Exploration and Field Study of a Browser-based Password Manager using Icon-based Passwords*

Kemal Bicakci[1] Nart Bedin Atalay[2] Mustafa Yuceel[1] Paul C. van Oorschot[3]

[1] TOBB University of Economics and Technology, Turkey
[2] Selcuk University, Turkey
[3] School of Computer Science, Carleton University, Canada

Abstract. We carry out a hybrid lab and field study of a password manager program, and report on usability and security. Our study explores iPMAN, a browser-based password manager that in addition uses a graphical password scheme for the master password. We present our findings as a set of observations and insights expected to be of interest both to those exploring password managers, and graphical passwords. Motivated by our findings, but also of independent interest, we also present a new salt generation method using blind signatures, to protect against offline attacks, decreasing user inconvenience by generating salt significantly faster than earlier work (Halderman et al. 2005).

Keywords: password managers, graphical passwords, field study, security and usability, salt

1 Introduction

Despite continuing status as the default method for Internet authentication, passwords have well known deficiencies. They are often highly predictable, not well protected, and have many usability issues. Seriously complicating this, users must remember not just one, but multitudes of passwords. Given the growing number of web sites users have passwords for [10], it is almost impossible to avoid the poor practice of re-using a password across several accounts, with obvious negative security implications [5, p.3]. On the other hand, using distinct passwords increases the occurrence of forgetting, or mis-matching passwords across sites.

Password managers offer to ease usability problems related to a multiplicity of passwords, by reducing the memory burden to a single master password. They may be implemented as standalone programs or extensions to web browsers. The latter is more convenient for Internet applications, relieving users from the task of starting up a separate program, and providing protection against phishing attacks [31].

We carry out a hybrid lab and field study to explore the usability of a browser-based password manager, including user perception of acceptability. While many password managers exist (see §2), their usability has received surprisingly little attention. A few preliminary lab studies have considered usability [31, 8, 4], but to our knowledge, no field study of password manager programs has been reported in the literature,[4] leaving a gap in understanding usability and security issues in natural environments—which is amplified by the challenge of emulating, with high ecological validity, factors related to password managers, especially those involving *changes in user behavior*. For example, users may access all accounts by entering a master password to the manager program, rather than site-specific passwords; in actual practice, will they choose to do so?

Our field study is further distinguished by exploring graphical passwords for the master password of a password manager. A motivating factor is their claim to offer several advantages over text passwords [19] but also of special interest in our work, they may help to reduce the likelihood of inducing insecure behavior [4, 8]. The graphical scheme we use is GPI [3], wherein user passwords involve recognizing a sequence of icons from a large displayed set. Around the password interface of GPI, we design and implement a password manager program called *iPMAN* (**i**con-based **P**assword **MAN**ager).

Beyond reporting on the hybrid study of iPMAN, we present our observations and insights from an evaluation of the resulting data. Some lessons generalize to other password manager tools, while others apply

* January 21, 2011 for pre-proceedings of RLCPS 2011, co-located with FC'11 (revision to appear, Springer LNCS).
[4] An informal test for PassPet reported preliminary information about results [36].

to stand-alone graphical passwords. The study also provides additional insight on the GPI scheme itself. The selection of weak (graphical) master passwords by many participants motivated a further contribution to protect against password guessing attacks, of independent interest beyond password managers and graphical passwords: a new salt generation method which avoids the long user wait time of earlier work [17].

2 Background and Related Work

The numerous graphical password (gp) schemes proposed in recent years can be classified into three types according to the memory task involved: recall-based schemes (e.g., DAS [19]), cued-recall schemes (e.g., PassPoints [35]), and recognition-based schemes (e.g., PassFaces/Face [9]).

It is known from the cognitive psychology literature that recognition memory—being able to recognize something previously encountered—is easier and longer-lasting than recall-based memory [23]. Numerous recognition-based graphical password schemes leveraging this human ability have been developed and tested. Users are given a set of pictures, and must recognize and select a subset of them as a password. Most recognition-based gp schemes explored to date have been implemented and tested with relatively small password spaces, e.g., comparable to 4-digit PINs. In general these schemes can be parameterized to yield larger spaces (e.g., using more faces per screen in PassFaces, and/or more than 4 rounds of screens), but usability has not been tested under those circumstances.

GPI and GPIS [3] are recognition-based gp schemes comparable in many ways to PassPoints, including in theoretical password space size, for reasonable parameterizations of each. In GPI and GPIS a password is an ordered sequence of icons (mini-pictures) which represent objects belonging to certain categories. The categories and objects are based on a category norm study by van Overschelde et al. [28]. Icons in a common category are grouped and presented in a common row to ease memorizing the password by forming associations between the password, icons and the categories. The idea is that category structures, an organization familiar to the human brain, will enhance memory performance. In GPI, users self-select a portfolio of password icons; in GPIS their portfolio is initially system-assigned and can be changed later. A lab study [3] found GPI less vulnerable to hot-spot issues [34] than Passpoints.

There are two common password manager approaches [16]. The *password wallet approach* uses a master password to encrypt a file of site-specific passwords, stored in encrypted form and decrypted as required. Numerous manager programs implement this approach, including Apple's Keychain [24], Password Safe [29], and the Firefox browser's built-in password manager; some are implemented as browser extensions, and may support advanced features like automatic form filling, e.g., LastPass [21], 1Password [1]. In the *hashing approach*, which iPMAN takes, the master password is combined with site-specific information to generate site-specific passwords. These include early systems [14, 2] and browser extension implementations such as PasswordMaker [20], Password Composer [30], PwdHash [31], Password Multiplier [17] and PassPet [36].

Single sign-on solutions (e.g., the OpenID initiative [27, 32]) also aim to mitigate the password fatigue due to the effort required to remember large numbers of passwords. Our anecdotal observation is that only built-in password managers in web browsers are widely used and other password managers appear to be of considerable interest to a minority of users (for personal use), whereas single sign-on solutions seem to be used (and marketed) more by those with enterprise goals. As such, password managers are more a "grassroots" movement, and single sign-on systems more a corporate movement.

A lab study by Chiasson et al. [8] of implementations made publicly available by the original designers of PwdHash [31] and Password Multiplier [17] found major usability problems, and noted the danger of password manager interfaces inducing mental models resulting in security exposures—e.g., users unable to properly activate software may reveal their master password to a visited site.

In a lab study involving a browser-based password manager GPEX, Bicakci et al. [4] found that graphical passwords had better usability characteristics than text passwords. The PassPoints-based user interface involved clicking cells demarked by a visual grid. Lab study results indicated that user performance for common tasks (e.g., login, migrate password) was better than for PwdHash. In contrast to PwdHash, improper usage does not cause security exposures in GPEX as the cued-recall aspect of a GPEX master password precludes it from being submitted to the wrong site. Another study using a graphical password as a password manager is by Govindarajulu and Madhvanath [15].

