A Closer Look at Recognition-based Graphical Passwords on Mobile Devices

Paul Dunphy
Newcastle University
Newcastle upon-Tyne, UK
p.m.dunphy@ncl.ac.uk

Andreas P. Heiner
Nokia Research Center
Helsinki, Finland
andreas.heiner@nokia.com

N. Asokan
Nokia Research Center
Helsinki, Finland
n.asokan@nokia.com

ABSTRACT

Graphical password systems based on the recognition of photographs are candidates to alleviate current over-reliance on alphanumeric passwords and PINs. However, despite being based on a simple concept – and user evaluations consistently reporting impressive memory retention – only one commercial example exists and overall take-up is low. Barriers to uptake include a perceived vulnerability to observation attacks; issues regarding deployability; and the impact of innocuous design decisions on security not being formalized. Our contribution is to dissect each of these issues in the context of mobile devices – a particularly suitable application domain due to their increasing significance, and high potential to attract unauthorized access. This produces: 1) A novel yet simple solution to the intersection attack that permits greater variability in login challenges; 2) Detailed analysis of the shoulder surfing threat that considers both simulated and human testing; 3) A first look at image processing techniques to contribute towards automated photograph filtering. We operationalize our observations and gather data in a field context where decentralized mechanisms of varying entropy were installed on the personal devices of participants. Across two working weeks success rates collected from users of a high entropy version were similar to those of a low entropy version at 77%, and login durations decreased significantly across the study.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Access Controls, Authentication*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Human Factors,Security

Keywords

Graphical Passwords, Shoulder Surfing, Mobile Devices

Symposium on Usable Privacy and Security (SOUPS) 2010, July 14–16, 2010, Redmond, WA USA
.

1. INTRODUCTION

User authentication on mobile devices is an area of increasing significance as their technical specifications improve, with modern devices being comparable to PCs of the recent past. This along with the emergence of fast 3G data connections and wireless connectivity has contributed to devices becoming as important in everyday life as personal computers, in terms of the services they provide and the data they store. Consequently mobile devices are now used for security sensitive tasks such as authenticating to on-line services, authorizing commercial transactions, [3, 18] and accessing corporate e-mail.

In these cases where the authenticity of users is of high importance the device itself should implement an authentication mechanism to defend against opportunistic access by attackers to data and services. Official UK Home Office statistics report that 700,000 devices were stolen throughout the UK in 2001 with unofficial estimates as high as 1.3million [16]. Moreover, statistics from 2005 report that 69% of mobile device thefts occurred when the device was left unattended [14]. In the face of such figures, users attributing privacy solely to their assumed physical possession of the device are taking a risk; as the possessor of an unprotected device assumes the full capabilities offered by the on-board credentials. This access can even occur with the consent of the user, as it is common for users to briefly share mobile devices – particularly with friends and family – for making or receiving calls [20].

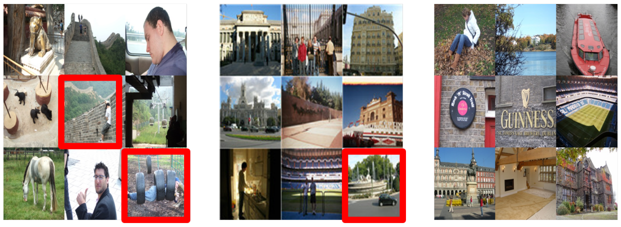

Figure 1: Typical login procedure for recognition-based systems, the user would see each grid of images in turn, and must identify any personal password images that appear. Images highlighted represent example selections.

Currently mobile devices support a binary security model where the user is either legitimate or not. PINs are the *de facto* authentication mechanism used in this model due to their ease of implementation, cost and accessibility to multiple platforms. However it is very common for users to forget a sequence of numbers, even when used frequently [29]. One survey [6] reports that only one-third of mobile device users actually enforce PIN protection, 30% find PINs to be inconvenient, but 85% are in favour of more device security.

Graphical Passwords on Mobile Devices?: One option to reinforce device security is user authentication based on *graphical passwords* [33]. These are knowledge-based authentication mechanisms conceptually underpinned by *dual-coding theory* that postulates that both visual and verbal information are processed differently in the brain and both are complimentary to the encoding of an object in memory. Shepard [31] shows that test subjects were 97% accurate in identifying previously seen images when presented with 68 pairs, and making a binary response. The goal is to design knowledge-based authentication systems around this effect to help users reliably retain random authentication secrets of higher entropy than is possible with PINs or alphanumeric passwords. Arguably the most intuitive genre is based on *recognition*. In a typical login protocol the user is challenged to recognize pre-assigned *key images* among *decoy images*. Fig. 1 illustrates an example challenge.

Variations of this approach have been proposed many times in previous research [8] [26] [34] [17], however despite being built on an intuitive concept – and their usability being highlighted in each user study – only one commercial example exists [25], and other developments are non-existent. We believe this lack of usage can be attributed to perceived shortcomings in security and deployability. Security concerns include a perceived vulnerability to *shoulder surfing*, where an attacker uses simple observation techniques to learn authentication credentials and later use them in a *replay attack*; and *intersection attack* where the frequency of images appearing at login can be used to determine their importance.

A more pragmatic limitation exists in terms of deployability. For example, to bootstrap systems with images; administrators must source images to use and filter them to reduce the potential for logins comprising confusing visual searches that cause false-negative login results. Many previous works place this problem out of scope.

Recognition-based systems have the potential to form deployable, decentralized, solutions of varying entropy on mobile devices for a number of reasons: 1) They require only simple interaction that is suited to T9 keypads or touchscreens; 2) Device screen resolutions are improving quickly (Nokia N900 has a 800x480 display); 3) The perceived vulnerability to observation is intuitively offset by small screen sizes; 4) Devices are often camera-enabled which means many users already have photographs on devices that can be used for initialization.

Contributions: Motivated by the above, our contribution is to investigate pending pertinent aspects of security, deployability and usability:

- **Shoulder Surfing Analysis** We analyze the *key image portfolio* [1] concept proposed to complicate observation attacks, and secure its inherent vulnerability to intersection attack. We also quantify the benefit of this approach by developing a computer simulation of an attack assuming short term memory and camera attacks are the adversary. We also carry out a user study to explore how real users compare.

- **Novel solution to intersection attack** A widely-accepted solution to an intersection attack is to keep each login challenge the same. We present a new simple method that combines the *key image portfolio* with a *decoy image portfolio* that innately permits greater visual variation in login challenges.

- **Decentralized infrastructure using only personal images** Users have increased memorability for personal images [36], and assuming the presence of these on the device of the user (or on other importable mediums), these can be used to initialize systems and is immune to image source-based attacks such as social engineering [26].

- **Automated image filtering techniques** In the case where users initialize a system using their own images, there must exist an initial, automated phase of filtering to remove those that could contribute to false-negative login results. We make a first exploration of image processing techniques to automatically filter *visually similar* and *unmemorable* photographs.

Based on our observations we gather field data from two exemplar systems of varying entropy. We discovered that average successful login durations recorded from both systems significantly decreased after one week of usage, and overall success rates were similar across both systems at 77%, although in one week of reduced usage, users of a low entropy system were significantly more successful than those using a high entropy version.

The rest of the paper is organized as follows: Section 2 reviews related work, Section 3 considers the various traditional threats associated with systems of this genre and our own analysis and solutions. Section 4 considers how manipulations of the images themselves can have both security and usability implications. This leads us to introduce two implementations in Section 5 and their evaluation in user studies from Section 6. Finally we discuss our findings from Section 8 and conclude in Section 10.

2. RELATED WORK

There are typically three genres of graphical password system: *recall based*, *cued-recall based* and *recognition-based*. Draw a Secret (DAS) [19] is the most prominent recall based system and involves creating a freehand drawing and remembering its composition aided by a drawing grid. This can achieve significantly higher levels of entropy than recognition and cued-recall based systems with a comparable cognitive load. Passpoints [40] is the main exemplar of the cued-recall genre where the user must remember a sequence of clicks positions in an image. Image processing techniques [9] and less sophisticated attacks [35] have been applied to render this system effectively broken.

