User Study, Analysis and Usable Security of Passwords based on Digital Objects

Robert Biddle, Mohammad Mannan, P.C. van Oorschot, Tara Whalen

***Abstract*—Despite all efforts, password schemes intended to deploy or encourage the use of strong passwords have largely failed. As an alternative to enable users to create, maintain and use high quality passwords willingly, we propose Object-based Password (*ObPwd*), leveraging the universe of personal or personally meaningful digital content that many users now own or have access to. ObPwd converts user-selected digital objects to high-entropy text passwords. Memorization of exact passwords is replaced by remembering password objects. We present the design details, variants, and usability and security analysis of ObPwd; and report on the results of a hybrid in-lab/at-home user study on 32 participants. The results suggest the scheme has good usability, with excellent memorability, acceptable login times, and very positive user perception, achieved while providing strong security for the threat context explored. We believe this work lays the foundations for a promising password selection paradigm.**

I. Introduction and Motivation

Text passwords remain ubiquitous, despite endless criticism. People consistently choose 'weak' passwords [7] for many reasons, including users trying to manage on average 25 password-protected accounts [8]. Losing strategies include blaming users, and imposing complex password rules. Some claim that choosing weak passwords (despite repeated advice otherwise) is a rational economic response [12]. As alternatives to passwords come with their own deployment barriers, passwords appear likely to continue to dominate user authentication for the foreseeable future, despite repeated rumors of their impending death.

Some argue [12, 8] that strong passwords are non-essential for preventing automated online dictionary attacks; e.g., password-protected sites can present challenge CAPTCHAs after (e.g., three) failed attempts, or lock out the targeted account temporarily. However, the latter can affect legitimate users, and CAPTCHA schemes are regularly defeated by improved attacks in the artificial intelligence arms-race, by human solvers, or bypassed due to implementation flaws. Bulk guessing attacks [8] may yield access to accounts when attackers know many valid userids, even if lock-out rules are used. A recent exploration [25] of the feasibility of online dictionary attacks highlights the critical security vulnerability of human-generated password.

To address these issues, we introduce *ObPwd*, an object-based password scheme to generate passwords used infrequently, used in common web authentication, or used to access encryption keys. The basic idea is as follows. Many users currently possess a large collection of digital content such as photos, audio recordings, videos, documents and email messages. Much of this content is *mobile*: stored on personal devices (e.g., USB drives, laptops), protected remote servers (e.g., email providers with HTTPS access), or uploaded to personal sites (some password-protected). ObPwd generates a password from such items by computing a hash from the user-selected object then converting the hash bitstring, by known techniques [11, 20], to an appropriate password format, e.g., a string of keyboard characters or a word sequence.

In place of remembering exact passwords, users only need a strategy to remember *which* password object they chose (e.g., hints for an image, video, or text passage from a web page/document). The proposal moves the user from "what you know" to a hybrid that requires both access to a digital object ("what you have"), plus knowing which object to use ("what you know"). Recalling or browsing through personal and emotionally meaningful content appears to be more satisfying and rewarding than complying with standard password guidelines and procedures. Users can use affective objects for authentication in existing password-protected sites. While being more satisfying does not itself increase security, the underlying entropy of digital objects (in the threat model assumed) together with the rejection of password advice in traditional schemes, provides ObPwd security and usability advantages. The generated password can be written down in a 'secure' place, or re-created from content when needed. ObPwd requires no modifications to password system interfaces, and the system side (remote or local) need not be aware of ObPwd, facilitating deployment.

Our contributions include the basic design and variants of an object-based drop-in replacement for text password schemes; usability and security analysis; and results and interpretation of a 32-participant hybrid user study, including exploration of performance, user acceptance and multi-password interference. Implementations of the new mechanism are publicly available as a Firefox browser extension, and stand-alone applications in Microsoft Windows, Mac OS X, Linux, and Android.[1]

II. Object-based Password Scheme and Variants

Operational assumptions and steps. We assume that password-generating objects are selected mainly from (i) a user's personal, locally stored digital content; and (ii) a large and preferably stable public collection of files (e.g., pdf files and text strings therein), including from large academic digital archives. The idea is that the inaccessibility of private content, and/or the large size of pools of source objects, precludes effective offline dictionaries. Ideally users would not choose objects from their (publicly accessible) personal website or public profiles in social networking sites. (However, salting such objects may adequately reduce risks; see Variant 1 below.) To enable *access-from-anywhere*, users carry password-generating objects with them, or have online access to the objects (e.g., email messages). Passwords generated by ObPwd, and hints (text reminders) to objects, can optionally be written down.

ObPwd requires a malware-free user environment (as do text passwords), and may be at risk to network-based observation if public password objects are retrieved for use in web login. Attacks and

R. Biddle, P.C. van Oorschot and T. Whalen are with School of Computer Science, Carleton University, Ottawa, Canada

M. Mannan is with Electrical and Computer Engineering Dept. University of Toronto, Toronto, Canada

Version: February 11, 2011. Authors ordered alphabetically. Contact author: m.mannan@utoronto.ca. Parts of this work appeared in preliminary form in a workshop paper [13]; an extended version is also available [14].

[1] The ObPwd FAQ and download page is available at http://www.ccsl.carleton.ca/~mmannan/obpwd/.

countermeasures are discussed in Section VI-B. ObPwd involves the following steps:

1) A user selects a memorable object M from local media or the web. To preclude offline dictionary attacks and predictable object prefixes, M must exceed a minimum size (e.g., $m = 160$ bytes). To bound the time to hash large objects (e.g., a 4GB movie), M is also truncated (e.g., n = $100,000$ bytes).
2) The user indicates the selected object to the ObPwd tool (e.g., through a file dialog, or drag and drop interface), which generates the hash $H = h(M)$, where h is a cryptographic hash function.
3) Generate a password $pwd = \text{Hash2Text}(H)$, where Hash2Text is a function converting binary hash output to keyboard character strings [20]. H may be truncated depending on the required size of pwd. The object and password should not be stored on the same media as the protected content. For web login, pwd may require special encoding [20].
4) The password may be copied to a desired site/application, written down or saved, or used to generate high-entropy encryption keys.