3 iPMAN Password Manager Implementation

In iPMAN, the hashing approach is used with GPI (see §2) as the password entry interface, thus precluding password reuse across sites.[5] No server-side changes are needed to use iPMAN. A user first double clicks on the password field to activate a dialog box to display a panel of icons (see Fig. 1(a)) and then clicks individual icons to select an ordered set of icons to create their master password. The placement of icons is static, i.e., identical for all users. After the "Enter Password" button is clicked, the panel disappears, and the browser extension converts the master password to a site-specific character-based password, which is automatically inserted in the password field. The iPMAN master password is not stored.

(a) User Interface of iPMAN

0-1 Times
2-5 Times
More than 5 Times

1	1	1	2	1	1	2	1	1	1
1	7	1	3	2	2	11	1	1	3
0	0	0	2	0	0	2	0	0	0
0	0	0	3	1	0	2	0	1	1
0	0	0	0	0	1	1	0	0	0
1	1	1	3	0	0	0	1	1	1
1	2	2	3	3	1	0	1	0	1
0	0	0	0	0	2	1	0	0	1
0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	2	0	0	0
0	0	0	0	1	2	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	1	2	0	1	0	0	0	2	0
0	0	0	0	1	1	0	1	0	1

(b) Frequency of icon selection as part of master passwords in field study.

Fig. 1: iPMAN Interface and Frequency of Icon Selection in Field Study

The above procedure is the same for both password creation and subsequent password entry; iPMAN has no special session with a different interface to create the master password. This feature is more a technical requirement than a design choice: a password manager implemented as browser extension cannot distinguish first-time users from existing users, and is thus unable to automatically present a different interface for master password creation. This is also why GPI is used instead of schemes which suggest stronger passwords, such as GPIS [3] above, or the cued-recall PCCP [7] which aims to address user choice issues common in other schemes by persuading users to select more random passwords.

We implemented iPMAN as a Firefox extension, testing on Firefox 3.5 and 3.6.[6] Each interface panel row has 10 icons (each 32 x 32 pixels) belonging to a single category from 15 system-configured categories, for a

[5] By contrast, in the password wallet approach, if site-specific passwords are user-chosen, password reuse may occur even if users are encouraged to choose unique passwords for different sites.

[6] Version 2.1 of iPMAN is available at `http://bicakci.etu.edu.tr/iPMANV2.1.xpi`

total of 150 icons and panel size 320 x 480 pixels. The user interface (Fig. 1(a)) includes two other buttons and a check box. The user may press the "Reset" button to start over again, if they wish to change the icons selected. The "Close" button closes the dialogue box without sending the password to the system. By checking the check box (default: unchecked), the user may elect to have the system display the site-specific password (see Fig. 2) before it is inserted into the password field. This functionality is motivated by an earlier study [8] indicating that some users wish to know their "actual" passwords.

The cardinality of iPMAN's theoretical password space is $P(Y, X)$. P denotes permutation, Y the number of panel icons in total, and X the number of constituent icons in passwords. $P(150, 6) \approx 2^{43}$ matches the common configuration of the cued-recall scheme PassPoints [35], and is significantly larger than the $9^4 = 2^{12.7}$ possible passwords for common implementations of recognition-based schemes like PassFaces [9].

Site-specific passwords $SP = post\_process(H(URL\_info||master\_pswd))$ are generated by hashing part of the site's URL with the master password (the latter encoded as indices of clicked icons); e.g., URL_info may simply be `google.com` for `https://www.google.com/accounts/ServiceLogin?...` This generation method suffices unless a password must be identical for two or more sites having different domain names.

Fig. 2: Dialog box showing site-specific password.

Contradictory password rules on different sites may preclude a single password format being suitable for all sites. This (and special URLs as above) can be addressed by using a password policy file [31] for password hash post-processing to conform to site policies. For visited sites not listed in the policy file, default post-processing is performed. We introduce a central repository[7] shared between all users to relieve users from the burden of manually updating policy files; iPMAN clients automatically check for an update and retrieve the latest version of a policy file. To avoid security problems, care must be taken to ensure the policy file cannot be controlled by attackers [31]. Caching the most recent such file allows continued operation should the online connection be temporarily lost.

4 User Study of Password Manager

We conducted a hybrid user study which includes lab and field study components to evaluate iPMAN usability (efficiency, effectiveness, acceptability) and security. While the iPMAN password manager differs from others with respect to its user interface, it also has common features and characteristics, and involves similar user behavior issues to other password managers that use hashing to generate site-specific passwords—e.g., the functionality provided by using a master password, using this one password on different accounts, and attitudes towards password security. Tasks common to such managers include converting existing passwords, remote login, etc. Our study thus provides insight about the usability of both password managers in general, and of graphical passwords as the interface for password managers.

We investigated effects of password rules on the usability of iPMAN. The strength of iPMAN passwords decreases if users choose fewer icons within passwords. A long-standing strategy to reduce weak passwords is password rules. We imposed a password length rule of exactly 6 icons on half the participants; the others chose unrestricted passwords. We compared usability metrics of the two groups to investigate the effects of password rules on login time and login success rate. (In an earlier field study, Tao and Adams [33] compared success rates for creating a new graphical password under various password policies for Pass-Go.)

4.1 Methodology

Our small study, approved by ethics committee of Middle East Technical University, involved 20 students (11 male, 9 female) of average age 21.9 years. None had participated in a password usability study before. Participation was voluntary. Participants could leave the study at any time. At the end of the study, a camera was given to one randomly selected participant. To investigate the effect of a password length rule on usability and security, participants were randomly split into two groups: 10 could choose their own password lengths (Free Choice Group), the other 10 (Six Icons Groups) were required to choose exactly six icons.

[7] See `http://myuceel.etu.edu.tr/rules.xml`

Procedure. To begin, we invited participants individually to a lab session for a questionnaire on Internet and password usage. Onto each participant's computer we installed a version of iPMAN that included a logging function to collect data necessary for usability analyses. We informed participants that the software would record information on passwords to their computer; that there were no online data transmissions to remote machines; and that user data including their passwords would be collected at the end of the study by the experimenter. We provided participants detailed written instructions about usage of the system similar to the explanation in the previous section, and answered any questions regarding using iPMAN.[8] We set up a server site allowing users to generate site-specific passwords from computers missing the iPMAN extension (similar to the PwdHash remote-login page [31] but implementing the iPMAN interface as a web application). We showed participants how remote-login works and told them they could use that site/page when desired to login from secondary machines. We provided the site URL in the instruction sheet. The experimenter gave his/her phone number to participants, who were free to ask for help at any time during the study.