Table 1 collates recognition-based systems featuring in previous research and highlights key characteristics. Passfaces [25] is the main exemplar and has the only commercial presence. In field studies this has been shown to exhibit increased login success rates over alphanumeric passwords [2]. Also, one study explored how the stimuli of *faces* might affect the ubiquitous practices of writing down and sharing passwords [11]. Dhamija and Perrig [8] evaluated a system based on recognition of fractals and first considered the *intersection attack*. De Angeli et al. [1] evaluated different forms of a system called VIP, and reported that static positioning of images between login challenges aided memorability. They also report that while there does exist a picture superiority effect that can be exploited in authentication, the benefit of this can be lost if a scheme is not designed to harness this effect. In general, this genre is suitable to achieving entropy comparable to a randomly generated 4-digit PIN (13 bits) due to its inherent poor scalability. Entropy is increased by requiring users to remember more key images, and be challenged with more decoy images, which can make the visual search tiresome.

In order to reduce the threat of shoulder surfing more complicated entry is often required. Weinshall [39] presents a method where users do not explicitly select key images, but use them to algorithmically obtain a result that if observed is not useful in a replay attack. However using knowledge of the algorithm used for entry, Golle and Wagner [15] were able to derive key images using a SAT solver.

Table 1: A comparison of prominent recognition-based schemes by various key features. *"'User picks"'* refers to users choosing key images

System	Key src.	Decoy src.	User Picks	Filter-ing	Entropy (bits)
VIP3 [1]	stock	stock	no	manual	12
Awase [34]	user	stock	yes	none	13
UYI [17]	user	stock	yes	none	11
Pering et al. [26]	user	peers	yes	manual	20
Déjà vu [8]	fractals	fractals	yes	manual	16
Passfaces [25]	stock	stock	no	manual	13

Awase [34] and Use your Illusion (UYI) [17] are both designed for mobile devices. In Awase users upload personal images to a server to comprise key images, while decoys are drawn from the images of other participating users, a configuration shared with Pering et al. [26]. User recall was 100% over as long as 16 weeks however their measures to protect against intersection attack are unclear. UYI considered the difficulty of displaying images for authentication on low resolution devices by blurring images in a controlled way. They reported 100% recall over 4 weeks, except in the condition where users were assigned key images when this reduced to 89%.

Pering et al. [26] evaluated a novel system for public terminals, where users would recognize a sequence of *one-time* images from their personal collection. They also explored vulnerability to social engineering attacks where attackers were given a subset of a user images and asked to use the information gained (e.g. recurrence of individuals or themes) to guess the key images. In the worst case attackers were 50% successful. Dynahand [27] is a system where users are asked to recognize their own handwriting. This work also considered the problem that some people have similar handwriting and considered automated methods of filtering using statistical moments and *vector quantisation*.

These schemes can be further organised between two categories, *one-key-per-screen* [25, 26] and *multiple selection* [17, 1, 34, 8]. In the first instance key images are evenly distributed across login screens so that on each screen displayed to the user, one key image exists amongst decoys. In the latter, zero, one, or more than one key image can appear per screen; this is desirable as entropy can be adjusted without increasing the number of grids presented. Another approach altogether is to use image passwords as a memory aid for recall of PINs as seen with FacePIN [12]. This requires no changes to infrastructure and in a small study showed promise.

3. TRADITIONAL THREATS & COUNTER-MEASURES

3.1 Mobile Authentication Threat Model

On mobile devices there are typically two types of PIN code that can be enabled: a *Subscriber Identity Module* (SIM) code that is managed by the Network Provider, and a *device code* that is managed by the user. The SIM code aims to prevent unauthorized access to services on the SIM, and if entered incorrectly x times will prevent any access to the device until a *personal unlock code* (PUK) is obtained from the Network Provider. If the PUK code is entered incorrectly y times the SIM is blocked forever. Repeated incorrect entry of the device code does not usually block the device permanently, however access can usually be restored at the expense of the stored data by a reformat. It is at this level the implementation of a custom authentication mechanism is most feasible causing the least impact to existing infrastructure.

The worst case scenario is of theft, as an attacker has unrestricted physical access to the device. Mobile devices are most commonly stolen for the resale value of the handset rather than targeted access to data. In fact 25% of stolen devices reappear on different networks (an underestimate due to tampering possibilities), however despite this it is still common for thieves to explore the data and services on the device before resale [16]. Theft means a brute force attack is feasible since repeated incorrect attempts on device-level authentication does not normally render the device permanently *locked*. Although in practice this is not a desirable option as the attack must be performed online, and methods to frustrate this exist such as the device locking for a time period. A *lunchtime attack* is where an attacker tries to compromise the device protection over time, through physical access while the legitimate user is absent. Each time the attacker has the intention not to leave a trace to ensure that future attempts will not be hindered, by for example never triggering PIN lockout. There are two classes of lunchtime attacker, firstly a *naive* attacker who can only make random guesses, and secondly a *knowledgeable* attacker who by means of shoulder surfing, or another eavesdropping technique, has gained some knowledge that will assist in the login procedure.

We propose that knowledgeable attackers are most likely to be individuals known to the user, particularly due to the social pressures that can prevent users from practicing security conscious behavior in the presence of colleagues or friends. This could be referred to as a *friend attack*. Insecure behavior has been noted in previous work as a response to such social pressures [29]. Therefore in an image-based authentication context it should be assumed that users will voluntarily permit attackers to view logins, and so systems should be designed to at least overwhelm short term memory (attackers resorting to using a video camera to record login attempts would arouse suspicion).

3.2 Intersection Attack

An *intersection attack* is possible when the frequency of an image appearing at login can be used to determine its role as either a key or a decoy. Perhaps the only practical method proposed in previous research to complicate the shoulder surfing threat is innately vulnerable to this attack [1]. In the VIP system the user is assigned a *key image portfolio*, of which a random subset is exposed at each login. The effect of this is that observing key images at one login might not be useful when logging in at the next. In the described setup each decoy image is certain to appear at login, but due to the variation afforded by the key image portfolio each key image has a 50% chance. In this scenario an attacker can derive the key images without an attack that involves manipulation or observation of the legitimate user. Current wisdom to protect against this attack is not to implement a key image portfolio at all, to ensure every login challenge is the same [17] [8] [25]. One side-effect of this solution is that the login trial is intuitively more vulnerable to observation, as everything the attacker learns is useful in an immediate replay attack.

We propose a simple new approach that preserves the shoulder surfing resistance of the VIP method, and removes its vulnerability to intersection attack. In addition to a key image portfolio we introduce a *decoy image portfolio*. If there will be variation in the key images presented across login challenges, there should be exactly the same variation in the decoy images so that patterns in frequency do not emerge. To achieve this keys and decoys should both be randomly selected from larger, fixed portfolios where the same ratio exists between images selected for login and the size of the portfolio. In other words, if k_l and d_l refer to the number of keys and decoys (respectively) displayed at any given login challenge, and k and d refer to the total number of key and decoy images (respectively) being assigned to a user, then

$$\frac{k_l}{k} = \frac{d_l}{d}$$

To use this formula in practice a system administrator would choose values for k, k_l, d_l, then calculate $d = d_l \frac{k}{k_l}$. Calculation of the *minimum number of images* required to bootstrap a system for the given parameters is given by $d + k$.

3.3 Shoulder Surfing Analysis

Shoulder surfing is an attack where login credentials are observed at the point of login, and reused in a later *replay attack*. It is extremely difficult to quantify the extent of the shoulder surfing threat. Statistics are scarce as instances of shoulder surfing are likely to be supporting techniques to fraud, a crime particularly under-reported to Police. Despite this, there is much anecdotal evidence that suggests this is a problem and is a attack the average user can readily perceive and carry out.

