Resilient-to-Interests-Disclosure,* which we discuss in Section 3. It is both *Resilient-to-Birthday-Disclosure* and *Resilient-to-Physiological-Data-Disclosure*.

In summary, PGA provides very good usability, especially on touch-screen devices. As shown in Table IX, PGA offers more usability benefits than all other discussed schemes. In particular, it is one of only two schemes classified as *Enjoyable-to-Use* and one of only three schemes that offer *Assistance-to-Recall*. The good user experience provided by PGA might be a major reason for it to be included in a commercially popular operating system. For deployability, only PGA offers more benefits than Fingerprint and RSA SecurID, both of which require the addition of some physical devices on either the authenticator or authenticatee side. The proprietary protection for PGA may prevent it from being deployed on more devices or websites. On the security side, the size of the password space is very important. As listed in Table I, PGA offers larger password space than the other two popular touch-screen-friendly schemes. However, its space size with current settings is smaller than that for text-based passwords generated with strict composition rules. When it comes to the number of security benefits in UDSP, PGA offers the same number of security benefits as Draw Pattern, PINs, and text-based passwords, all of which are considered not very secure. Even though the security of a scheme cannot be accurately measured by the number of security benefits it offers, this number can tell us how many attack methods a scheme could be resilient to. Obviously, the security of PGA cannot compete with some more secure schemes, such as RSA SecurID. From the perspective of privacy, PGA offers the least number of privacy benefits in all discussed schemes, as shown in Table XII. The introduction of users' own pictures makes PGA enjoyable to use at the expense of exposing their images. Therefore, users may hesitate using PGA-like schemes on devices that do not belong to them. The good usability of PGA comes with a price not only on its security features, but also on its privacy protection.

9. DISCUSSION

9.1. Other Attacks on PGA

In addition to keyloggers that record users' finger movements, other attack methods may affect the security of PGA and other BDAS schemes. Shoulder-surfing, in which 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 are text-based password. Several new shoulder-surfing-resistant schemes [Forget et al. 2010; Zakaria et al. 2011] were proposed recently. However, usability is always a major concern for these approaches. The smudge attack [Aviv et al. 2010], which recovers passwords from the oily residues on a touch-screen, has also been proved feasible to BDAS schemes and could pose threats to PGA.

9.2. Limitations of Our Study

Although 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 normaly do on Windows 8^{TM} PGA. As previously mentioned, subjects in *Dataset-1* used their passwords for access more than four times a week on average. However, the data protected by the passwords in the website was not sensitive. In this case, subjects preferred passwords with simpler gesture-type combinations that were easier to reproduce quickly. There is a lack of recall session for *Dataset-2*, since subjects pretended to access their bank information but did not have anything at risk. There is a lack of recall session for *Dataset-2* since subjects chose passwords relatively quickly and never came back to use the chosen passwords. Schechter et al. [2007] 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. In addition, we did not record whether a subject used a tablet with touch-screen or a desktop with mouse. The different input methods may affect the composition of passwords. Moreover, *Dataset-2* includes multiple passwords per user, and this may have impacted the results.

10. RELATED WORK

Biddle et al. provided an excellent survey on the first 12 years of the history of graphical passwords in Biddle et al. [2011]. Most graphical password schemes proposed in academia and those used in real-world products from 1999 to 2010 have been covered in that paper. In all existing schemes, PGA is most similar to BDAS [Dunphy and Yan 2007]. De Angeli et al. [2005] stated that the weakness of all knowledge-based authentication systems reflects a tradeoff between security and human memory constraints. Based on the memory tasks involved in remembering and inputting passwords, these two papers divided all extant graphical password schemes into three categories. In recall-based systems (aka drawmetric systems), no password cue is given. Exemplary schemes include DAS [Jermyn et al. 1999], YAGP [Gao et al. 2008], Passdoodle [Varenhorst et al. 2004], PassShapes [Weiss and De Luca 2008], and Pass-Go [Tao and Adams 2008]. In recognition-based systems (aka cognometric systems or searchmetric systems [Renaud 2009]), users need to memorize a portfolio of images during password creation and recognize them to log in. Exemplary schemes include Face [Brostoff and Sasse 2000], Story [Davis et al. 2004], and D*éjà* Vu [Dhamija and Perrig 2000]. And, in cued-recall systems (aka locimetric systems), users remember specific locations within an image. Exemplary schemes include PassPoints [Wiedenbeck et al. 2005b], Cued Click-Points [Chiasson et al. 2007], and Persuasive Cued Click-Points [Chiasson et al. 2012]. PGA is more like a cued-recall system because users need to recall the locations of gestures, and the chosen background images could serve as cues for users. However, PGA also has some features that were only previously observed in recall-based and recognition-based systems. For example, in contrast to previous cued-recall systems in which users only remember locations, PGA also requires users to recall their gesture types associated with locations. The task to recall gesture types in PGA is similar to the tasks in recall-based systems. According to our study, user-chosen gesture types are highly related to the attributes of user-chosen gesture locations. Therefore, background images provide users with cues for both gesture locations and gesture types. In addition, to use PGA, users need to recognize certain objects in background pictures and even recognize the full pictures to counter phishing attacks. This recognition task gives PGA some features of recognition-based systems. In a sense, PGA blurs the boundaries of the three aforementioned graphical password categories.