Each participant chose their own master password and set it in the lab. There was no practice session. We asked them to use iPMAN on all sites they use. We requested they not use their browser's password auto-complete function during the field study. (Note: any such auto-complete use would not impact our statistics, such as login success rate, as our data collection occurred only for logins in which participants actively click on the icons.) They used iPMAN for 43.6 days on average. After this time, we invited them individually to our lab to collect the passwords and usability data logs on their computer. Users were notified again that their iPMAN passwords were collected but they were not asked to take a particular action. They were free for transition back to normal passwords or to continue using iPMAN with or without changing the master password. They were given a second questionnaire, and a short oral interview on the usability of iPMAN. Finally, 20-25 days later, each was invited to a surprise memory test for their master password.

4.2 Results

Questionnaire on Internet and Password Usage. All participants reported that they use the Internet every day except on vacation. Seventeen reported using Firefox as a browser. In self-rated computer skills, 13 (65%) rated themselves as average users, 4 (20%) as above average, and 3 (15%) as expert. 85% reported using the same regular text password on more than one site. 80% also indicated they were concerned about the security of their password. The two most common criteria cited in password choice were ease for remembering, and difficulty of being guessed by others. The majority of participants' usual text passwords were 8-9 characters and included only mixed case alphanumerics. These results were similar to a previous study [8].

User Support. During the study, 4 of 20 participants called for help. One was unable to change an existing password, because the website rejected passwords with special characters. We updated the password policy file to fix the problem. The other three reported that icons occasionally failed to appear on the panel. We found the problem was Java-related and suggested that participants address this problem by restarting Firefox. We later modified iPMAN to no longer depend on the Java run time environment. No participants called for a help about how to generate, change or update site-specific passwords with iPMAN.

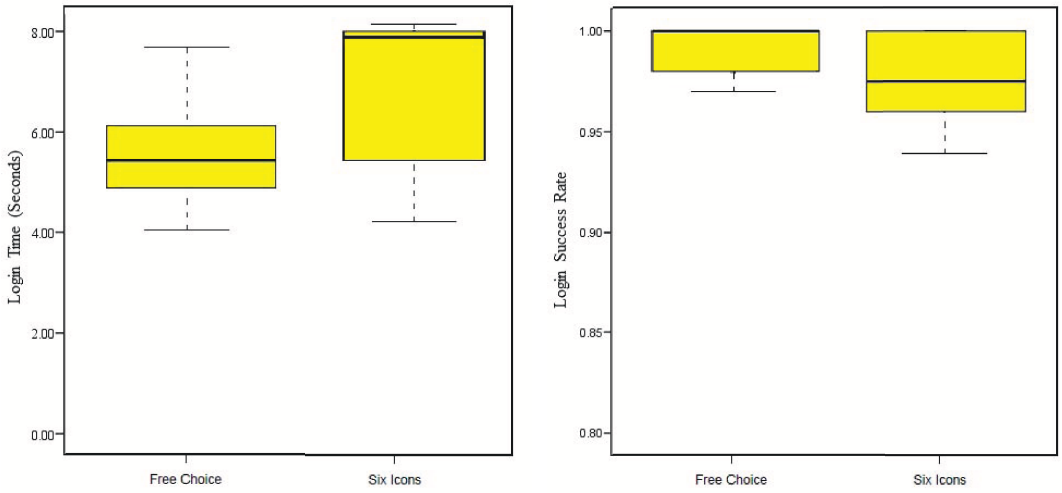
Effectiveness and Efficiency. The average length of master password was 3.50 icons (min=2, max=6, std.dev=1.08) for the free choice group, and fixed at 6 for the other group. One participant changed their master password during the study. For the following statistical analyses, the level of significance used is 0.05.

During the study participants made a total of 1197 login attempts with iPMAN. The per-participant average was overall 59.6 (std.dev=29.3, min=31, max=128), for the free choice group 56.9 (std.dev=29.2), and for the six icons group 62.4 (std.dev=30.8). The difference between groups was not significant [t(18)=-0.41, n.s.], suggesting that both groups used iPMAN equally often. On average, participants logged in to 2.35 different sites (stdev=0.48, min=2, max=3) with iPMAN, less than our expectation. Despite the instruction to use iPMAN for all sites, participants preferred to use it for popular sites like Gmail and Facebook but

[8] Providing information beyond the written instruction was part of our ecological design, and might be expected in an enterprise setting. Our objective was not to assess learning performance itself.

not pages visited less frequently. We view this as a finding of interest (see later discussion), rather than a failure to understand instructions.

Efficiency was measured based on the time taken by users to enter their master password. For each participant average time for correct password entry was calculated. Participants entered their iPMAN password in 6.31s on average (std.dev=1.6): 5.80s for the free choice group (std.dev=1.57), and 6.81s for the six icons group (std.dev=1.51). See Fig.3(a). The difference between the groups was not significant [t(18)=-1.46, n.s.]. As is well-known, failing to find a statistically significant difference between groups does not reflect identity. Our result may also reflect a small difference or low statistical power.



(a) Time to log in with iPMAN (b) Login success rate with iPMAN

Fig. 3: Login time and success rate across different password rule groups.

Effectiveness was measured by number of correct master password entries. Of 1197 total password entries, 1178 succeeded on first attempt (98.4%). For the 19 incorrect password entries, in 11 participants clicked either immediately to the right or left of the correct icon; 5 accidentally clicked the same icon twice; in one the user confused the order of icons. Two of the 19 incorrect entries were consecutive. The login success rate was 98.3% on average (std.dev=1.9%, min=94%, max=100%). There was a significant difference in login success rate between free choice (mean=99.2%, std.dev=1.3%) and six icons group (mean=97.4%, std.dev=2.1%) [t(18)= 2.42, p<.05]. See Fig.3(b). We note that the login success rate was very high in both groups, yielding very low standard deviations, which may affect the significance test. All 19 incorrect logins were made by 10 participants; the other 10 entered their entire master password correctly each time. On average, participants who did not make a mistake logged in 56.6 times (std.dev=28.94, min=31, max=127); those who made a mistake logged in on average 62.7 times (std.dev=30.98, min=34, max=128). There was no correlation between login success rate and number of logins (r(20)=0.084, n.s.).

Questionnaire and Interview. At the end of the study, all participants completed a questionnaire (see Appendix 1). Seven questions were borrowed from an earlier lab-based study on password managers [8], some of which were modified to suit our present study. Fig.4 summarizes responses on a Likert scale (1=strongly disagree; 5=strongly agree). Aggregate scores were: Perceived Security (mean 4.78, std.dev 0.44); Ease of Use (mean 4.45, std.dev 0.52); Perceived Necessity (mean 2.82, std.dev 0.61). Not Giving Control score was: mean 4.50, std.dev 0.76; strongly agree here means participants were fully comfortable with their ability to record, if desired, the resulting site-specific passwords. The free choice and six icons groups did not differ in any of these measures [t(18)<1.18, n.s.].