The following ObPwd variants can increase security, albeit decreasing usability; their suitability depends on target environments. *Variant 1 (salted ObPwd):* Use $H = h(M, s)$, appending a user-selected *salt* s, e.g., a weak password, as an input. The cost: memorizing s. If used only rarely, s may be written down in a safe place, rather than memorized. A user-selected n in place of the current limit ($n = 100,000$ bytes) may serve as an alternative form of salt. *Variant 2 (multi-stage/mixed-object ObPwd):* Use $H = h(M_1, M_2, ...)$ where M_i are multiple user-selected objects (e.g., two photos, or a music file and text string). Public and private objects could also be mixed. *Variant 3 (anti-phishing ObPwd):* Use $H = h(M, url)$, appending the URL of a target site (cf. [20]). This can be implemented in a browser extension without user involvement.

III. OBPWD USER STUDY: METHODOLOGY

Here we report on a formal user study, designed to explore usable security of ObPwd (as a Firefox extension, simplified to accept only local files, and hard-coded to generate 12-character alphanumeric passwords).

A. Study Goals and Setup

Our hybrid study combined two lab sessions with a phase conducted in participants' regular "at-home" environment to approximate an ecologically valid context of use. The lab sessions were designed to carefully measure usability factors, such as effectiveness (e.g., login success rate), efficiency (e.g., login times), and satisfaction (e.g., perceived ease-of-use). The at-home component allowed an examination of naturalistic behavior outside of the controlled lab setting. We also wished to develop an understanding of the types of files users select to create their passwords, and their rationale. For example, do users pick the same type of file for all logins? Are they concerned mainly with convenience, or password memorability?

Memory load was another important aspect explored. The lab study had two sessions, 7–10 days apart, to evaluate participants' ability to recall passwords over time. The design of our study also incorporated password interference (cf. [3, 6]), asking participants to use a different password for each of 8 study websites.

Test websites. In each lab session, participants logged into four websites created for study purposes, varied in appearance, content, and implied level of sensitivity. Participants followed a role-playing scenario, as a new purchasing department employee required to create new web accounts and log in to sites such as a credit union and a news blog.

Participants. Participants were recruited within a university campus, using email lists and an institutional research study recruitment website. Participants were required to use Firefox regularly, and to provide a laptop that they used regularly, on which they consented to install the ObPwd extension.

Thirty-two participants (20 female, 12 male) completed all parts of the study. Participants ranged in age from 16–59, with a mean age of 22; their age distribution was as follows (range, count): (16–19, 19), (20–29, 10), (30–59, 3). 75% of participants used the web more than 10 hours per week. On a 1-7 scale of concern about the security of passwords online (1: "not at all concerned," 7: "very concerned") the mean rating was 5.57. 16% of participants had programming experience, 41% had created web pages, and all had installed software. Five participants used Mac and the rest used Windows (no Linux users).

B. Procedure

The user study consisted of three parts: two lab sessions and one at-home component. The lab sessions took place in an office; participants brought their own laptops, used to complete all experimental tasks.

a) Lab Session 1. After informed consent was obtained, participants were given a brief demonstration which introduced them to the ObPwd Firefox extension. While in real-life, users may not get any training, we provided a brief demo in order to explore performance among users with a basic familiarity. (However this brief training is no match to the countless years of experience users have with text passwords.) Participants were then asked for permission to install the extension on their laptop. They were also given some guidelines on choosing appropriate files for creating passwords, e.g., warnings about dynamic files and the risk that if others had access to that file, could re-create that password and potentially break into user accounts.

Participants completed two practice tasks with ObPwd before the actual trials began. Each trial was conducted on all 4 websites in each lab session. Websites were logged into in random order. Participants were asked to use a different password object file for each website (as commonly advised), although this was not enforced. (Any password re-use was detected, however.) Participants were also informed that they were permitted to record password information if desired, including hints to help them remember the file, or the generated passwords, on paper or in a file on their laptop. However, they were requested not to use Firefox's password manager to record passwords, as this would preclude use of ObPwd itself for entering passwords. (During the practice session, participants disabled the Firefox option of remembering passwords for sites on our study domain.) The trials were in two phases. The first had six parts:

1) *Create*: Participants registered on a website when visiting it for the first time, using a unique 8-character username and a password they created from their own files using ObPwd. This password was entered into the password field (automatically, or through copy-paste), then entered again for verification.
2) *Confirm*: Participants entered their username and password on the Login page of the site, to confirm that they could successfully log in. If login was unsuccessful, they were allowed to attempt login as often as desired, and could reset their password if needed. After a successful confirmation, they logged out.
3) *Distraction*: Participants were asked to browse their file system seeking files fulfilling specific criteria (e.g., filenames starting with a specific letter). If not found, they could stop searching

after one minute. This task was designed to flush working memory related to their password object and simulate a longer passage of time by focusing attention on a separate task (cf. [16]).

4) *Login*: Participants attempted to log in to the site a second time, again trying as often as desired with the option of resetting their password if needed.
5) *Website information task*: After a successful login, participants were directed to find a fact on the site, e.g., a phone number. As above, this simulated passage of time, and familiarized them with the simulated site that they would be revisiting later in the session; once the fact was located, they logged out.
6) *Questionnaire*: Participants answered questions on 7-point semantic differential scales (from "very easy" to "very difficult") about the password file they selected, the ease of creating passwords and using the extension to login, and their perceived likelihood of being able to successfully login in one week's time.