The basic idea of attacking graphical password schemes is to generate dictionaries that consist of potential passwords [Thorpe and van Oorschot 2004]. However, the lack of sophisticated mechanisms for dictionary construction affects the attack capabilities of existing approaches. Davis et al. [2004] used the relations of users' demographic information and their passwords to guess Face passwords. Thorpe and Van Oorschot [2007] 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 [Thorpe and Van Oorschot 2007], they presented another approach that creates dictionaries by predicting hotspots using image-processing methods. van Oorschot and Thorpe [2008] cracked DAS using password complexity factors, such as reflective symmetry and stroke-count. Dirik et al. [2007] modeled user choice in PassPoints and proposed automated dictionary attacks. Salehi-Abari et al. [2008] 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. To attack PGA passwords, there are at least three requirements an attack approach should meet: (i) the approach should work on complex gestures as well as on simple click, (ii) the approach should learn patterns from harvested passwords even if they are not collected from the target picture, and (iii) the approach should generate dictionaries for previously unseen pictures. Our attack framework differs from previous efforts by introducing the idea of a selection function that abstracts and models users' password creation processes and therefore meets all three requirements.

The security and vulnerability of text-based password have attracted considerable attention because of several infamous password leakage incidents in recent years. The approaches for guessing text-based passwords greatly influenced the design of our attack framework. Zhang et al. [2010] studied password choices over time and proposed an approach to attack new passwords from old ones. Castelluccia et al. [2012] proposed an adaptive Markov-based password strength meter by estimating the probability of passwords using training data. Kelley et al. [2012] 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 PGA passwords, the idea of abstracting users' selection processes of password construction introduced in this article could also be applicable to cracking and measuring text-based passwords. Veras et al. [2014] presented such an approach to extract and understand semantic patterns in text passwords.

11. CONCLUSION

We described an empirical analysis of Windows 8^{TM} PGA passwords collected from our online user studies. The empirical analysis has helped us understand user choice patterns in background picture, gesture location, gesture order, and gesture type. We presented a novel attack framework that makes use of the extracted user choice patterns to guess Windows 8^{TM} PGA passwords. Using the proposed attack framework, we demonstrated that our approach was able to crack a considerable portion of picture passwords in various situations. Based on the empirical analysis and attack results, we comparatively evaluated PGA with a set of extended benefits from four categories;

namely, usability, deployability, security, and privacy. We also discovered that PGA provides more usability benefits than all other schemes we analyzed. However, it has more limitations in deployability and security benefits than most of the schemes we analyzed. Moreover, it brought up some privacy concerns, such as disclosing users' images. We believe the findings discussed in this article could advance the understanding of PGA and its advantages and limitations.

REFERENCES

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. 2012. Measuring the objectness of image windows. *IEEE Transactions Pattern Analysis and Machine Intelligence* (2012), 2189–2202.

Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. 2010. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*. USENIX Association, 1–7.

Dana H. Ballard. 1981. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition* 13, 2 (1981), 111–122.

Kemal Bicakci, Nart Bedin Atalay, Mustafa Yuceel, Hakan Gurbaslar, and Burak Erdeniz. 2009. Towards usable solutions to graphical password hotspot problem. In *Proceedings of the 33rd IEEE International Conference on Computer Software and Applications Conference*, Vol. 2. IEEE, 318–323.

Robert Biddle, Sonia Chiasson, and Paul C. Van Oorschot. 2011. Graphical passwords: Learning from the first twelve years. *Computer Surveys* 44, 4 (2011).

Joseph Bonneau. 2012a. Guessing human-chosen secrets. University of Cambridge.

Joseph Bonneau. 2012b. The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 538–552.

Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. 2012a. *The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes*. Technical Report UCAM-CL-TR-817. University of Cambridge, Computer Laboratory.

Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. 2012b. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 553–567.

Joseph Bonneau, Sören Preibusch, and Ross Anderson. 2012c. A birthday present every eleven wallets? The security of customer-chosen banking PINs. In *Proceedings of the the 16th International Conference on Financial Cryptography*.