The questionnaire contained additional Yes/No questions. We summarize responses as follows: 19 participants (95%) reported no difficulty in remembering their password; 14 (70%) reported benefiting from icon categories for remembering their password. One reported writing the master password on paper, but also that he/she never looked at it.

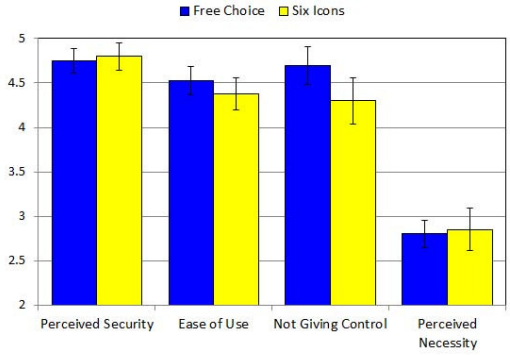


Fig. 4: Response means for question groups. 1=strongly disagree, 5=strongly agree. Bars show std errors.

We also asked participants their opinions about the remote login page. Four of 20 participants reported that they had used the remote login feature. None found it user-friendly. Participants reported difficulties remembering the web site URL. There were Java-related problems with the page. One participant reported being unable to generate their password with the remote login page. We asked participants if they'd like to continue to use iPMAN after the study. None reported that they would definitely use it; 15 (75%) reported they had not decided yet; 5 (25%) reported that they would definitely not use it. We asked these latter 5 the reason for not using iPMAN furthermore; 4 reported that their original passwords were secure enough.

Memory Persistence. To explore the persistence of memory for master passwords, 20-25 days after the field study ended we invited participants to the lab for a surprise test. All 20 accepted and participated in a test shortly after. We asked each to click their master password. All remembered their password correctly in their first trial although 19 of 20 reported no longer using their iPMAN master password after the field study. The remaining one had continued to use iPMAN.

5 Discussion of Results and Summary Observations

Having collected and reported the user study results, here we analyze and discuss them and their implications for the design of password managers, for conducting user studies on such password manager programs, and also for the specific icon-based graphical password interface of iPMAN. Introducing users to both a new graphical password interface and a password manager makes it hard to separate the effects of each individually. Nonetheless, the results give us the following intuitions packaged as a set of summary observations. We defer security-specific discussion to §6.

Users resist migrating their existing passwords. A large-scale study conducted in 2007 [10] indicates Internet users have about 25 accounts requiring passwords. The average number of sites our participants logged into with iPMAN was low; they did not follow instructions to migrate all of their passwords to it, and preferred to change their passwords only for frequently visited sites. The convenience of being able to login to multiple sites by entering only a single password is more apparent when so used on larger numbers of accounts. But in iPMAN and similar password managers including those having a text password interface this requires that users have migrated all or most of their old passwords to new ones generated from the master password. Our field study suggests that the short-term hurdle cost of perhaps a few minutes on each site to migrate passwords dominates the perceived longer-term benefit in the minds of users. Another plausible explanation of user resistance is that the "path-of-least-resistance" [12] works against migrating all or most web account passwords. As a result, the impact of the password manager on the web experience of participants was minimal during the field study. This is one reason we believe the perceived necessity score was low and why users were reluctant to continue using iPMAN after the study although they reported that they found iPMAN easy-to-use and secure (in their perception). In the literature, we are not aware of any discussion of the prohibitive nature of this initial one-time cost of migrating passwords. We conjecture that if researchers fail to find an innovative way to reduce the user pain associated with migrating passwords,

then the wallet approach to password managers (which avoids migrating passwords) will maintain a major usability advantage over the hashing approach.[9]

There is a trade-off between transportability and usability. In many existing designs, the available tuning knobs offer a trade-off between security and usability. In password management, there is a third dimension—transportability—which we define as the ability and ease to login from secondary devices other than a user's primary computer. Transportability may be regarded just as one aspect of usability. To explain what we mean by *trade-off*, we first revisit the password wallet approach, which as noted above, has the usability advantage that users can start using it without needing to migrate passwords. On the other hand, it suffers an important deficiency: the master password is useless for login from a secondary machine unless the password wallet is moved to that machine.

In theory, transportability requirements related to password managers can be satisfied by the hashing approach—but not every browser comes with pre-installed password hashing functionality. Thus to support transportability in our field study, we adopted the remote login page method [31]. Our study confirmed previous work [31, 8] illustrating usability challenges of the remote login site idea (installing the manager program on the remote machine also raises issues [8]). We thus lean towards the belief that the password hashing approach can address transportability only if the manager is integrated in (all major) browsers rather than implemented as an extension. Otherwise, manually entering site passwords continues to be a more transportable choice (though less usable in other aspects) than using either class of password managers.

Usability Comparison: Master vs. Regular Password. An important advantage of a master password comes from users repeatedly entering the same password time and again for different sites—repetition and habit reinforce memory and usability. This advantage is illustrated by comparing the usability results of our field study on iPMAN with those from the lab study by Bicakci et al. [3] on the stand-alone version of GPI.

Recall that the user interface in iPMAN for master passwords is identical to GPI. The lab study of GPI involved two sessions. First, participants generated passwords with six icons on a GPI interface, then one week later they were invited to a session to login with their GPI passwords; 4 out of 23 forgot their GPI passwords. In contrast in the field study, in a surprise memory test performed 20-25 days after it ended, all participants still remembered their passwords.[10] The difference between the memory performances was significant $[\chi^2(1) = 3.835, p = 0.052]$. We conjecture that the difference is due to participants' repeated rehearsals of their master password while using iPMAN, reinforcing a strong memory of it. In the lab study, the time to enter the correctly remembered password for GPI was 17.5 seconds on average (stdev = 22.30), substantially longer than the average time to login with iPMAN presented in Fig.3(a) $[t(41) = 2.202, p < .05]$. In the final week of the field study, participants entered their iPMAN passwords around 0.5s (on average) faster than the average login time of 6.31s, which also shows that participant login times improved as their experience with the system increased.

It is reasonable to also expect improvements for passwords in regular use as users become familiar with them. For instance, in a field study [33] of the Pass-Go graphical password scheme, login success rates were low in the first three weeks but became stable at around 90% after week 7. Not contradicting the results of previous work, the results of our field study suggest that by habitual use of a single master password across different sites user performance may reach higher levels than when several distinct passwords are used.

Impact of Password Rules on Usability. As another usability result, we observe that forcing users to select six icons did impact the usability of iPMAN as follows. There was no difference between the free choice and six icons groups with respect to the login time. But there was a statistically significant difference with respect to the login success rate. On the other hand, login success rate was high in both groups (99.2% and

[9] Our observations differ substantially from those of Yee et al. [36, §7].