After the above steps, the second phase of trials began. Participants revisited the sites a second time, again in a random order. This phase had three parts: login, website information task, and questionnaire (similar to those in phase one). At the end, each was given information about the at-home component.

b) At-home component. This portion was designed to simulate realistic use of ObPwd (outside the lab environment). Participants were asked to complete two sets of tasks. One set was to choose three real-world websites, try to use ObPwd to create passwords for these and log in, and answer a short questionnaire The second set was to revisit the same four sites from the lab session (in random order), log in exactly once to each, and complete a short questionnaire. Participants could interleave tasks from these two sets.

c) Lab Session 2. Participants returned for a four-part second lab session 7–10 days after their first. In Part 1, they re-visited the four sites from the first session (in a random order), tried to log in, and answered a short questionnaire. Part 2 repeated the steps from the first session, on four new websites (to allow us to compare attitudes and behaviors across two sessions). Part 3 consisted of a second round of logins on the new sites. In Part 4, participants answered a comprehensive questionnaire about their experiences with ObPwd. This included questions about managing multiple passwords, notable positive or negative aspects of ObPwd, and scales to measure several usability factors. Because ObPwd can also be used for secondary authentication, such as through Personal Verification Questions (PVQs), a series of questions on PVQs was also included.

d) Post-study questionnaire. Participants could opt-in to a post-study questionnaire, sent approximately four weeks after their first lab session. Those who opted in were asked, by email, about their continued use of ObPwd (if any) after the completion of the formal study.

IV. User Study Results

As customary in HCI studies, we first present the results, followed by the interpretation in Section V. The results consist of descriptive statistics; inferential statistical analysis was not performed, as in this study, ObPwd was not formally evaluated against an alternative authentication scheme.

a) Password creation times. The time taken to create a password was calculated from the time that the Registration page request was received at the web server, to the time that the "register" request (from a button on that Registration page) was recorded in the website database. This time includes typing an 8-character username, selecting a file using ObPwd, and entering that password twice (in two password fields). Participants who used the copy-paste option of ObPwd could paste the password twice in succession, but those who used the auto-paste had to recreate the password a second time. (As passwords were masked with asterisks on the page, users could not copy them directly from that field.) The times are shown in Table I.

	Mean	Median	Std. Dev.
Sites: Lab Session 1	52.3	46.0	32.1
Sites: Lab Session 2	35.6	29.1	21.6

TABLE I
TIME (IN SECONDS) TAKEN TO CREATE A PASSWORD

Sites	Action	Mean	Median	Std. Dev.
Lab Sess. 1	Confirm	18.5	15.5	11.0
	Login #1	18.4	15.4	12.6
	Login #2	26.3	20.1	25.8
	At-home login	43.4	24.2	61.5
	Sess 2 login	33.3	25.1	39.9
Lab Sess. 2	Confirm	14.5	13.1	7.4
	Login #1	21.8	14.6	24.2
	Login #2	19.3	16.1	12.9

TABLE II
TIME (IN SECONDS) TAKEN TO LOG IN

b) Login times. The time taken to login was calculated from the time that the Login page request was received at the web server, to the time that the participant successfully logged into the site. This time is cumulative: it includes time taken for any failed attempts, until the point when a successful login occurs. It includes the typing of an 8-character username, entering the password (e.g., using ObPwd to locate the file and re-create the password) and clicking on the "Login" button on the web page. The sites in Week 1 were logged into on five occasions: when confirming the password (initial login); twice in Lab Session 1; at home; and revisiting during Lab Session 2. The sites in Week 2 were logged into on three occasions: confirming, and twice in Lab Session 2. The times are shown in Table II.

	% Success 1st attempt	% Success within 3 attempts	# Passwd mismatch errors	# Passwd resets
Sess. 1 sites	65	90	42	6
Sess. 2 sites	93	99	14	2

TABLE III
LOGIN SUCCESS RATE AND ERRORS

c) Login success rate and errors. A login attempt was any instance when the user clicked on the "Login" button, whether or not that attempt led to a successful login. In Lab Session 1, with 32 participants and 4 websites, the optimal number of logins would be 128 (i.e., each person four times). The actual number of attempts differed from this for two reasons: (i) any failed attempt added to the total; and (ii) some participants neglected to log out between steps of the trial; e.g., they might log in and fail to log out before the distraction task, then complete the web information task without logging in a second time. For each such instance, one login attempt would be missing. Because the password hashes were stored, we could track when a participant tried using a password created for a different account (a "password mismatch" error). Participants could also attempt to log in as often as desired; we report here the percentage of login attempts successful on the first try, and those that were successful within three attempts (allowed on many sites). Results in Table III include the total number of password mismatch errors across all login attempts, and the total number of times participants reset their passwords during the entire study.

File type	Lab Session 1	Lab Session 2	At-home
Image	58	58	40
Document	30	36	16
Music/audio	23	30	26
Video/movie	12	3	9
Other	5	1	5

TABLE IV
TYPES OF FILES USED TO CREATE PASSWORDS

d) Files chosen for passwords. Participants identified the type of file they chose for each password (e.g., music file, document). These file types were aggregated into categories, as listed in Table IV. The total number of possible passwords generated in each of the two lab sessions was 128 (4 sites × 32 participants). In Lab Session 1, 50% of participants chose the same file type (such as photos) for all four websites; in Lab Session 2, 69%. For the at-home component with real-world websites, the total number of possible passwords was 96 (3 sites × 32 participants); 41% of participants used the same file type for all three sites. A chi-square test shows no evidence that the categorizations across all three conditions (two lab and one at-home) are different with statistical significance $(\chi^2(8) = 15.1617, p > 0.05)$.

e) Memorability and recording passwords. Participants had the option of recording password information, including hints; they were asked whether they recorded such information and whether or not they used this on logins. In Session 1, during four trials, 16 participants (50%) recorded password information (primarily hints) at least once; the others did not record any memory aids. In Session 2, 18 (56%) did, at least once, while the remaining 14 did not record anything. For the at-home portion, half recorded password information, and half did not. The most common type of recording was writing password hints on paper. In responses to the background questionnaire, which asked about normal password management behavior, half of the participants indicated that they occasionally wrote passwords down. At the end of each trial, participants rated how likely they thought it was that they would be able to log in successfully a week later, for each website and password, on a scale of 1–7 (with 7 "highly likely"). The ratings were 6.40 (Lab Session 1), 6.30 (Lab Session 2), and 6.24 for the real-world websites in the at-home component.