Joseph Bonneau, Sören Preibusch, and Ross Anderson. 2012d. A birthday present every eleven wallets? The security of customer-chosen banking PINs. *Financial Cryptography and Data Security* (2012), 25–40.

Ali Borji, Dicky N. Sihite, and Laurent Itti. 2012. Salient object detection: A benchmark. In *Proceedings of the 2012 European Conference on Computer Vision*. Springer, 414–429.

Ali Borji, Hamed R .Tavakoli, Dicky N. Sihite, and Laurent Itti. 2013. Analysis of scores, datasets, and models in visual saliency prediction. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*. IEEE, 921–928.

Sacha Brostoff and M. Angela Sasse. 2000. Are Passfaces more usable than passwords? A field trial investigation. *People and Computers* (2000), 405–424.

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

Claude Castelluccia, Markus Dürmuth, and Daniele Perito. 2012. Adaptive password-strength meters from Markov models. In *Proceedings of the 19th Network and Distributed System Security Symposium*.

Sonia Chiasson, Alain Forget, Robert Biddle, and Paul C. van Oorschot. 2009. User interface design affects security: Patterns in click-based graphical passwords. *International Journal of Information Security* 8, 6 (2009), 387–398.

Sonia Chiasson, Elizabeth Stobert, Alain Forget, Robert Biddle, and Paul C. Van Oorschot. 2012. 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.

Sonia Chiasson, Paul van Oorschot, and Robert Biddle. 2007. Graphical password authentication using cued click points. In *Proceedings of the 12th European Symposium on Research in Computer Security*. Springer, 359–374.

Darren Davis, Fabian Monrose, and Michael K. Reiter. 2004. On user choice in graphical password schemes. In *Proceedings of the 13th Conference on USENIX Security Symposium*. USENIX Association, 11–23.

Antonella De Angeli, Lynne Coventry, Graham Johnson, and Karen Renaud. 2005. Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies* 63, 1 (2005), 128–152.

Rachna Dhamija and Adrian Perrig. 2000. D*éjà* Vu: A user study using images for authentication. In *Proceedings of the 9th Conference on USENIX Security Symposium*. USENIX Association.

Ahmet Emir Dirik, Nasir Memon, and Jean-Camille Birget. 2007. Modeling user choice in the PassPoints graphical password scheme. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*. ACM, 20–28.

Paul Dunphy and Jeff Yan. 2007. Do background images improve draw a secret graphical passwords? In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. ACM, 36–47.

Uriel Feige, László Lovász, and Prasad Tetali. 2004. Approximating min sum set cover. *Algorithmica* 40, 4 (2004), 219–234.

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9 (2010), 1627–1645.

Alain Forget, Sonia Chiasson, and Robert Biddle. 2010. 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*. ACM, 1107–1110.

Haichang Gao, Xuewu Guo, Xiaoping Chen, Liming Wang, and Xiyang Liu. 2008. Yagp: Yet another graphical password strategy. In *Proceedings of the 24th Annual Computer Security Applications Conference*. IEEE, 121–129.

Ross B. Girshick, Pedro F. Felzenszwalb, and David McAllester. 2010. Discriminatively Trained Deformable Part Models, Release 5. Retrieved from http://people.cs.uchicago.edu/ rbg/latent-release5/.

Brian Honan. 2012. Visual Data Security White Paper. Retrieved from http://www.visualdatasecurity.eu/wp-content/uploads/2012/07/Visual-Data-S ecurity-White-Paper.pdf.

Dawei Hong, Jean-Camille Birget, and Nasir Memon. 2006. Graphical passwords based on robust discretization. *IEEE Transactions on Information Forensics and Security* 1, 3 (2006), 395–399.

Ian Jermyn, Alain Mayer, Fabian Monrose, Michael K. Reiter, and Aviel D. Rubin. 1999. The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium*. USENIX Association, 1–14.

Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 2013. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2083–2090.

Jeff Johnson, Steve Seixeiro, Zachary Pace, Giles Van der Bogert, Sean Gilmour, Levi Siebens, and Ken Tubbs. US Patent 163201, 2012. Picture gesture authentication. (US Patent 163201, 2012).

Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 523–537.

Microsoft. 2013. Microsoft by the Numbers. Retrieved from http://www.microsoft.com/en-us/news/bythenumbers/ms_numbers.pdf.

Zach Pace. 2011a. Signing in with a Picture Password. Retrieved from http://blogs.msdn.com/b/b8/archive/2011/12/16/signing-in-with-a-picture-password.aspx.

Zach Pace. 2011b. Signing into Windows 8 with a Picture Password. Retrieved from http://www.youtube.com/watch?v=Ek9N2tQzHOA.