[10] While it is not always appropriate to compare lab and field study results, here the finding that success rates in the lab study were weaker despite its shorter intervening period, appear to only strengthen the observation. Regarding demographics: most participants in both studies were university students with similar web use profiles.

97.4% for free choice and six icons group respectively), and the difference is small (1.8%). We view this as an acceptable usability impact related to the six-icon password rule, albeit lacking a scientific metric.

Comparison of Survey Results. Earlier, we noted the limited number of usability studies on password managers. Using the usability criteria from one exception, Chiasson et al. [8], we put the same survey questions (with minimal necessary changes) to our participants. Our field study results reveal that iPMAN scores well on ease-of-use and perceived security scores which are higher than the scores reported [8] for PwdHash [31] and Password Multiplier [17]. Our survey results also confirm that users are more comfortable if they can learn their site-specific passwords. The only low score for iPMAN is on perceived necessity, which is similarly low for other managers [8].

Regarding possible reasons for the low score on perceived necessity in our study, aside from security not being the primary goal of most end users, we conjecture that users are trapped in a vicious downward spiral, in which the small number of web sites the password manager was used for is both cause and effect of low perceived necessity. Our hypothesis, which may be of interest to test in a separate study, is that if we could break the downward spiral and persuade users so that the percentage of a user's passwords migrated to the manager program is increased, the perceived necessity score would also increase.

We conjecture there is a threshold for this migration percentage that, once passed, removes the path-of-least-resistance [12] barrier in favor of continuing with the manager vs. turning back to old passwords.

Limitations. A notable limitation of our study is the small number of users: 20 participants is insufficient, especially for a comprehensive security analysis of user-chosen passwords.

While we highlighted that migration may pose a big challenge to adopting password managers using the hashing approach, this effect may have been amplified by the study design as the users not only had to migrate passwords, but also might feel it necessary to change passwords again after the study as the experimenters gained access to their passwords. It is also possible that part of the reluctance to adopt iPMAN, especially for sensitive accounts, may have been due to a concern about such access to passwords.

The study design involved users adopting both a password manager and a novel graphical password scheme. A design introducing only one of these conditions would allow more convincing conclusions. A future study could compare different user interfaces (e.g., graphical versus text) of password managers.

80% of participants indicated concern about the security of their passwords. Such a concern does not automatically imply security benefits of password managers (e.g., avoidance of password reuse) are understood and appreciated (indeed, 85% also reported reusing passwords). On the other hand, password managers also have usability advantages which may be appreciated more, especially among users who regularly forget passwords. Our study did not ask our participants how often they forgot their passwords. A future study could compare the perceived necessity score and other usability statistics between users who think that they have a password problem and users who already have coping strategies they think work just fine.

6 Security Discussion

6.1 Security Analysis and User Choice Effects

Password security and user choice issues. The theoretical password space is easily computed (see §3 and Table 1). In practice, two issues can significantly reduce the effective password space. One is hot-spots and related phenomena: certain points on the graphical interface that are more likely to be selected by users. These reflect the general rule that password schemes allowing free user choice suffer from skewed password distributions, a known issue for both text and graphical password schemes [9, 25, 34]. Fig.1(b) shows the frequency of each icon being selected as a part of master passwords—e.g., the number 7 means that 7 out of 20 or 35% of participants chose "BMW" icon as part of their password. A second issue is the predictability of patterns within a set of icon selections. If click-selections are not independent, attackers may reduce the password space even in the absence of highly popular individual icons (cf. [6, 26]). (Recall that §3 explains why, despite the user choice issues, iPMAN uses the GPI scheme rather than, e.g., GPIS or PCCP.) Next, we will analyze the passwords collected in our field study with respect to these two user choice issues.

Impact of Hot-spots on Password Security. As explained now, the field study results confirmed that due to user-choice issues, user-selected passwords differ from a random selection of a similar number of passwords across the password space. To quantify the loss in security, password entropy $H(X) = -\sum_{i=1}^{n} w_i \cdot log(w_i)$ is commonly considered, despite its utility being limited by unknown probabilities. To calculate $H(X)$ we need the number of passwords n and the probability w_i for each (across some target user population). Of course, we do not know the w_i. To derive a coarse estimate as done elsewhere [34], assume that the data collected in our field study is representative of larger populations. We derive all icon permutations that constitute a password, then calculate the probability of occurrence of each password using the frequency of each icon being selected (see Fig. 1(b)). Finally we calculate rough estimates of password entropy using $H(X)$ defined above. Table 1 presents these estimates for different password lengths (different numbers of icons) together with the sizes of the corresponding theoretical (full) password space. We see that the estimated entropy of six-icon iPMAN passwords is considerably lower than the theoretical value, but still not lower than estimates given elsewhere [34] for Passpoints whose theoretical space (2^{43}) matches six-icon iPMAN passwords.

Number of Icons Chosen	3	4	5	6
Theoretical Space (in bits)	21	28	36	43
Entropy Estimate (in bits)	17	22	28	33

Table 1: Theoretical password space and coarse entropy estimate of iPMAN.

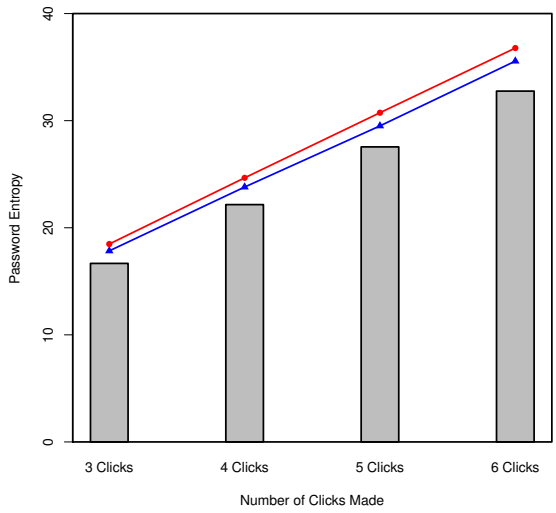
While the above entropy estimation method is useful for approximate results and comparisons, the low estimated entropy is partially due to the small number of passwords involved. For such small password datasets, only a subset of click-points (i.e., icons) could be clicked even in a perfectly random click distribution, inherently causing an entropy value lower than that possible for larger samples. We therefore augment our analysis using "random simulation datasets" [6].

In a random experiment, we randomly chose 10 passwords with six icons each and 10 passwords with exactly the same number of icons each as the users in the Free Choice group. We repeated this experiment 100 times. We computed the entropy for each of the 100 experiments using the coarse approximation explained above. Then, we compared the entropy for this set of 20 randomized selections (using the min and max of the 100 averages) to the 20 in the field study dataset. Fig. 5(a) shows an upper and lower graph line (max and mins found for randomized datasets), and how the data from the 20 users in the field study fall outside the randomized set. Since each randomized dataset represents a chance to include the observed data, with 99% probability the field study dataset did not occur by chance. This gives strong evidence that user choice skewed the password distribution, with a visual representation of the skew.