f) Password reuse. Participants were requested to refrain from reusing password objects, as we wanted to observe how users cope with multiple ObPwd passwords. While we did not block reuse, we detected any occurrences of reuse across sites. 15.6% of passwords were reused (40 out of 256). Participants also self-reported whether they had reused any passwords; 9 of the 13 who were detected as reusing passwords recalled having done so for study sites.

g) Password visibility. Because ObPwd could be used with either manual or automatic pasting of passwords, participants were asked whether they looked at the generated password (only possible with manual paste) or used ObPwd to paste the password into the field without the intermediate step. In Session 1, six participants (19%) looked at the password when it was first created in at least one trial; in Session 2, four (13%) did so. When the passwords were used later to log in, none looked at the password: all logged in using the automatic paste feature.

h) Perceived usability. After the password creation phase, a 7-point rating scale was provided for participants to indicate their perceived level of usability for four factors: ease of thinking of a file to select for the password; ease of locating the file chosen; ease of creating the password using ObPwd; and ease of logging into the website with the chosen password. 1 represented "not at all easy" and 7 "very easy." Table V reports results. The majority of ratings had a median of 7; the lowest was the difficulty of choosing an object file, with a mean of 5.98.

	Lab Session 1		Lab Session 2	
	Mean	Median	Mean	Median
Choose file	5.98	6.0	6.05	6.0
Locate file	6.20	7.0	6.52	7.0
Create password	6.77	7.0	6.83	7.0
Log in	6.73	7.0	6.63	7.0

TABLE V
PERCEIVED USABILITY FOR PASSWORD CREATION TASKS

i) Real-world usage. Participants provided comments about their experiences using ObPwd during the at-home component, on three real-world websites. They indicated the kinds of websites they used ObPwd to create passwords for, which were wide-ranging and included video-hosting sites, blogs, webmail, news and sports sites, and social networks. They were asked to report on any problems encountered; four instances were reported (of a total of 96 sites). The first was a problem caused by changing a password: a participant who changed their Hotmail password had trouble using MSN messenger because they did not know that these two accounts used linked passwords (through Windows Live), and thus passwords for both accounts were changed. A second person had problems with a Flash login screen, which prevented them from right-clicking and launching ObPwd. A third had problems with a video file, stating that the process was slow for creating a password; investigation suggested a problem with an API in specific versions of Firefox, now resolved in later versions. Finally, when one participant used the generated password on a blog site, it failed the requirements for that site. (Possibly that site's password rules required non-alphanumeric characters, not supported by the version of ObPwd used.)

j) Overall user experience. Participants provided feedback about their overall experience with ObPwd through a questionnaire, which included a series of seven-point Likert scales (1="strongly agree," 7="strongly disagree."). Results are listed in Figure 1, which presents boxplots. (In boxplots, the heavy vertical bar is the median, the box shows the two central quartiles and the dashed line shows outer quartiles.)

Participants also described any aspects of ObPwd that they particularly liked or disliked. The most commonly-cited disadvantages were: concerns over being able to use ObPwd on other computers; concerns over losing passwords when files changed; not being able to use ObPwd with other browsers; and time to log in. Advantages cited most frequently were the automatic pasting of the password; the wide selection of passwords; the ease of creating and remembering passwords; and increased perceived security. Some illustrative user comments describing the positive aspects were: "My password was a picture! That was cool."; "It was easy, convenient, and good for security."; "Easy to create, secure, only have to remember a file, not how it's spelled (i.e., uppercase or numbers after)."

k) Personal Verification Questions (PVQs). One potential use for ObPwd is for PVQs, which could allow a user to input her ObPwd password as a difficult-to-guess response for free-format PVQs. Although this use was not explicitly evaluated in this user study, participants were asked about their experiences with PVQs, and to predict their likelihood of using ObPwd for PVQs. All participants knew what PVQs were: 92% had set up PVQs on websites; 84% used them for password resets, and 54% for logins to sites. 52% predicted

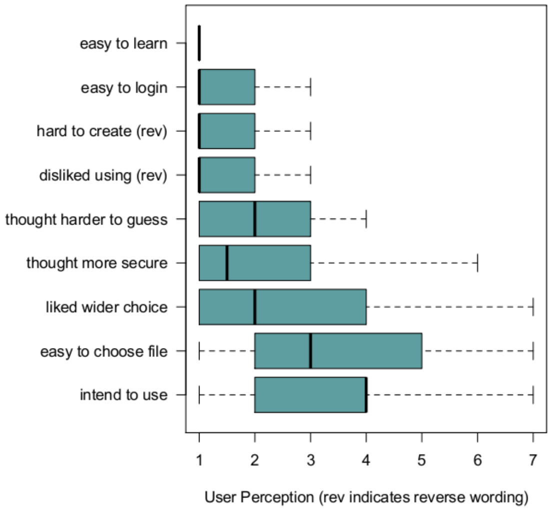

Fig. 1. Perceived usability of ObPwd (Likert-scale)

they would use ObPwd for future PVQs; on a scale from 1–7 (7 = "highly likely" usage of ObPwd), the median rating was 6. 32% of participants were unsure if they would use ObPwd for PVQs, stating that it would depend on factors e.g., the security level of the web account, the availability of ObPwd and password objects, and the ability to create their own PVQ on a specific website. In total, 84% of participants either expected to use ObPwd for PVQs, or could conceive of situations in which ObPwd could be so used.

l) Post-study usage. At the end of the study, we offered to uninstall the ObPwd extension, if participants so chose; of 32 participants, only one accepted this offer. The others kept the extension. Participants could also opt-in to a post-study questionnaire: 26 of 32 (81%) opted-in; questionnaires were emailed to these 26, and 16 responded. Out of these 16, 6 (38%) had used the ObPwd extension after the study, and 7 (44%) had recommended ObPwd to e.g., a colleague or friend.