The mobile device context does force a common sense rethink of the threat posed by an impersonal shoulder surfer due to the size of the devices and their portability. However it must still be assumed that each login will be observed, particularly by friends and colleagues, with whom users might neglect to shield their input due to the connotations of mistrust.

Clearly the shoulder surfing resistance offered by the image portfolios does not render systems immune to shoulder surfing, however the goal is to increase the number of observations required for an attacker to learn enough key images to guarantee the success of a future replay attack. Since this appears to be the only practical defense proposed to date, we attempted to quantify the benefit. We created a model of an attack that incorporated the behavior of the key and decoy image portfolios described previously, where $k_1 = 4, d_l = 32$. The value of k was a variable and d adjusted accordingly to preserve the correct ratios. The intention was to explore the capabilities of an attacker of differing memory accuracies, against increasing key and decoy portfolios, where the attacker is attempting to identify key images. The model encompassed the following behavior:

1. A login challenge occurs, the attacker observes legitimate user entry of each key image and given a probability p remembers it.

2. The attacker then has one attempt to authenticate and is presented a new challenge, if the attacker can identify 4 images that he has previously observed being selected, the attack is successful, otherwise unsuccessful.

3. The cycle repeats, with the knowledge of the attacker increasing each time until an attack is successful.

Fig. 2 reports the mean number of observations required for an attacker to learn: 1) Enough key images to login once 2) All key images. Intuitively the number of required observed logins increases with the size of the key image portfolio, and the decreasing accuracy of observer memory. Considering the most likely case where the goal of the attacker is to achieve one login, a camera equipped attacker on average requires less than five observations, and needs more than 10 only when the size of the key portfolio is increased to 14. In practice, increasing the size of the key portfolio to this level is likely to be detrimental to usability. Even when the attacker has only a 50% chance to remember observed images the key portfolio would need to be of size 10 to force more than five observations. Another case may arise where an attacker would like to obtain the entire key image portfolio. The effect of inaccurate memory has a greater impact in this scenario, particularly for the larger key image portfolios. This is intuitive as if an attacker misses one image, the random nature of login challenges means that the missing image may not reappear for some time.

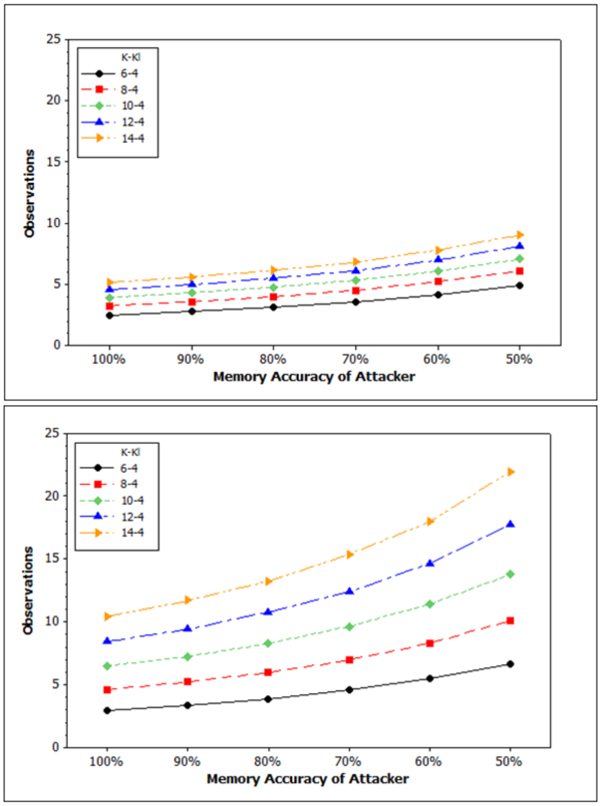

Figure 2: Top: Number of observations required by an attacker to learn enough key images for one successful login. Bottom: Number of observations required by an attacker to learn the entire key portfolio, useful for unrestricted future access. Both the result of 10,000 simulated logins.

The previous model assumes that an attacker is only observing the key images. However it is clear that in viewing a successful login, an attacker with the means for perfect recall (e.g. camera-equipped) can also learn from the images the user does *not* select. In this new scenario, the functionality of the previous model is preserved but the attacker records all images and selections (where $k_l = 4, k = 6, d = 48, d_l = 32$). The goal of the attacker is to obtain enough knowledge to perform a successful login, there are three desirable login scenarios for the attacker: 1) the attacker can identify all key images in the challenge set; 2) the attacker can identify all decoys in the challenge set; 3) The attacker knows the key/decoy role of every image. Fig.3 illustrates the outcomes of 10,000 simulated attacks and the likelihood of each result. On average 84% of attacks are successful due to the attacker concentrating purely on key images presented in the challenge set. In 12% of cases the attacker knows the role of all images in the challenge set, and in only 4% of cases the attacker can identify all decoys in the challenge set (and so derive the keys). It is interesting to note that scenario three is more likely that scenario two. Since the likelihood of scenario three is intertwined with the likelihood of scenario one (which is high), scenario two is least likely since this involves knowing the decoys and *not* the keys.

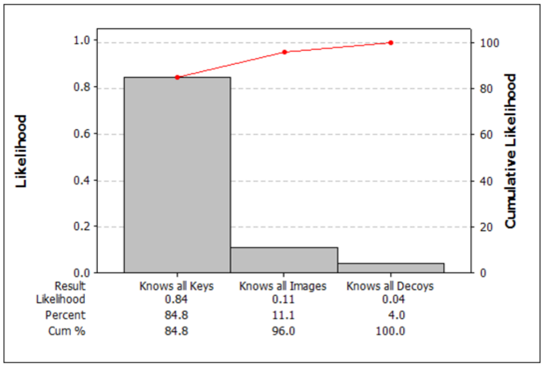

Figure 3: Pareto chart illustrating the likelihood of three desirable login scenarios for an attacker who has carried out shoulder surfing attacks over time. In this model an attacker has perfect retention and observes all images in a challenge set and selections. Based on simulations of 10,000 logins.

4. THE IMPORTANCE OF IMAGES

4.1 Image Source Impacts Security

Recognition-based mechanisms require a collection of images with which to construct login challenges. Images tend to be drawn from one of three sources: *stock* images are drawn from public repositories and tend to be of high technical quality, images of *peers* can be used in centralized systems where other participating users consent to their images being used to comprise authentication challenges of other users [26] [34]. *User* images are drawn from the personal collection of the user and are thought to be the most memorable candidates to comprise key images [36]. The threat arises when systems allocate roles to images based on the location they were sourced, typically key images and decoy images are either drawn from *distinct* sources or *identical* sources.

A high level attack called a *source intersection attack* is possible where key images and decoys are each drawn from **distinct** sources. For example key images are *user* images whereas decoys are *stock* images. To illustrate Let D be all images from the image database known by the attacker to be the source of decoys, and C be images presented in a login challenge, and k the key images. Performing $D \cap C$ reveals $C - k$ decoys with the remainder certain to be key images. Automatic techniques exist to facilitate this attack; PerceptualDiff [32] can determine whether two images are perceptually similar. Parameters of this program can be altered to only permit very strict matches.

Online attacks where an attacker can make better than random guesses involves an *image quality* attack; where the attacker assumes stock images (decoy images) are of higher photographic quality than user images and uses this to make guesses. Where the photographic quality is similar, a vulnerability to *social engineering* remains where an attacker can use knowledge of the user e.g. holidays, family members, hobbies etc. to prioritize guesses. While these opportunities exist, the overwhelming advantage of this configuration is that the memory task for the user is the easiest, as no spontaneous memory association must be formed with the key images as they must simply identify which images are presented from a personal collection, and as such is similar to the task of *automatic detection* [30] in visual search.

The most desirable configuration from a security perspective is unfortunately less desirable from a usability perspective where key images and decoys are drawn from the **identical** sources e.g. keys and decoys are both stock images or both user images. This is because users must create a spontaneous memory association to remember key images and must search images in a *controlled search* [30] fashion. In this case the attacker has no means of making a better than random guess without shoulder surfing. Passfaces [25], VIP [1] and Déjà vu [8] are implemented in this manner according to their respective publications.