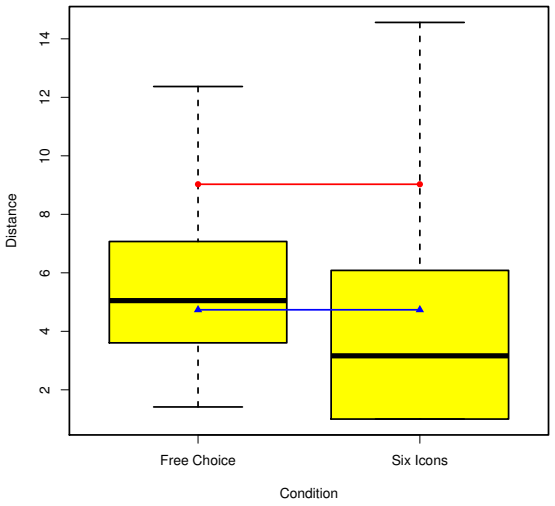
Exploiting Click Patterns. Graphical passwords can be guessed by forming attack dictionaries based on click-order patterns. These dictionaries can be optimized by additional techniques, e.g., image processing methods, visual attention models, human-computed datasets [26]. Our field study does not provide a database rich enough for a comprehensive click-order analysis. But as an example of a possible attack, consider a simple strategy exploiting a click-order pattern based only on locality (clicks being near one another), earlier shown [26] to be successful on Passpoints. Upon examination we see that 4 of 10 participants from the six-icons group clicked adjacent icons in strict left-to-right or right-to-left order. Thus a dictionary with only 150 entries (the total number of icons in the interface) could find 40% of passwords in the six-icons group, using each icon as the start of both left-to-right and right-to-left patterns. This particular pattern was not observed in the free choice group.

Any patterns actually observed in even a small field study are interesting to point out. We have confirmed with the authors of the lab study on GPI [3] that the interfaces were the same in both studies and the click-order pattern of strict left-to-right or right-to-left order did not appear in their lab study. One plausible explanation for this difference in user choices is that in the lab environment participants wanted to be more "secure" to help the study, whereas in the field study there was greater motivation to choose passwords thought to be more easily remembered. This result again highlights the issue of ecological validity of lab studies, possible security impacts, and the difficulty in designing meaningful tests and collecting representative data in password research.

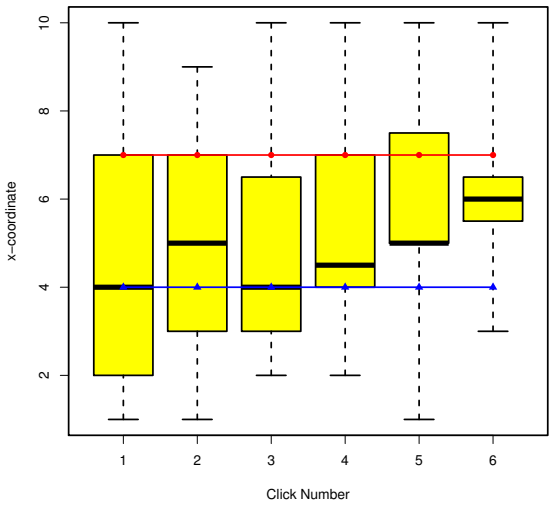
A more sophisticated attack on click-order patterns considers the Euclidean distance between chosen icons, e.g., the length of the segments formed between two clicks. Fig. 5(b) illustrates the distance between



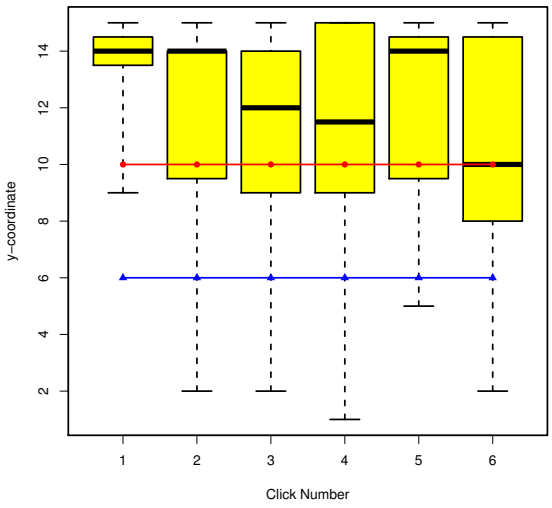
(a) Bar graphs: estimated password entropies. Top lines: randomized max, min.



(b) Box plots of Euclidean distance (in no. of icons) between adjacent clicks.



(c) Box plots of click distribution along panel's x-axis. (max x-coord: 10)



(d) Box plots of click distribution along panel's y-axis (max y-coord: 15)

Fig. 5: Comparison of field study dataset with simulation datasets. Lines in red (with circles) and blue (with triangles) resp. represent the simulation datasets' max and min entropies in (a), and max and min median values in (b), (c), (d). Origin (0,0) is bottom-left in the panels.

two adjacent clicks in free choice and six icons groups together with the min and max median values among randomized datasets. In six icons group, segments are shorter (median 3.1 icons) than those of the simulated datasets. This characteristic may allow an attacker to predict higher-probability passwords.

We also look at how clicks in the field study are distributed across the panel of icons, in Figs. 5(c) (x-axis) and 5(d) (y-axis). The latter illustrates that users tend to favor passwords with icons from top (higher) rows. The pattern for the x-axis is less pronounced, but still biased to the left. Since the medians of the clicks along the y-axis except the last click fall outside of the max (red) and min (blue) lines, again with 99% probability we can say that it is unlikely this pattern would occur by chance.

Impact of Password Rules on Security. In the field study, 9 of 10 users in the free choice group chose passwords consisting of four icons or less. This result reiterates the importance of password length rules for GPI. Although the full password space for the six icons group was significantly larger, we observed that more predictable patterns (i.e., strict order of left-to-right or right-to-left) emerged. Thus, supplemental password rules are advisable for GPI to eliminate the most frequently occurring click patterns. In particular, we suggest proactively disallowing passwords having any two icons adjacent despite the reduction in theoretical password space due to this change. Thus our field study, albeit small, provides important insight towards fine-tuning "little but important" details and instructions, such as password rules and passwords to rule out proactively, informing and motivating a future, larger study.

Master passwords should be salted with friendly salt. Some have argued [11] that offline attacks do not pose great risk to traditional site passwords due to the difficulty of stealing hashed password files from servers. But the risk is greater for iPMAN and similar password managers: attackers have greater incentive to obtain master passwords, and may, for example, first capture a site password through standard phishing techniques, and then mount an offline attack to recover the master password to allow access to many sites.