V. Interpretation of Results

a) Login times and errors. Over 90% of logins were successful within three attempts; by Lab Session 2, over 90% of were successful on the first attempt. However, the mean login time was 19 seconds (by the end of Lab Session 2). To place this time in context, we compare it to Forget et al. [9], who reported login times for 8-character text passwords as well as Persuasive Text Passwords (PTPs).[2] In the PTP variant that provided the optimal combination of security and usability, the login time was 17.1 sec.; in this variant, users must memorize two extra characters in addition to their password. For ordinary 8-character passwords, the mean login time was 11.4 sec. Thus, the ObPwd login task took approximately twice as long as text passwords, and slightly longer than PTPs. A few participants commented on the longer time required, although overall, participants gave high ratings to the ease of logging in with ObPwd (see below).

In multi-account scenarios participants may try one or more wrong passwords before a correct one, thus increasing overall login time. When such multiple password interference is considered, ObPwd login times appear comparable or even slightly better than regular text password and the PassPoints graphical password schemes, as reported by Chiasson et al. [3]; see Table VI. Here *Text*, *PP*, and *OP* represent text password, PassPoints, and ObPwd respectively; *Recall-1* is login in the first test session (Login #2 in Lab Session 1 for ObPwd – see Table II), and *Recall-2* is login to the accounts created in the first session after more than a week (7-10 days in ObPwd, and 12-15 days in Chiasson et al. [3]). Additionally, ObPwd login times include network delays (to the web server). The login success rates for Session 2 of the ObPwd study were much higher ($> 90\%$, see Table III) than those reported [3] for both text passwords and PassPoints (59% and 57% respectively, within three attempts, after 12-15 days). This suggests that ObPwd may provide advantages for password memorability over time; however, the experimental designs of these two studies varied too widely to make their data strictly comparable.

	Mean			Median		
	OP	Text	PP	OP	Text	PP
Recall-1	26.3	29.3	15.1	20.1	14.0	11.8
Recall-2	33.3	42.1	47.0	25.1	26.8	32.6

TABLE VI
Login times (sec) in ObPwd, text, and PassPoints

b) Files chosen and reasons for choice. The most commonly-chosen file type was images. For some participants, the visual characteristics of images allowed them to associate the password file with a specific study website: e.g., one participant stated that for a site with a yellow background, he chose a photo of his girlfriend in a yellow dress. A wider variety of files was used for the real-world sites than for the lab sessions. This may be due to a wider range of websites being available, which gives a greater number of possibilities for locating files suitably associated with those sites. One participant used a journalism course document for a study guide site, and music for an online classified site: "[a] song that reminds me of my mom, who uses [that site]." When identifying the reasons for file choices, memorability was a key issue: the most commonly-cited factor influencing the choice was that it was easy to remember which file was picked, followed by the ability to associate the file with a specific website.

c) Memorability. Participants were almost always able to log in successfully, although they did sometimes try to log in with a password from a different account. We asked them what tricks (if any) they used to keep track of which passwords were used on which sites. Visual cues, as mentioned above, were described. One participant described associating music titles with websites: "...insurance [site] (if you get hurt or die), I linked it with a song, 'Knocking on Heaven's Door'." Some participants used files from the same folder every time; this is facilitated when using the same media type repeatedly, which are often filed together within one folder. Others used letters to link sites and password files, choosing existing filenames that started with the same letter as the website's title. Participants who recorded password information, such as hints on paper, were able to use those hints instead of relying on mental associations to choose the right file.

d) Password reuse. Password reuse was limited,[3] even though each user was dealing with at least 8 ObPwd passwords with a high ($> 90\%$) login success rate; see Figure 2. This may indicate that ObPwd could reduce the effect of multiple password interference. (However, we do not know whether ObPwd will reduce password

[2] The basic idea of PTP is as follows: system-generated characters are inserted at random positions into a user-chosen initial text password; users can accept the proposed password, or request (until satisfied) alternative suggestions.

[3] According to a large-scale study [7], each web password is shared across 3.9 different sites, with 25 accounts on average per user (cf. 1.2 sites/password in our test for 8 accounts).

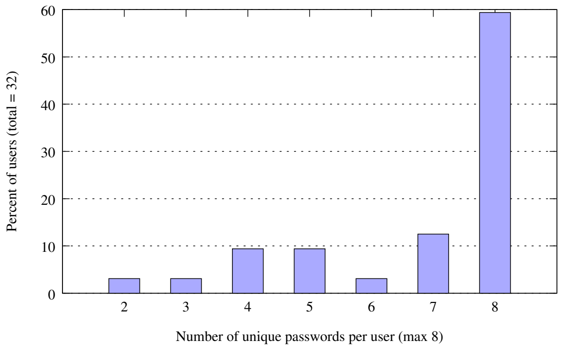

Fig. 2. Password reuse

reuse in practice.) In contrast, most new authentication schemes are not tested under interference. We also found one password hash common among three individuals (but no other two-way collisions). Information gathered from participants suggests that they selected the same sample music installed with the OS; this likelihood may have been increased as such sample music was used as an example password object during demo. Proactive blacklisting by ObPwd, of popular pre-installed files, may thus be prudent; alternatively, the use of ObPwd Variant 1 (Section II) would address such collisions.

e) Password visibility. By the end of the study, all participants used the automatic paste feature for entering passwords, instead of copy-paste, thus did not view the generated password. Participants did not provide reasons for this preference; we speculate that this could be due to convenience, or because the password itself is of limited perceived utility, given that it need not be typed in. This finding is in contrast to that in a password manager study [4], in which participants were frustrated when they were unable to view their passwords; with ObPwd, this option is always available, which may give users a sense of control.

f) Perceived usability and user experience. Participants rated ObPwd highly on the four usability factors measured, including ease of login — despite the relatively long login times. Logins were generally successful within a few attempts, likely contributing to the high rating. There also appears to be a positive *affective* component to ObPwd: affect is the influence of emotional involvement, widely regarded as part of broader strategies in interaction design to promote user engagement and greater cognitive involvement [17] (cf. Passpets [26]). Although the task of finding an appropriate password-object file may take some time, it is not perceived as onerous. Instead, people appear to enjoy interacting with personally-significant objects.