4.2 Image Filtering & Usability

Considering our ideal decentralized infrastructure for a recognition-based system, one key operational limitation is that clearly – given the diversity in content and quality of user photograph collections – images must be filtered in an automated fashion for every user of the system. All photograph-based systems of this type in previous research place image filtering out of scope, yet to create a system of this type applicable beyond the lab; automated methods of filtering are needed.

There are a number of problematic characteristics that should be highlighted yet we concentrate our first efforts on *unmemorable* and *visually similar* images. In practice users could be asked to perform the filtering, however over increasingly large sets the visual search becomes tiresome, and the user cannot be relied upon to do this with great accuracy. There are difficulties in developing an automated approach, firstly judgments over which images should be filtered can be subjective, which increases the difficulty in developing a single solution. Also when executing image processing algorithms over a large album there exists a trade-off between the quality of the result and the time taken to execute, which is a concern as filtering must occur in real-time. For this reason it is likely that filtering cannot occur locally on the device due to current processor speed limitations. The following presents a first look at problem cases and considers some automated image processing techniques for their detection:

4.2.1 "‘Unmemorable"’ images

We define *unmemorable* images as those that contain few instances of visual content for the user to remember. Examples include photographs taken in low light, or where the image contains excessive movement. The negative impact of these images would be that if assigned to the user they would be difficult to memorize.

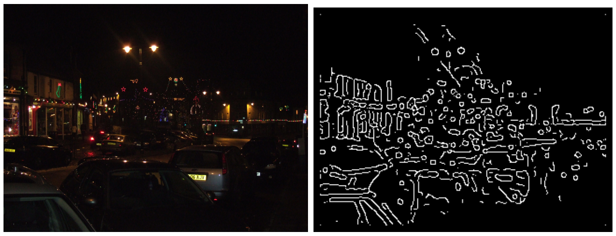

Figure 4: One image before and after canny edge detection. The original (left) is one we class *unmemorable* and the result after edge detection is that less than 10% of the image contains edges.

Canny edge detection [4] is one method that could be used to detect such images. Edge detection is a fundamental image processing technique that extracts areas in an image where brightness

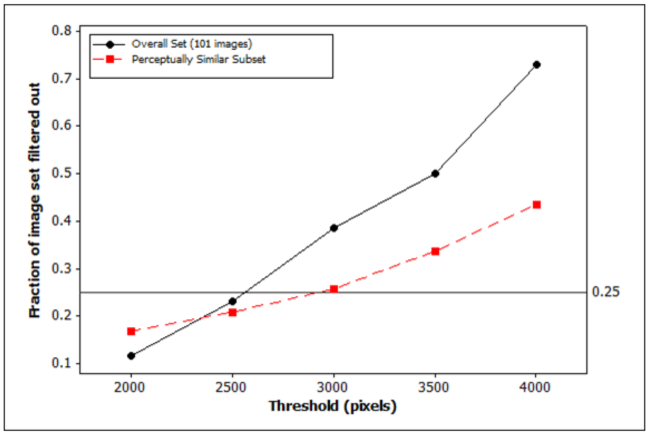

Figure 5: Filtering efficacy using PerceptualDiff at differing thresholds. For threshold calculation we gathered a set consisting of 101 photographs, of which 52 (26 pairs) were deliberately selected as perceptually similar, 49 were random. The goal is to maximize the filtering of the perceptually similar images (at least one per pair), whilst minimizing loss from the overall set. In the best case only 25% of the overall set would be removed (marked on the graph) for 100% of the similar set.

changes sharply. The result of executing this algorithm is a binary image of black and white pixels where white indicates the presence of an edge (fig. 4). By calculating the $\frac{pixels_{white}}{pixels}$ we gain knowledge of the percentage of the image taken up with edges. If this is below a certain threshold we can say there is insufficient *entropy* in the image to be considered further. In our informal testing 10% was found to be an acceptable threshold. This algorithm takes ≈ 90 seconds on an Intel Core 2 Duo running at 2.2GHz to process 50 images of resolution 2848x2136.

4.2.2 Visually Similar Images

The most difficult problem is detecting images that are visually similar. In the worst case key images may appear indistinguishable from one or more decoy images. Wang et al. [38] summarize the various ways humans can assign semantic meaning to images:

1. Semantic types (e.g. landscape photograph, clip art).
2. Object composition (e.g. a bike and a car parked on a beach, a sunset scene).
3. Abstract semantics (e.g. people fighting, happy person, objectionable photograph).
4. Detailed semantics (e.g. a detailed description of a given picture).

It is most likely that users will focus memory strategies on techniques 1,2,3 that all involve remembering verbal generalizations of the image content, rather than detailed descriptions. This means that any images that share a theme, or contain the same people are likely to cause confusion as the user is likely to focus their memory strategy on this generalization that is likely to conflict with other images.

Fig. 6 displays examples of such images. VIP [1] considered one early solution involving manual textual tagging of images; no images with the same tag e.g. flower, could appear in the same login challenge. Disadvantages of this include the requirement of a human in the loop and reliance on the quality of the tags. Renaud and Olsen [27] explored computer vision techniques to eliminate decoys similar to key images in hand-drawn scribbles.

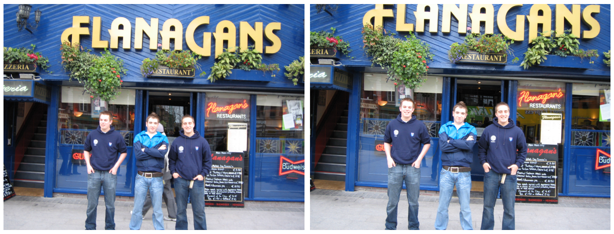

Figure 6: Two photographs taken moments apart that would be too perceptually similar for users to distinguish.

Methods to try and determine computationally if two images are similar are being explored in the field of *content-based image retrieval*. PerceptualDiff [32] is open source software that uses a computational model of the human visual system to determine if two images are *perceptually* similar. The algorithm used is intended to support a more economical execution of the task of *global illumination* in graphics rendering. This was a solution we chose to explore further, as one particularly damaging case is that of fig. 6 where two photographs have roughly the same content yet might be taken from a different angle. We chose to explore this further and modified the code of PerceptualDiff to iterate an entire directory. The procedure was for each image I in the user collection, $I \in \{i_1, i_2 ... i_n\}$, the image is compared to every other excluding itself, if there is a match, I is removed from the set, and the next image in the set is assigned to I. The time taken to process 50 images resized to 80x78 on an Intel Core 2 Duo running at 2.2GHz was ≈ 120 seconds. Fig. 5 illustrates the efficacy of the filtering at different thresholds.

5. IMPLEMENTATION

We developed two implementations to operationalize our observations and analysis, one of *high entropy* and one of *low entropy*. This was to reflect the fact that recognition-based mechanisms – within certain bounds of complexity – are flexible as to the entropy they can usably provide. Both embody the interventions discussed previously, and have the capability to be bootstrapped using images taken directly using the camera functionality of device. A high level overview of the user experience is as follows: firstly the user provides images to a client application that resizes the images and initiates the image filtering process. The resulting set of images is then transferred to the device, from which subsets are then randomly selected to populate the *key image portfolio* and the *decoy image portfolio*. For the two implementations, we decided to ask the user to identify 4 random key images from the key portfolio of 6, amongst a random subset of decoys from the decoy portfolio. This number was informed by intuitive usability concerns raised by asking the user to retain more than this number. The selected images from each portfolio were then shuffled together to comprise the login challenge.