Even relatively weak individual passwords of about 20 bits may withstand online attacks if lockout rules are in place [11]. On the other hand, master passwords should be from a larger password space to offer protection against offline attacks. Large precomputed look-up tables built once even at relatively high cost may be used to break passwords across all users of a login site. Recall that in the implementation of iPMAN, part of the login site's URL is hashed with the master password to generate site-specific passwords, serving as a domain-specific salt and precluding "global" precomputed tables being used across all sites.

In addition to domain-specific salt which uses the public URL information, the master password can be hashed with a private salt—a user-specific private constant stored locally on the user's machine. Private salting[11] increases the computation necessary to break passwords offline if we assume that the attacker cannot capture the salt, and precludes "site-specific" global precomputed tables. The attacker now has to customize any lookup tables to individual sites as well as to individual users making such tables much less effective. In fact, this can be achieved with (non-private) *personalized salt* as well (e.g., using the username).

While salting raises the bar against offline attacks, it complicates login from a remote or secondary device where the salt is absent. If we could assume that users chose master passwords that (without additional protection) were resistant to offline attacks, one could argue to avoid salting, for transportability reasons (as preferred by Ross et al. [31]). However, this assumption has failed in practice; across many different password systems, effective password spaces have been only tiny subspaces of the full password spaces, due to user choice. In our questionnaire participants reported that most of their text passwords outside our study were 8-9 upper/lowercase alphanumeric characters (thus weak).[12]

Our study also found that a significant portion of passwords are either vulnerable to attacks using very small click-order pattern dictionaries as in the six-icons group, or are short as in the free choice group. Recall that 90% of passwords in the free choice group contained four icons or less; these short passwords could be found using an exhaustive lookup table of around 2^{28} entries (see Table 1). Given this relatively small size of table, one may argue that the major difficulty to carry out an offline attack shifts to capturing site-specific

[11] Hereafter, we use *salt* and *salting* to refer to private salt and private salting.

[12] If master passwords are similarly chosen then this discussion generalizes to password managers with text passwords, but we are aware of no empirical investigation that explores this assumption.

passwords, e.g., through a phishing site; master passwords could then be easily recovered using a lookup table customized to hashing with that site's URL information.

We conclude that for security, it is highly desirable that a password manager implements salting, but to support usability, salt must be user-friendly for secondary devices. To meet this requirement, we next propose a new method for salt generation supporting secondary devices, which might be called fast or friendly salt.

6.2 Fast Salt Generation supporting Secondary Devices

The above findings regarding password choice motivate use of password strengthening for iPMAN. Here we propose a new method supporting secondary devices and preventing offline attacks, with salt generation two orders of magnitude faster than a previous salt generation method by Halderman et al. [17].

The previous proposal first generates $s_t = H^t(username \mid\mid master\_pswd)$ as the user-specific salt, then uses it to generate the site-specific password

$$P = post\_process(H(URL\_info \mid\mid master\_pswd \mid\mid s_t)) \tag{1}$$

Soliciting user input allows s_t to be re-generated on secondary machines, albeit requiring a wait of about 100 seconds [17] using the value of t recommended to make offline attacks sufficiently difficult. Our new method involves contacting a semi-trusted online signature server T to construct a user-specific salt that depends on the username and master password, and is thereafter stored on the primary device. Including username in the calculation makes any information gained through an online interaction with the signature server useful only for that particular username. The new salt is T's digital signature on the hash of the concatenation of the username and master password: $s = sig_T(H(username \mid\mid master\_pswd))$.

We call T semi-trusted, as it does not see the cleartext hash. Instead, the client uses a pair of blinding/unblinding functions $f(\cdot)$ and $g(\cdot)$, obtains a blind signature $sig_T(f(m))$ [22, p.475] and unblinds it by computing $g(sig_T(f(m))) = sig_T(m)$ for $m = H(username \mid\mid master\_pswd))$ to recover s. The signature algorithm must be deterministic (generating a fixed signature for fixed data). Using an RSA blind signature for a server with public key (n, e), private key d, blinding factor k (a fixed integer coprime to n), $f(m) = m \cdot k^e \ (mod\ n)$ and $g(m) = k^{-1}m \ (mod\ n)$, the exchanges between client and signature server are:

Client→Server:	$k^e \cdot m \pmod n$	[blinded hash of master password]
Client←Server:	$k^{ed} \cdot m^d \pmod n$	[blinded signature of hash]

Clients (e.g., browsers using iPMAN) contact the server only when the salt is not locally available. This approach precludes the building of an attack dictionary offline, since a server signature is needed to generate each candidate salt. To protect against online attacks, servers may track signature requests and adopt lock-out strategies with geometrically increasing lock-out times [11].

Our proof-of-concept implementation of the new salt generation method uses Java and Javascript on the server and client sides, respectively. The server side runs on Google's infrastructure as a Google App Engine application. Our client side hard-coded the signature server URL so that it can automatically contact the server. For our client machine with Intel Core2Duo T9400 2.53 GHz CPU, 1024-bit RSA and 64-bit blinding factor, total time to generate the salt (client blinding and unblinding, server processing and network delay) is around 500ms (for 2048-bit RSA, another 500ms). This provides a salting method which prevents offline attacks and incurs drastically less user inconvenience than earlier work to strengthen master passwords.

Similar in concept to our salt generation method, but for a different application environment, the Password-Hardening Protocol [13] provides users strong secrets from passwords by interacting with servers.

7 Conclusion

Our work is the first, to our knowledge, to report in the literature on a field study of a password manager. We believe the knowledge gained will be useful to a broad audience interested in password management. The study found high login success rates and persistent password memory using a manager with an icon-based master password. To counter the observed weakness of user-chosen master passwords—user choice issues now

being generally expected in graphical (and text) password schemes which allow user choice—a new method for salt generation supports secondary devices and significantly reduces the waiting time of earlier proposals.

We recognize, as a major obstacle to voluntary widespread use of tools like iPMAN, the secondary importance users give to password security. Another obstacle is the short-term adoption cost, e.g., users must allocate time and attention to migrate existing passwords. We note that "password wallet" approaches have major usability advantages since they do not require that users migrate their passwords. While it is tempting to conclude that the security benefits of a password manager are large, but not fully appreciated by users, we are aware of no clear scientific evidence or convincing metric to support such a claim. It can also be argued, with equal lack of convincing scientific evidence, that users who reject all advice towards increasing password security (typically, to avoid usability penalties) are making a rational choice [18].

Password managers offer to ameliorate a ubiquitous and significant usability issue which also impacts security: requiring users to choose and remember multitudes of passwords. We encourage more research exploring password manager software which stands up to not only security analysis on paper, but critical issues in practice, including password choice and usability as observed in ecologically valid user studies.