ObPwd was rated highly on ease of learning, and ease of choosing files to use as passwords; participants disagreed with questionnaire assertions that ObPwd made it hard to log in or that they disliked using it. Participants also thought that passwords created with ObPwd were harder to guess, and were more secure than, their usual text passwords, and that they had a wider selection of passwords to choose from. We asked if ObPwd would be adopted for use beyond the study. The Likert-scale answers were unclear on this point, with a mean rating of 3.44: this value tends towards agreement, but not strongly. For clarification, we asked about usage in the post-study questionnaire: 38% used the extension after our study, demonstrating that participants found value in the tool outside of the study environment.

g) Context and limitations. The user study reported is of limited scope, and leaves many issues to explore, e.g., how users would fare in accessing password objects from multiple computers/devices, and the effects of object modifications (see Section VI-B). Providing timing comparisons (Section V, item (a)) with related schemes previously reported in the literature is not ideal (vs. carrying out new independent such studies in precisely comparable conditions), but is nonetheless useful to provide a rough sense of context, e.g., indicating whether login times are comparable or differ wildly. Comparing the usability and security of different authentication alternatives is itself a difficult task with numerous challenges (see Chiasson et al. [1]).

VI. Security Analysis

We provide a practical security analysis of ObPwd, and consider attacks, and other risks.

A. Entropy Estimation

In our discussion of entropy, we consider passwords generated from local unshared objects, to match our user study. For ObPwd in general, if the digital objects used are public web content, the entropy of resulting passwords will largely depend on the predictability of user choices (and the common biases of users), rather than the absolute size of the object space (see also Section VI-B, item (c)). For context related to the difficulty of estimating password entropy, see Weir et al. [25].

Length constraints on password objects. ObPwd uses at most the first $n = 100,000$ bytes from an object, for three reasons: to reduce file or URL read/download time; to reduce hashing time; and to capture sufficient entropy (as some file types may have a large, e.g., few hundred-byte header structure with limited variation). We assume this limit is certainly sufficient to obtain 160 bits (recall that SHA-1 is used) of entropy from most user-chosen file/URL objects. The minimum object size of $m = 160$ bytes is designed to take into consideration user-selected text blocks that may not be rich in entropy. Under the pessimistic assumption that on average, each input byte of a selected text string provides at least one bit of entropy, this value of m provides entropy appropriate to the 160-bit output of SHA-1, to be used as input to Hash2Text.

Entropy metrics. Consider the threat of a brute-force guessing attack. Assume ObPwd passwords are l characters long, each independent and equi-probable from an alphabet of b characters, as consistent with the paragraph above. Then the *Shannon entropy* in bits is given by $H = l \cdot \log_2(b)$, and serves as an upper bound measure of security. Another metric is *min-entropy* [24]; informally, this measures the difficulty of guessing a password for any of a number of (un-targeted) accounts. Under the above assumption, implying ObPwd passwords themselves are equi-probable, their Shannon entropy and min-entropy are equal. A further metric, appropriate for a targeted attack on a selected user account, is *guessing entropy* G [15, 24, 25] (see also Pliam's related discussion of *marginal guesswork* [18]): the expected number of guesses for a correct guess, ordering candidate guesses from most to least probable. Formally, $G = \sum_{i=1}^{K} i \cdot p_i$, where the elements $P_i, 1 \leq i \leq K$, in the password space are indexed in non-increasing order of their probabilities p_i. To execute the optimal strategy assumed by the guessing entropy formula, the attacker must have perfect knowledge of the probability distribution of passwords in the system. For the case of K equi-probable passwords, the formula simplifies easily to $G = (K + 1)/2$ guesses, in any order.

Entropy of ObPwd password (default settings). For the default alphanumeric character set (mixed-case letters plus digits), the SHA-1 hash output of a password object is mapped into the 62-character set. As above, approximating the entropy of user-selected objects to a full 160 bits matching SHA-1, we model each character in the output password as equi-probable and independent — that is, for practical purposes, such ObPwd passwords are random (non-redundant). This assumption is justified under the model in which an attacker, having no access to a password object (or its hash), has no attack better than to guess random strings from the character set. In this case, for an ObPwd password ($l = 12$, default character set), $H = l \cdot \log_2(62) = 71.45$ bits and $G = 2^{70.45}$. This model is reasonable for users choosing personally created objects that remain private, e.g., unshared photos or once-public photos locally altered (but not, e.g., for common public objects, such as shared images and popular soundtracks).

For a rough comparison, by NIST's historical password entropy-estimation heuristics [2], assuming a 94-character alphabet (common printable characters excluding space), a 12-character user-chosen text password has about 24 bits of entropy: 4 bits for the first character, 2 each for the next 7, and 1.5 each for the last 4. If policy requires both uppercase and special characters, this rises to 30 bits; cf. 71.45 above. (Note that the recent empirical analysis by Weir et al. [25], showed that the NIST heuristics do not provide a good model of the level of effort that would be required by an intelligent attacker employing an optimized guessing strategy, i.e., they do not model "guessing entropy" well. For example, passwords with at least seven characters offered about 8.67 bits of guessing entropy in one experiment, compared to NIST estimated 16 bits.) ObPwd entropy in bits increases linearly with the password length (e.g., to 119 bits for length $l = 20$), with virtually no usability impact as users need not memorize the generated passwords. Expanding the ObPwd alphabet set (e.g., from the default 62 to 94 characters) increases entropy further, but many websites require alphanumeric-only characters.

B. Attacks on ObPwd and Risks

Below we discuss attacks on ObPwd and other risks as may arise from a large scale adoption of this scheme.

a) Malware and guessing attacks. As for text passwords, ObPwd passwords/objects are vulnerable if the user platform/device is compromised. However, when password guessing attacks are used to compromise a system (e.g., SSH guessing attacks [19]) or spread to other systems (recall the Morris worm [21]), strong passwords as generated in ObPwd may delay or prevent the compromise.

b) Network-based/man-in-the-middle attacks. If ObPwd is used in regular web login, we strongly recommend that the password objects be stored in local media when passwords are generated on-the-fly (right before login) from public web objects. If a password is re-created from plaintext web content the following attack is possible. An attacker records traffic from the intermediate network (e.g., a wireless access point, web proxy) looking for a user entering a content-hosting site right after or before requesting an authenticating website, thus capturing or narrowing down candidates for the password-generating content. In contrast, when ObPwd is used for encryption/decryption in a user's local media, network-based password objects do not allow access to protected content.

c) Building an attack dictionary. A global password attack dictionary might be built as follows. Assume users who choose to use publicly available objects will do so mostly from highly popular websites. Photos, comments and other information from social networking and photo-sharing sites can be crawled regularly to harvest publicly available password objects. Many search engine providers maintain an updated archive of the public Internet and are also in a vantage point to observe user choices. They may create dictionaries of global password objects, perhaps orders of magnitude larger than current text dictionaries.