5.1 High Entropy Authentication

The high entropy system is the most resistant recognition-based mechanism to brute force attack we have seen on mobile devices; and provides six-times more entropy than a randomly generated PIN $\binom{36}{4}$. The user interface is visually similar to those seen in previous research [17] [34], where images are presented in a 3x3 grid. Previous research favors this design due to an intuitive keypad mapping available between the on-screen images and the numeric

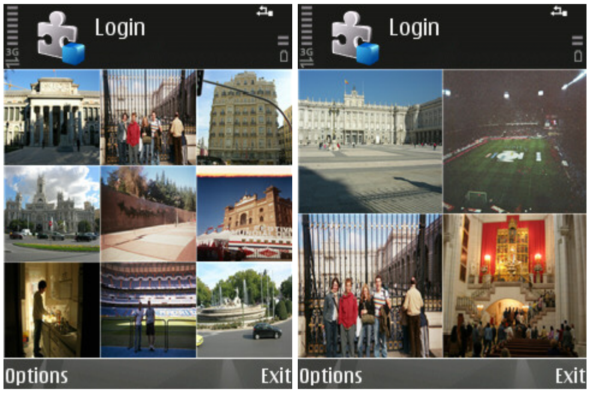

Figure 7: Screenshots of the *High Entropy* and *Low Entropy* versions of the software. In the high entropy images were in a 3x3 grid, and in the low entropy version appeared in a 2x2 grid.

keys on a device keypad. The disadvantage of this configuration is that images are displayed small in size which can cause problems for users with less than perfect vision. In this system 36 images are displayed across 4 screens in the 3x3 grid. The required number of images to populate each image portfolio with respect to the desired ratios are $k_l : k = 6 : 4$ and $d_l : d = 48 : 32$ indicating in total a user must provide at least 54 images after filtering.

5.2 Low Entropy Authentication

The low entropy implementation adopts a new user interface convention for this genre of system on mobile devices. Images appear in a higher quality form, and 24 appear across 6 screens in a 2x2 grid, designed to offer entropy comparable to a randomly generated PIN $\binom{24}{4}$. It could be argued that low screen resolutions seen in many current devices is a transient problem, however designs to accommodate this scenario in the immediate term have implications for usability and accessibility as users are better able to identify images on-screen. In this configuration the chosen image portfolio ratios are $k_l : k = 6 : 4$ and $d_l : d = 30 : 20$ indicating in total a user requires 36 images after filtering to begin using the mechanism.

6. DATA COLLECTION IN THE WILD

Previous work published in the usable security community has suggested laboratory studies can offer misleading results of password recall when compared to field studies [5]. In the years preceding this work in the wider HCI literature, arguments over the validity of results obtained in these two configurations have long existed. This debate is particularly strong in the field of Mobile HCI due to the recognition that mobile devices are typically used in more dynamic contexts than desktop computers, and so should not be evaluated in the same environment. Nielsen et al. [24] argues that field studies are most effective in uncovering issues of cognitive load and interaction style. Rogers et al. [28] comment that field studies are good at demonstrating how people appropriate technologies in their intended setting, but are expensive and difficult to conduct. Of course there are arguments that dispute this added value, Kjeldskov et al. [22] comment that field studies are not worth the added value and a good lab study uncovers just as many usability issues.

To date, controlled laboratory studies have yielded high success rates in all instances for recognition-based systems. Due to only subtle design differences with these systems, we had no reason to believe our mechanisms would perform differently in such a context. This motivated us to shift our attention away from a controlled lab study, to a more pressing issue of how the high entropy and low entropy systems might perform in everyday use on the personal device of each participant. Our goals were to explore:

- What is the typical usage time of these systems?
- What level of user accuracy can be expected?
- How would users appropriate the mechanisms into daily life?

6.1 Method

We recruited 17 participants within the organization through the use of internal mailing lists, with the incentive of free cinema tickets. Although this was reduced to 16 in the early phases of the study as the device of one participant failed and participation could not continue. We chose a between-subjects design and split participants randomly between the two systems, this meant 8 using the low entropy system and 8 using the high entropy version. All participants were smartphone users for regularly accessing work email and other web-based services, which we believe placed them within a key target group given our perceptions of who might need enhanced user authentication.

To initiate the experiment, participants visited the research lab with a portfolio of approximately 80 images either already on their personal device or on removable storage. We chose to ask for 80, as in our informal tests the filtering software removed approximately 20-30 images from a typical collection. This filtering took the form of the execution of software involving a combination of our slightly modified PerceptualDiff [1], canny edge detection, and manual intervention as a last resort (in depth analysis of our filtering mechanisms is ongoing work, fig.5 shows its efficacy). The assigned mechanism was installed on the personal device of each participant, and the resized, filtered images imported automatically. For the enrollment period participants were automatically assigned key images and asked to achieve 3 consecutive correct logins in the presence of the moderator. The mechanism was not actively securing the device, but was an application that allowed the user to test retention of key images throughout the study. Performance data such as success/failure of the login, time/date of the login and login duration were logged automatically, and upon entry of a secret key combination could be output to a file.

Our study design was similar to that of Everitt et al. [13] where participants were sent emails when it was desired they should perform a login. For our purposes, this configuration had the potential to be more effective since all participants read emails on their device, the same location on which the mechanism would be installed. This gave us the opportunity to control the frequency of logins to the system. For the first week participants would be asked to login twice per working day, for the second week participants were asked to login twice per day, every two days. We hoped to gauge the effect of reduced usage on success rates after an intensive first week of usage. On days where a login was required, we sent participants an email at 10am and 3pm to request that they perform a login. Upon receiving the email the participant would be required to open the application and attempt to login. Upon success, the program disappeared into the background, but if a participant could not login after three attempts they would be *locked out*, and offered a logged reminder so they could continue with the study. At the end of the study, the participants visited the lab so that the log file of performance data could be extracted. An example entry from an extracted log file (with added formatting) is the following:

[1] Available at http://homepages.cs.ncl.ac.uk/p.m.dunphy/downloads

```
******************Login***********************
Tue Sep 01 10:42:40 GMT+01:00 2009 Success:false
6.322seconds Keys: 8528634 5027079 7641793 8493317
Decoys: 5328514 836515 2090387 2921383 4222148
288468 7874519 6318636 2298412 4005625 7220047
7546626 9930905 6975851 5766799 3251317 7881604
52555 2854716 8836977
*****************End Login*******************
```

From these entries we were able to analyse statistics of user performance. The sequence of numbers refers uniquely to each image used in the system.

6.2 Results

6.2.1 *Participation*

Of the 16 participants, 10 were male and six female, 12 in the range 18-28 and four in the 29-39 range. Education levels were high with six to BSc level, nine to MSc and one to PhD level. The mobile devices owned by participants were all Symbian S60 devices, with 11 owning the *Nokia N95*, others included the Nokia E61 and E65. Screen resolutions of devices were 320x240 (E61) and 240x320 (N95 and E65).

6.2.2 *Success Rates*

We define the success rate as $\frac{successful\_logins}{number\_of\_logins}$ and calculated this across all logins. We collected 319 logins across two working weeks: 178 from users of the low entropy system and 141 from users of the high entropy version. Of the 319 logins, 30 occurred at the weekend outside of the requested time period, however we included these in our analysis and added data to week one. As the number of logins was not strictly controlled by the mechanism, on average participants in the high entropy group logged in 17.6 times ($\sigma = 7.9$), compared to 22.3 ($\sigma = 5.9$) in the low entropy group.

Participants were accurate in authentication trials, as only two *lockouts* were experienced, both from the high entropy group – both from the same person – in the week of reduced usage. Tables 2 and 3 break down the performance into each week of the study, and fig. 8 illustrates the spread of success rates across the study.

Table 2: Success Rates and attempts recorded for each mechanism during week one where participants were requested to login twice per working day.

	Attempts	Success Rate	Lockouts
High Entropy	85	84%	0
Low Entropy	123	70%	0

Table 3: Success Rates and attempts recorded for each mechanism during week two, where participants were requested to login twice, every two days.