Ashwini Rao, Birendra Jha, and Gananand Kini. 2013. Effect of grammar on security of long passwords. In *Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*. ACM, 317–324.

Karen Renaud. 2009. Guidelines for designing graphical authentication mechanism interfaces. *International Journal of Information and Computer Security* 3, 1 (2009), 60–85.

Amirali Salehi-Abari, Julie Thorpe, and Paul C. van Oorschot. 2008. On purely automated attacks and click-based graphical passwords. In *Proceedings of the 24th Annual Computer Security Applications Conference*. IEEE, 111–120.

Stuart Schechter, Cormac Herley, and Michael Mitzenmacher. 2010. Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks. In *Proceedings of the 5th USENIX conference on Hot Topics in Security*. USENIX Association, 1–8.

Stuart E. Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. 2007. The emperor's new security indicators. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*. IEEE, 51–65.

Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. 2005. Graphical passwords: A survey. In *Proceedings of the 21st Annual Computer Security Applications Conference*. IEEE, 10–19.

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

Hai Tao and Carlisle Adams. 2008. Pass-Go: A proposal to improve the usability of graphical passwords. *International Journal of Network Security* 7, 2 (2008), 273–292.

Julie Thorpe, Muath Al-Badawi, Brent MacRae, and Amirali Salehi-Abari. 2014. The presentation effect on graphical passwords. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2947–2950.

Julie Thorpe and Paul Van Oorschot. 2004. Towards secure design choices for implementing graphical passwords. In *Proceedings of the 20th Annual Computer Security Applications Conference*. IEEE, 50–60.

Julie Thorpe and Paul Van Oorschot. 2007. Human-seeded attacks and exploiting hot-spots in graphical passwords. In *Proceedings of 16th USENIX Security Symposium*. USENIX Association, 8.

Julie Thorpe and Paul C. van Oorschot. 2004. Graphical dictionaries and the memorable space of graphical passwords. In *Proceedings of the 13th Conference on USENIX Security Symposium*. USENIX Association, 135–150.

Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the security of graphical passwords: The case of Android unlock patterns. In *Proceedings of the 20th ACM Conference on Computer and Communications Security*. ACM, 161–172.

Paul C. Van Oorschot, Amirali Salehi-Abari, and Julie Thorpe. 2010. Purely automated attacks on PassPoints-style graphical passwords. *IEEE Transactions on Information Forensics and Security* 5, 3 (2010), 393–405.

Paul C. van Oorschot and Julie Thorpe. 2008. On predictive models and user-drawn graphical passwords. *ACM Transactions on Information and System Security* 10, 4 (2008), 5.

Paul C. van Oorschot and Julie Thorpe. 2011. Exploiting predictability in click-based graphical passwords. *Journal of Computer Security* 19, 4 (2011), 669–702.

Christopher Varenhorst, M. V. Kleek, and Larry Rudolph. 2004. Passdoodles: A lightweight authentication method. *MIT Research Science Institute* (2004).

Rafael Veras, Christopher Collins, and Julie Thorpe. 2014. On the semantic patterns of passwords and their security impact. In *Proceedings of the Network and Distributed System Security Symposium*.

Paul Viola and Michael J. Jones. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57, 2 (2004), 137–154.

Roman Weiss and Alexander De Luca. 2008. PassShapes: Utilizing stroke based authentication to increase password memorability. In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*. ACM, 383–392.

Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. 2005a. Authentication using graphical passwords: Effects of tolerance and image choice. In *Proceedings of the Symposium on Usable Privacy and Security*. ACM, 1–12.

Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. 2005b. PassPoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies* 63, 1 (2005), 102–127.

Qiang Yan, Jin Han, Yingjiu Li, and Robert H. Deng. 2012. On limitations of designing leakage-resilient password systems: Attacks, principles and usability. In *Proceedings of the 19th Network and Distributed System Security Symposium*.

John C. Yuille. 1983. *Imagery, Memory, and Cognition*. Lawrence Erlbaum Associates, Inc.

Nur Haryani Zakaria, David Griffiths, Sacha Brostoff, and Jeff Yan. 2011. Shoulder surfing defence for recall-based graphical passwords. In *Proceedings of the 7th Symposium on Usable Privacy and Security*. ACM, 6–17.

Yinqian Zhang, Fabian Monrose, and Michael K. Reiter. 2010. The security of modern password expiration: An algorithmic framework and empirical analysis. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*. ACM, 176–186.

Ziming Zhao, Gail-Joon Ahn, Jeongjin Seo, and Hongxin Hu. 2013. On the security of picture gesture authentication. In *Proceedings of the 22nd USENIX Security Symposium*. USENIX Association, 383–398.

Received February 2014; revised November 2014; accepted December 2014