To build a custom dictionary on a target user, a user's network provider, proxy sites, certain nodes in anonymous browsing services (Tor exit nodes), or parties who can monitor the user's traffic may gather available data from their personal site, social-networking sites, frequently-visited sites, etc. To access private objects, a dictionary builder may seek to breach private storage (e.g., by malware infection of user machines).

d) Sharing favorite images. Despite recommendations to choose private objects (e.g., personal photos) not publicly shared, users may use in ObPwd precisely the favorite photos they are most likely to share with others. Such risks would be reduced if users posted a modified (e.g., resized) image on the public web, using the original for ObPwd in local media, but such an expectation fails usability goals.

e) Password update, mobility, and lost/stolen media. Password renewal with ObPwd is the same as for current text passwords. Using multiple computers (e.g., home/work PC, laptop) for login requires users have ready and constant access to password objects from multiple platforms. While not an issue for users who regularly carry laptops and mobile devices holding large collections of personal objects, for others, private objects may be used from mobile media such as USB storage. The use of protected online objects (e.g., email text) may also be preferred when mobility is critical. The ObPwd implementation permits roaming also by allowing users to write down the actual (hash output) password, but the usability and user acceptability of this remains unexplored. Losing password objects (e.g., lost media, accidental deletion) is equivalent to forgetting a password; users may resort to existing password recovery/reset mechanisms. Users may favor objects of special significance which they may already keep multiple copies of as backup (photos/videos of favorite trips, weddings, celebrations); this can reduce risks from lost or inaccessible media.

f) Risk of object modifications. ObPwd passwords depend on the first $n = 100,000$ bytes of a selected object. Modifications to content (updating document files, editing image files) preclude re-creating the password, unless the original object or generated password is backed up. Users updating metadata embedded in media files may also be problematic; such metadata is generally stored at the beginning or end of media files. To address this, ObPwd might be modified to drop (e.g., 1000) bytes from each end of such files.

VII. Summary of ObPwd Features and Related Work

Related work. Of countless publications on passwords, here we mention only few schemes designed to strengthen passwords (entropy) or enhance usability. Gibson et al. [10] proposed *Musipass*, an authentication technique relying on the universality of music and human ability to remember/recognize music. Disk encryption software TrueCrypt allows users to optionally use any file from their local system or certain smart cards along with a possibly empty/weak password for generating keys for encryption of disk volumes, instead of deriving encryption keys solely from user-chosen passwords.[4] The currently implemented product does not provide a way for users to write down the resulting encryption keys (for backup). In contrast, the idea behind ObPwd is to facilitate strong passwords from user-chosen content for general use (e.g., for web login passwords, encryption keys, or otherwise).

a) User-chosen strong passwords for ordinary users. Humans are inherently pattern-oriented, and most user-chosen passwords are weak: users are unable to create a string that is high in entropy but memorable over a long period. ObPwd offers the advantages of both system-generated (high-entropy) and user-chosen (easy to

[4] This feature is apparently available since version 4.0 (Nov. 2005); see www.truecrypt.org/docs/keyfiles.

remember) passwords, without their disadvantages (respectively: hard to remember, easy to guess). Users get strong passwords (i.e., which are resistant to dictionary attacks) simply by choosing personally meaningful photos or other digital objects, without being subjected to arbitrarily complex password rules about length, uppercase and special characters.

b) Coarser memory tasks. ObPwd does not require exact password recall; instead, users need to locate their password objects, which involves less detailed memory than regular text or graphical password recall.

c) Engagement promotes user acceptance. The mundane technical task of creating and recalling text passwords is replaced by selecting and recalling objects (e.g., personal photos) that are more both familiar and reportedly satisfying to users. This provides a positive affective experience, leading to strong user engagement with ObPwd.

d) Secure password sharing. ObPwd may enable better password sharing than text schemes without sacrificing confidentiality to third parties — e.g., if two users pre-share digital photos (say through personal media), one can choose a specific image as the password object, and send the other a hint or description (e.g., "our whitewater kayaking photo") over public media or email. An eavesdropper seeing the hint cannot generate the shared password without access to the object itself, assuming the hint is not an obvious link to a publicly-accessible object. This form of *user-friendly codebook*, using meaningful objects, has advantages over sharing a list of randomly generated secret keys.

e) ObPwd as strong graphical passwords. ObPwd provides a middle ground between text and image-based password schemes, allowing use of images while retaining simple advantages of text passwords (low cost, no system-side changes, written records). In contrast, most graphical password schemes [22] use system-assigned images/random art, and require server side changes. Most offer a large password space in theory, but due to bias in human selection the password space used in practice is much smaller [5, 23]. Even if thousands of users choose their own picture of the Eiffel Tower as their password object, ObPwd will generate unique strong passwords as long as they do not share identical photo objects.

f) No changes to server or password interfaces, including PVQs. Deployment barriers are low, requiring no system-side changes at enrollment or login, nor to client-side software interfaces as alphanumeric character strings are produced. ObPwd passwords can be used as answers to PVQs, e.g., in answer to "What is your mother's maiden name?", use a high-entropy ObPwd password generated from a memorable private image of your mother.

VIII. Concluding Remarks

Choosing multiple, long-term memorable, high-entropy secrets is not a basic human capability. Current password generation techniques and password-restricting rules have largely failed to yield strong passwords. Creating passwords from personally meaningful/memorable digital objects offers a user-friendly alternative to more complex password rules.