Acknowledgements. We thank Hakan Gurbaslar for help in executing the field study, Robert Biddle and Sonia Chiasson for comments that considerably improved the paper, and anonymous referees. This research is supported by TUBITAK (The Scientific and Technological Research Council of Turkey) under project number 107E227. The fourth author is Canada Research Chair in Authentication and Computer Security, and acknowledges partial funding from NSERC for the chair, a Discovery Grant, and NSERC ISSNet.

References

1. 1Password. `http://agilewebsolutions.com/products/1Password`.
2. M. Abadi, L. Bharat, A. Marais. System and method for generating unique passwords. US Patent 6141760, 1997.
3. K. Bicakci, N.B. Atalay, M. Yuceel, H. Gurbaslar, B. Erdeniz. Towards Usable Solutions to Graphical Password Hotspot Problem. 33rd Annual IEEE Int. Computer Software and Applications Conference, 2009.
4. K. Bicakci, M. Yuceel, B. Erdeniz, H. Gurbaslar, N.B. Atalay. Graphical passwords as browser extension: Implementation and usability study. 3rd IFIP WG 11.11 Int. Conf. on Trust Management, 2009.
5. J. Bonneau and S. Preibusch. The Password Thicket: Technical and Market Failures in Human Authentication on the Web. 9th Workshop on the Economics of Information Security (WEIS), 2010.
6. S. Chiasson, A. Forget, R. Biddle, P.C. van Oorschot. User interface design affects security: patterns in click-based passwords. Int. J. Inf. Security 8(6):387-398, 2009.
7. S. Chiasson, A. Forget, R. Biddle, P.C. van Oorschot. Influencing Users Towards Better Passwords: Persuasive Cued Click-Points. BCS-HCI 2008, Liverpool, U.K.
8. S. Chiasson, P.C. van Oorschot, R. Biddle. A Usability Study and Critique of Two Password Managers. USENIX Security 2006.
9. D. Davis, F. Monrose, M. Reiter. On user choice in graphical password schemes. USENIX Security, 2004.
10. D. Florencio, C. Herley. A large-scale study of web password habits. 16th Int. Conf. World Wide Web (WWW 2007).
11. D. Florencio, C. Herley, B. Coskun. Do Strong Passwords Accomplish Anything? USENIX HotSec 2007.
12. B.J. Fogg. 2003. Persuasive Technologies: Using Computers to Change What We Think and Do. Morgan Kaufmann Publishers, San Francisco, CA, 2003.
13. W. Ford and B. Kaliski. Server-Assisted Generation of a Strong Secret from a Password, 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, June 14-16, 2000.
14. E. Gaber, P. Gobbons, Y. Mattias, A. Mayer. How to make personalized web browsing simple, secure, and anonymous. Financial Crypto '97, LNCS 1318, 1997.
15. N. Govindarajulu and S. Madhvanath. Password management using doodles. In 9th International Conference on Multimodal Interfaces (ICMI), November 2007.
16. P. Guttmann. Manuscript chapters, Usable Security, `http://www.cs.auckland.ac.nz/~pgut001/pubs/usability.pdf`.
17. J.A. Halderman, B. Waters, E.W. Felten. A convenient method for securely managing passwords. 14th International Conf. on World Wide Web (WWW 2005).

18. C. Herley. So long, and no thanks for the externalities: The rational rejection of security advice by users. NSPW 2009.
19. I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin. The design and analysis of graphical passwords. 8th USENIX Security 1999.
20. E. Jung. Passwordmaker. `http://passwordmaker.mozdev.org`.
21. Lastpass. `http://lastpass.com/`.
22. A. Menezes, P.C. van Oorschot, S. Vanstone. Handbook of Applied Cryptography. CRC Press, Boca Raton, 1996.
23. W. Kintsch. Models for free recall and recognition. In D. A. Norman (Ed.), Models of Human Memory, Academic Press, New York, 1970.
24. Mac OS X Reference Library. KeyChain Services Programming Guide. `http://developer.apple.com/library/mac/navigation`.
25. P. C. van Oorschot, J. Thorpe. On predictive models and user drawn graphical passwords. ACM TISSEC 10(4) article 17:1–33, 2008.
26. P.C. van Oorschot, A. Salehi-Abari, J. Thorpe. Purely automated attacks on Passpoints-style graphical passwords. IEEE Trans. Info. Forensics & Security, 5(3):393–405, 2010.
27. OpenID Foundation. `http://openid.net/`.
28. P. van Overschelde, K.A. Rawson, J. Dunlosky. Category norms: An updated and expanded version of the Battig and Montague (1969) norms. Journal of Memory and Language 50:289–335, 2004.
29. Password Safe. `http://passwordsafe.sourceforge.net/`.
30. J. la Poutre. Password composer. `http://www.xs4all.nl/~jlpoutre/BoT/Javascript/PasswordComposer/`.
31. B. Ross, C. Jackson, N. Miyake, D. Boneh, J. Mitchell. Stronger password authentication using browser extensions. USENIX Security 2005.
32. S.-T. Sun, Y. Boshmaf, K. Hawkey, K. Beznosov. A Billion Keys, but Few Locks: The Crisis of Web Single Sing-On. NSPW 2010.
33. H. Tao and C. Adams. Pass-Go: A proposal to improve the usability of graphical passwords, International Journal of Network Security 7(2), 2008.
34. J. Thorpe, P.C. van Oorschot. Human-Seeded Attacks and Exploiting Hot-Spots in Graphical Passwords. USENIX Security 2008.
35. S. Wiedenbeck, J. Waters , J. Birget, A. Brodskiy, N. Memon. PassPoints: Design and longitudinal evaluation of a graphical password system. International Journal of Human-Computer Studies 63(1–2), 2005.
36. K. Yee, K. Sitaker. Passpet: convenient password management and phishing protection. SOUPS 2006.

Appendix: Questionnaire on Usability of iPMAN

The following questionnaire, discussed in §4.2, is included here for completeness, as common in HCI studies.

Perceived Security:
1. My passwords are secure when using iPMAN.
2. I do not trust iPMAN to protect my passwords from cybercrime.
Not Giving Control:
3. I am comfortable with knowing my actual password for a web site.
Ease of Use:
4. iPMAN is difficult to use.
5. I could easily log on to web sites and manage my passwords with iPMAN.
Perceived Necessity:
6. I need to use iPMAN on my computer to protect my passwords.
7. My passwords are safe even without iPMAN.
Additional Yes/No Questions:
8. It was difficult to remember a iPMAN password.
9. The category structure was helpful while I was generating my password.
10. I wrote down my icon password somewhere.
11. I experienced difficulties when I tried to login using remote login site.