	Attempts	Success Rate	Lockouts
High Entropy	56	67%	2
Low Entropy	55	89%	0

Across both systems, performance remained stable between week one and week two, with 77% in week 1 and 78% in week two. Analyzing success rates per system, this was 77% for *both* systems. From week one to week two the success rate of low entropy users increased from 70% to 89% and was statistically significant (Mann-Whitney $U = 3, p < 0.05$) however the decreased performance in the high entropy system, from 84% to 67%, was not significant. The performance difference between the two systems in week two was significant (Mann-Whitney $U = 3, p < 0.05$). To understand the success rates in more detail we classified logins using a convention similar to that used by Renaud and Olsen [27] (high entropy-low entropy):

- A single successful attempt not preceded by an erroneous attempt (90-107).
- 1-2 failed attempts followed by a successful attempt (18-29).
- A failed attempt not followed by a new attempt within 30 mins (6-3).

We were also able to consider how success might be affected by the time of day participants carried out the login. Across both systems, 145 logins were recorded in the morning (AM), and 174 in the afternoon (PM). This analysis contained no significant differences, nevertheless considering AM logins the success rate was 78%, and for PM logins this was 76%. Users of the high entropy system experienced degradation in performance from AM to PM. In the mornings the success rate was 83% and in the afternoons this fell to 69%. Users of the low entropy had a more consistent performance with an average AM success rate of 73%, and this increased to 75% for PM.

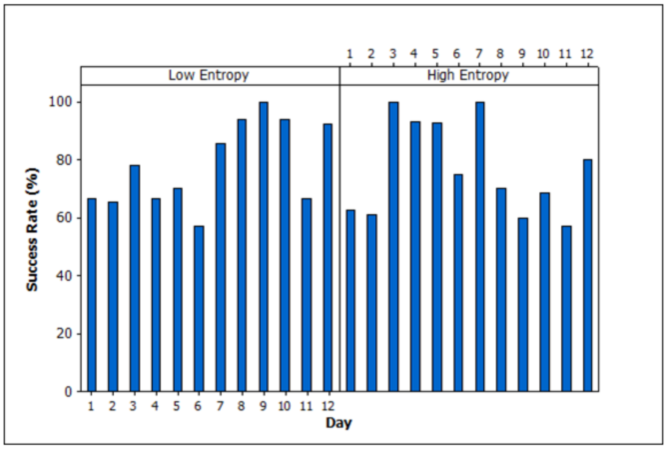

Figure 8: Success rates per day per system. Participants were reminded to use the system less frequently during the second week and this affected success rates in the high entropy group.

6.2.3 *Login Durations*

As well as the accuracy of the recall, login durations were recorded. Login durations were recorded from the user first seeing the login screen until the final key press. The following discussion refers to *successful* login durations. The mean login duration across both groups was 19.8 seconds ($\sigma = 3.8$). Considering the high entropy group alone this was 19 seconds ($\sigma = 4.7$) compared to a mean of 21 seconds ($\sigma = 4.9$) using low entropy. In a two-sample t-test this difference was not significant ($p = 0.366$).

Considering the change in login duration for each week of the study however was most interesting. In the case of both systems, login durations became faster. In the first week the average high entropy login lasted 22 seconds ($\sigma = 4.9$), in the second week this fell to 15 seconds ($\sigma = 3.6$). This was significant in a two-Sample t-test ($t = 4.09$ $p < 0.01$). A similar effect was noted for users of the low entropy system, in the first week the average login was 23 seconds ($\sigma = 4.7$) and in the second week this fell to 17 seconds ($\sigma = 2.7$). Again this was significant in a two-sample t-test ($t = 2.84$ $p < 0.05$). While in the second week there was less data, both changes were significant.

Users of both systems experienced a similar level of improvement in terms of login durations. However comparing systems on a week by week basis did not produce significant results. The fastest instance of a correct login was 9 seconds, with the slowest being 76 seconds. It is likely that the user generating the latter was multi-tasking at the same time as performing the login.

7. REPLAY ATTACK STUDY

This study was designed to supplement the earlier analysis on shoulder surfing to explore the number of observations required for a human attacker to compromise each implementation in the context of the *friend attack* described previously. This phase took place one week into the field study, across two sessions where participants attended with others using the same system. A key benefit of doing this was that participants had already gained one week of experience with their assigned system and so its functionality had become habitual, this equipped participants sufficiently to launch their own observation attacks. Participants were paired and randomly assigned roles as either an *attacker* or a *victim* as seen in other similar studies [21]. The scenario offered to participants was the following:

> You and your new partner are friends; the victim has just called over the attacker over to show a new application on their device. However while both of you are looking at the screen, the device asks the victim to login to continue. The victim does not know their friend is untrustworthy, and is actively trying to learn their key images. So the victim continues to login...

The victim was asked to login to their device, holding it in a way that was not sharing the screen with the attacker unrealistically. After viewing a login the attacker had a decision to make:

1. The attacker has learned enough information to attempt to login, and was given a maximum of 3 attempts to reflect a "three strikes and you're out" policy.
2. The attacker asks the victim to perform another login. This occurred a maximum of 10 times.

After this phase the participants switched roles and repeated the procedure.

7.1 Results

The average number of observations required for imposters to login was 7.5 ($\sigma = 1.8$) against the high entropy system compared to 4.5 ($\sigma = 0.6$) against the low entropy version. A Mann-Whitney test shows this to be a significant difference ($U = 0, p < 0.05$). Referring back to the simulations presented in fig. 2, high entropy participants performed as well as attackers with 30% memory accuracy (not detailed on the graph), whilst low entropy participants were 50% accurate. This difference is reasonable since the shoulder surfing task for high entropy participants was more difficult, more images were displayed and of a lower quality. There were 4 instances where participants were unable to login as an imposter, 3 using high entropy and 1 using low entropy. The mean login durations of legitimate users was 16 seconds ($\sigma = 9$) and 23 seconds ($\sigma = 24$) for an imposter ($U = 469, p < 0.01$). We calculated this using knowledge of the time and date of this study in the system log. Fig. 10 illustrates the difference in the distribution of performance between legitimate users and imposters.

Figure 9: The context of the replay attack study, participants could sit or stand. This would give an impression of how human memory compared to our shoulder surfing simulations.

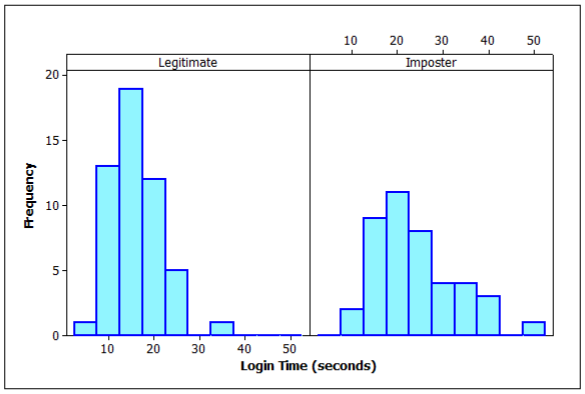

Figure 10: Histograms of the login durations collected from legitimate users and those posing as imposters. There is a clear difference between the login durations of legitimate users and *imposter* users. This could be used to inform design of a *login timeout*.

During the study we did observe that the key image portfolio did provide some temporary resistance to an impostor login. In the context of a lunchtime attack this temporary delay could resist attack for a significant period of time depending on the access gained to the device by an attacker.

8. QUALITATIVE EVALUATION

After the study we distributed questionnaires and had informal discussions with all participants to elicit opinions of the mechanisms and some security practices in general. We hoped that after using the mechanisms on their own devices intensively for two weeks they would have stronger and more interesting comments than if we had performed a short lab study. A selection of questions and comments are displayed below:

Do you use PINs on your device?
57% responded yes and 43% no. Interestingly, when asking those that responded negatively if there were items on their device they would consider to be private, everybody responded that there were a number of things. This suggests that those users attribute security of their device to their assumed possession.

How do you rate the time required to login?
This is an important question as the most crucial driver to user acceptance is often the convenience of use. A disadvantage of graphical password schemes more generally is that it typically takes much longer to authenticate than PIN, due to the visual search required. 64% of users said the time required to login was acceptable, while the remaining 36% thought the time cost was unacceptable.