While our user study provided insight into the types of local files chosen as ObPwd objects, a very large-scale field study is necessary to allow empirically-based quantification of the guessability of the resulting passwords, to search for usage patterns, and to develop best practice guidelines recommending or excluding certain classes of objects. As discussed herein, ObPwd passwords can heuristically be modeled as random strings, thereby providing security far outweighing conventional text passwords, but only under assumption that attackers do not have access to the password objects, and are unable to predict the use of popular objects that are publicly available.

Our hybrid user study exhibits strong ecological validity, including, beyond the usual return-to-lab sessions, a field component wherein participants used ObPwd passwords to access real-world web sites of their choice. The user study showed positive results: acceptable login times, very good login success rates, and extraordinarily positive user perception of the experience. Participants' comments showing a strong affective experience with ObPwd indicate a likelihood of both better engagement and memory. The user study and analysis suggest a novel combination in password authentication: the positive affective aspects associated with user-choice (plus acceptable performance), without the negative of password guessability typically accompanying user-choice.

Acknowledgments. The second author is supported by an NSERC PDF. The third author is Canada Research Chair in Authentication and Computer Security, and acknowledges an NSERC Discovery Grant. We also acknowledge partial funding by NSERC ISSNet.

References

[1] R. Biddle, S. Chiasson, and P. van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (to appear)*, 2011.

[2] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic authentication guidelines. NIST SP 800-63, Apr. 2006.

[3] S. Chiasson, A. Forget, E. Stobert, P. van Oorschot, and R. Biddle. Multiple password interference in text and click-based graphical passwords. In *ACM CCS'09*, Chicago, IL, USA, Nov. 2009.

[4] S. Chiasson, P. van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *USENIX Security*, Vancouver, Canada, Aug. 2006.

[5] D. Davis, F. Monrose, and M. K. Reiter. On user choice in graphical password schemes. In *USENIX Security*, San Diego, CA, USA, Aug. 2004.

[6] 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*, Boston, MA, USA, Apr. 2009.

[7] D. Florêncio and C. Herley. A large-scale study of web password habits. In *WWW'07*, Banff, Canada.

[8] D. Florêncio, C. Herley, and B. Coskun. Do strong web passwords accomplish anything? In *USENIX HotSec'07*.

[9] A. Forget, S. Chiasson, P. van Oorschot, and R. Biddle. Improving text passwords through persuasion. In *SOUPS'08*, Pittsburgh, PA, USA.

[10] M. Gibson, K. Renaud, M. Conrad, and C. Maple. Musipass: Authenticating me softly with "my" song. In *NSPW'09*, Oxford, UK, Sept. 2009.

[11] N. Haller. The S/KEY one-time password system. In *NDSS'94*, San Diego, CA, USA, Feb. 1994.

[12] C. Herley. So long, and no thanks for the externalities: The rational rejection of security advice by users. In *NSPW'09*, Oxford, UK, Sept. 2009.

[13] M. Mannan and P. van Oorschot. Digital objects as passwords. In *USENIX HotSec'08*, San Jose, CA, USA.

[14] M. Mannan, T. Whalen, R. Biddle, and P. van Oorschot. The usable security of passwords based on digital objects: From design and analysis to user study, 2010. Tech. report (TR-10-02), Comp. Sci., Carleton University.

[15] J. Massey. Guessing and entropy. In *IEEE Symposium on Information Theory (ISIT)*, Trondheim, Norway, 1994.

[16] L. Peterson and M. Peterson. Short-term retention of individiual verbal items. *Journal of Experimental Psychology*, 58, 1959.

[17] R. W. Picard. Affective computing. Technical report, 1995. MIT Media Lab, Perceptual Computing Group.

[18] J. O. Pliam. On the incomparability of entropy and marginal guesswork in brute-force attacks. In *Indocrypt'00*.

[19] D. Ramsbrock, R. Berthier, and M. Cukier. Profiling attacker behavior following SSH compromises. In *IEEE/IFIP Dependable Systems and Networks (DSN'07)*, Edinburgh, UK, June 2007.

[20] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In *USENIX Security*, Baltimore, MD, USA, 2005.

[21] E. H. Spafford. The Internet worm: Crisis and aftermath. *Communications of the ACM*, 32(6), 1989.

[22] X. Suo and Y. Zhu. Graphical passwords: A survey. In *ACSAC'05*, Tucson, AZ, USA, Dec. 2005.

[23] P. van Oorschot and J. Thorpe. On predictive models and user-drawn graphical passwords. *ACM TISSEC*, 10(4), Jan. 2008.

[24] E. R. Verheul. Selecting secure passwords. In *CT-RSA*, San Francisco, CA, USA, Feb. 2007.

[25] M. Weir, S. Aggarwal, M. Collins, and H. Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *ACM CCS'10*, Chicago, IL, USA, Oct. 2010.

[26] K.-P. Yee and K. Sitaker. Passpet: Convenient password management and phishing protection. In *SOUPS'06*.

Robert Biddle is a Professor in the School of Computer Science at Carleton University in Ottawa, Canada. His research interests are in "the secret life of software": how software can work in rich and subtle ways with human expression and human behaviour. His current work is on human-computer interaction and computer security, and on software design and development for new media.

Mohammad Mannan is an NSERC post-doctoral fellow at the University of Toronto (also supported by NSERC ISSNet). He has a Ph.D. in Computer Science from Carleton University (2009) in the area of Internet authentication and usable security. He is currently working on securing applications and operating systems using virtualization and hardware support.

Paul C. van Oorschot is a Professor of Computer Science at Carleton University in Ottawa, where he is Canada Research Chair in Authentication and Computer Security. He was Program Chair of USENIX Security 2008, Program co-Chair of NDSS 2001 and 2002, and co-author of the Handbook of Applied Cryptography (1996). He is on the editorial board of ACM TISSEC and IEEE TDSC. His current research interests include authentication and identity management, security and usability, software security, and computer security.

Tara Whalen is a Research Associate in the School of Computer Science at Carleton University and an IT Research Analyst at the Office of the Privacy Commissioner of Canada. Her research interests include privacy, the human factors of security, and the social implications of technology.