Did using the mechanism feel secure?
Providing the users with a feeling of security is something that the mechanisms both lacked. 79% of Participants indicated they felt more secure using PINs but could not come up with concrete reasons why. This is possibly because of the transparency of the mechanisms and their game-like nature.

Some interesting and recurring comments received were the following:

> *"‘I would prefer to choose my own images"’*
>
> *"‘When I was walking around I had to concentrate much more than when using a normal PIN, for that I don’t need to see the keypad"’*
>
> *"‘During one login a particularly funny image appeared so I showed it to my colleague"’*
>
> *"‘It’s much harder to crack numbers than images isn’t it?"’*

The majority of users expressed a desire to choose their *key images*. It is common knowledge that users are likely to choose alphanumeric passwords and PINs in predictable ways [23]. This trend has been noted in graphical password studies too [7] [9] [35]. By distribution of password selection being anything but uniform, an attacker can prioritize a guessing attack by any perceived biases. One feasible concession to this rule could be the system choosing a random subset of images – slightly larger than the number of required key images – and allow user selection from within this set.

The next two comments are unlikely to have been made in the context of a lab study. A number of users lamented the fact that the mechanisms demanded their visual attention for use. With PINs they made it clear they were able to input digits without viewing the screen (due to the tactile nature of the keypad) and multi-task more effectively. Next a number of users commented how they treated the logins as an enjoyable means to view the images on their device; one in particular commented how they would show particularly amusing images that appeared to work colleagues. This hints that our hypothesis of a *friend attack* is potentially realistic.

The final comment could provide an insight into why many users felt the mechanisms were less secure than PIN. This user has an incorrect mental map of what makes an authentication secret *crackable* and perceives the game-like nature of the mechanism to be a reflection of its seriousness regarding security. In addition, all their previous experience with mobile device authentication had been with PINs, which is likely also greatly informs their preference. A similar effect was noted in a user evaluation of device pairing methods [37] where users associate *more difficult* with *more secure*.

9. DISCUSSION

9.1 Study Limitations

The two week duration of the study was relatively short, however this was chosen as we hoped it would be sufficient to provide a glimpse of how users would appropriate the systems into their daily routines, and enable us to provoke some strong opinions. As a result, the success rates reported here are likely to be under-estimates, as user performance was not given a long time to stabilize. Any longitudinal study should make observations of user behavior in a real usage context; in our setup the mechanism was not actively securing the device or anything else of value. As has been commented on in previous security user studies, this can impact the motivation of participants to perform the required tasks.

The *replay attack* study attempted to recreate a shoulder surfing scenario, and we must consider the ecological validity of this method. Shoulder surfing is a difficult phenomenon to recreate in an artificial setting, as typically a victim is unaware of an attack taking place. However, our setup potentially fits well to our *friend attack* threat model as the victim *would* know they were under observation. An alternative approach to recreating such a scenario could involve observers viewing a video, but this could be subjected to the same arguments of ecological validity.

We hope to use the hindsight gained by this field study to inform design of future field evaluations in this area. In designing the experiment, we were confronted with a number of design issues that are relevant to others considering similar studies. Firstly participants expressed *apprehension towards full deployment* on personal devices. Our participants were active smartphone users and busy members of the organization, and due to the increasing importance of mobile devices they were concerned that unexpected software problems could block them from working. Secondly recruitment is based on *specific criteria*, participants should own devices on the targetted platform (or devices should be provided). This can reduce the size of the participant pool considerably. Using multi-platform programming languages such as Java can help, although ability to make low-level system calls is reduced. Finally, the *devices on a particular platform can be diverse*; one platform can contain different devices that can provide different user experiences e.g. screen resolutions and keypads.

9.2 Study Reflections

Overall we collected more attempts than we requested, but this was inevitable given the novelty of the mechanisms and the fact the number of logins was not limited. The maximum number of logins recorded from a particular person, on a particular day was six. Accuracy was similar across both mechanisms, in addition to the success rates, 60% of logins from the low entropy group could be categorized as *correct first time* compared to 63% of high entropy logins. All the accuracy data indicates that one intuitive hypothesis that participants of the high entropy system would have a reduced performance over those performing the low entropy task does not appear to be valid. After considering that this could be an anomaly attributed to the amount of data collected, another possibility is that in a visual search task, practice can decrease capacity demands [10]. So one explanation could be that once participants have gained enough practice with the mechanisms, performance is not linked to the entropy of the mechanism and is similar within certain bounds. This is partially supported by the fact that successful login durations were faster between week one and two.

The results of the replay attack study show both mechanisms to be vulnerable to shoulder surfing to some extent. *Impostors* observed on average 7.5 logins using the high entropy version before being able to login, whereas using the low entropy version this was an average of 4.5 logins. While the scenario did not take into account the likely time gap between a lunchtime attacker observing a challenge and having the opportunity to login, the results at least suggest a lower bound to attackers purely using human memory to record images. Our computer models describing a shoulder surfing attack illustrate the extremes of the shoulder surfing effect and could be used to inform design of a *password expiry* mechanism as values presented suggest the best-case observation scenarios to compromise one login, and an entire key image portfolio.

Desirable additional functionality for systems of this genre is to reason about whether a login is not being performed by the legitimate user. This is because unlike alphanumeric passwords, simply viewing the images presented in a login challenge may still convey information to an attacker. One means to achieve this is through analysis of the login durations, as an impostor should take more time to complete a login than legitimate users. This possibility was supported in our replay attack study, where imposter login durations were significantly longer than those of legitimate users. This could indicate that a *time-out* on logins is reasonable where the system believes the current login to be taking longer than usual.

Considering observations on both systems, the image portfolio concept explored is not a scalable method. Increasing the size of the key image portfolio does increase difficulty for the attacker, but perhaps most damagingly for the user. This effect is intuitively more prominent with larger sizes of the key portfolio, and is not a feasible solution unless a method giving users a secure reminder can be developed. Also in systems where key images and decoy images are drawn from the same source, it is critical that an effective enrollment process be designed, such as that seen in the Passfaces [25] online system to provide a more longitudinal encoding.

Finally, multi-password interference is a concern in this context despite the lack of attention we have given to the problem. Everitt et al. [13] demonstrate the degradation in user performance when managing more than one visual password sequence. Our intuition is that recognition-based systems can provide better resistance to interference over pure recall-based methods, as the login challenge innately constrains user actions. However in the context of the architecture presented in this paper, it is likely that completely different sets of images must be used for each different system. This means that any further development has to consider that users may upload similar images to different systems. A deeper consideration of this problem is future work.

10. CONCLUSION

We have gathered results that give a first suggestion of real-world performance levels to be expected from two recognition-based graphical password systems (of varying entropy) on mobile devices. Users were assigned key images and despite having relatively little training, performance was good irrespective of the mechanism as success rates were 77% for each system. Users of the high entropy version experienced two lockouts, where access was "‘blocked"’ after 3 incorrect attempts. Of course, in a study context outside of the lab this could be attributed to any number of reasons. The success rates reported are similar to those reported in a field study of Passpoints [5] where success rates ranged from 78% - 83%. Other field studies that provide means of comparison include Passfaces (95%) [2] and Dynahand (97.4%) [27]. While this may seem like a large difference, the low entropy system exhibited higher entropy than each of the aforementioned systems (excluding Passpoints), and it is likely that a longer study would have allowed any performance extremes to stabilize.

In a test that explored the shoulder surfing capabilities of user study participants we intuitively discovered that users of the high entropy system – where images were smaller and of reduced quality – needed on average 7.5 observations to obtain a successful login compared to 4.5 logins where participants attacked the low entropy system – where images were displayed larger and in higher quality. An attack on a PIN is likely to only need 1-2 observations, meaning visual authentication solutions using photographs are likely to provide more resistance in practice to shoulder surfing. One contributing reason is that all users of PINs are familiar with the *alphabet* from which PINs are drawn, which is not the case where login challenges are constructed with the personal images of the user.

Despite the increasing presence of biometrics for user authentication on consumer electronics e.g. laptops, knowledge-based authentication systems are likely to remain attractive due to being purely software-based solutions. Graphical password systems based on recognition potentially have a role to play in this area, due to accurate user performance in previous studies, including this one. One key limitation however, is that login durations recorded for our systems – and others – are still too long. User acceptance is often driven by convenience and login durations of approximately 20 seconds are unattractive to many users.

In this work we have concentrated particularly on mobile devices, however deployment of recognition-based mechanisms on the Internet presents different challenges [27]. Another goal of these systems should be to contribute to the alleviation of the password management problem currently experienced on the Internet. In order to make this a feasible goal, we feel image processing has an important role to play to improve the scalability of these systems. The image filtering mechanisms deployed in this work were not sufficient to entirely solve the filtering problem, particularly in determining image similarity due to its subjective nature. Future contributions will be the further exploration of image analysis to provide automated and intelligent selection of images for particular graphical password systems.

11. ACKNOWLEDGMENTS

This work was carried out during an internship carried out by the first author at the Nokia Research Center in Helsinki.

12. REFERENCES

[1] A. D. Angeli, L. Coventry, G. Johnson, and K. Renaud. Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems. *Int. J. Hum.-Comput. Stud.*, 63(1-2):128–152, 2005.

[2] S. Brostoff and A. Sasse. Are Passfaces more usable than passwords? A field trial investigation. In *HCI 2000: Proceedings of People and Computers XIV - Usability or Else*, pages 405–424. Springer, 2000.

[3] A. P. Bryan Parno, Cynthia Kuo. Phoolproof phishing prevention. In *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 1–19. Springer, February 2006.

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[5] S. Chiasson, R. Biddle, and P. C. van Oorschot. A second look at the usability of click-based graphical passwords. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 1–12, New York, NY, USA, 2007. ACM.

[6] N. Clarke and S. Furnell. Authentication of users on mobile telephones - a survey of attitudes and practices. *Computers and Security*, 24(7):519 – 527, 2005.

[7] D. Davis, F. Monrose, and M. K. Reiter. On user choice in graphical password schemes. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, Berkeley, CA, USA, 2004. USENIX Association.

[8] R. Dhamija and A. Perrig. Déjà Vu: a user study using images for authentication. In *SSYM'00: Proceedings of the 9th conference on USENIX Security Symposium*, Berkeley, CA, USA, 2000. USENIX Association.

[9] A. E. Dirik, N. Memon, and J.-C. Birget. Modeling user choice in the PassPoints graphical password scheme. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 20–28, New York, NY, USA, 2007. ACM.

[10] J. Duncan and G. W. Humphreys. Visual search and stimulus similarity. *Psychological review*, 96(3):433–458, July 1989.

[11] P. Dunphy, J. Nicholson, and P. Olivier. Securing Passfaces for Description. In *SOUPS '08: Proceedings of the 3rd symposium on Usable privacy and security*, New York, NY, USA, 2008. ACM.

[12] P. Dunphy and J. Yan. Is FacePIN secure and usable? In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 165–166, New York, NY, USA, 2007. ACM.

[13] K. M. Everitt, T. Bragin, J. Fogarty, and T. Kohno. A comprehensive study of frequency, interference, and training of multiple graphical passwords. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 889–898, New York, NY, USA, 2009. ACM.

[14] J. Flatley. Mobile phone theft, plastic card and identity fraud: Findings from the 2005/06 british crime survey. *Home Office Statistical Bulletin*, 2005.

[15] P. Golle and D. Wagner. Cryptanalysis of a cognitive authentication scheme (extended abstract). In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 66–70, Washington, DC, USA, 2007. IEEE Computer Society.

[16] V. Harrington and P. Mayhew. Mobile phone theft. *Home Office Research Study 235*, 2001.

[17] E. Hayashi, R. Dhamija, N. Christin, and A. Perrig. Use your illusion: secure authentication usable anywhere. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security*, pages 35–45, New York, NY, USA, 2008. ACM.

[18] L. N. Hoang, P. Laitinen, and N. Asokan. Secure roaming with identity metasystems. In *IDtrust 2008, Proceedings of the 7th Symposium on Identity and Trust on the Internet*, pages 36–47, March 2008.

[19] Ian Jermyn and Alain Mayer and Fabian Monrose and Michael K. Reiter and Aviel D. Rubin. The design and analysis of graphical passwords. In *SSYM'99: Proceedings of the 8th Conference on USENIX Security Symposium*, Berkeley, CA, USA, 1999. USENIX Association.

[20] A. K. Karlson, A. B. Brush, and S. Schechter. Can i borrow your phone?: understanding concerns when sharing mobile phones. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1647–1650, New York, NY, USA, 2009. ACM.

[21] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. Nicholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 1093–1102, New York, NY, USA, 2010. ACM.

[22] J. Kjeldskov, M. B. Skov, B. S. Als, and R. T. Høegh. Is it worth the hassle? exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *Mobile HCI*, pages 61–73. Springer-Verlag, 2004.

[23] D. V. Klein. "foiling the cracker" – A survey of, and improvements to, password security. In *Proceedings of the second USENIX Workshop on Security*, pages 5–14, 1990.

[24] C. M. Nielsen, M. Overgaard, M. B. Pedersen, J. Stage, and S. Stenild. It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*, pages 272–280, New York, NY, USA, 2006. ACM.

[25] Passfaces Corporation. The Science Behind Passfaces. *Company white paper*.

[26] T. Pering, M. Sundar, J. Light, and R. Want. Photographic authentication through untrusted terminals. *IEEE Pervasive Computing*, 2(1):30–36, 2003.

[27] K. Renaud and E. Olsen. Dynahand: Observation-resistant recognition-based web authentication. *Technology and Society Magazine, IEEE*, 26(2):22–31, Summer 2007.

[28] Y. Rogers, K. Connelly, L. Tedesco, W. Hazlewood, A. Kurtz, R. Hall, J. Hursey, and T. Toscos. Why it's worth the hassle: The value of in-situ studies when designing ubicomp. pages 336–353, 2007.

[29] M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'Weakest Link' - a Human/Computer Interaction Approach to Usable and Effective Security. *BT Technology Journal*, 19(3):122–131, 2001.

[30] W. Schneider and R. M. Shiffrin. Controlled and automatic human information processing: I. detection, search, and attention. *Psychological Review*, 84:1–66, 1977.

[31] R. Shepard. Recognition memory for words, sentences and pictures. *Journal of Verbal Learning and Verbal Behavior*, 6:156–163, 1967.

[32] Sourceforge. 'perceptualdiff' (2008) last accessed 02/12/2008. http://pdiff.sourceforge.net/.

[33] X. Suo, Y. Zhu, and G. S. Owen. Graphical Passwords: A Survey. In *ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, pages 463–472, Washington, DC, USA, 2005. IEEE Computer Society.

[34] T. Takada, T. Onuki, and H. Koike. Awase-e: Recognition-based image authentication scheme using users' personal photographs. *Innovations in Information Technology, 2006*, pages 1–5, Nov. 2006.

[35] J. Thorpe and P. V. Oorschott. Human-Seeded Attacks and Exploiting Hot-Spots in Graphical Passwords. In *16th USENIX Security Symposium, Aug.6-10*, 2007.

[36] T. S. Tullis and D. P. Tedesco. Using personal photos as pictorial passwords. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1841–1844, New York, NY, USA, 2005. ACM.

[37] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Proceedings of the Usable Security 2007 (USEC 07) Workshop*, volume 4886 of *Lecture Notes in Computer Science*, pages 307–324, Lowlands, Scarborough, Trinidad/Tobago, February 2007. Springer.

[38] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[39] D. Weinshall. Cognitive authentication schemes safe against spyware (short paper). In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 295–300, Washington, DC, USA, 2006. IEEE Computer Society.

[40] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon. PassPoints: design and longitudinal evaluation of a graphical password system. *Int. J. Hum.-Comput. Stud.*, 63(1-2):102–127, 2005.