An Empirical Study of Mnemonic Sentence-based Password Generation Strategies

Weining Yang, Ninghui Li, Omar Chowdhury, Aiping Xiong, Robert W. Proctor
Purdue University
West Lafayette, IN, USA
{yang469, ninghui, ochowdhu, xionga, rproctor}@purdue.edu

ABSTRACT

Mnemonic strategy has been recommended to help users generate secure and memorable passwords. We evaluated the security of 6 mnemonic strategy variants in a series of online studies involving 5,484 participants. In addition to applying the standard method of using guess numbers or similar metrics to compare the generated passwords, we also measured the frequencies of the most commonly chosen sentences as well as the resulting passwords. While metrics similar to guess numbers suggested that all variants provided highly secure passwords, statistical metrics told a different story. In particular, differences in the exact instructions had a tremendous impact on the security level of the resulting passwords. We examined the mental workload and memorability of 2 mnemonic strategy variants in another online study with 752 participants. Although perceived workloads for the mnemonic strategy variants were higher than that for the control group where no strategy is required, no significant reduction in password recall after 1 week was obtained.

1. INTRODUCTION

Passwords have been the most widely adopted user authentication mechanism in the past and are likely to continue to be an important part of cybersecurity for the foreseeable future due to their ease of use and wide deployment [8, 9, 19]. At the same time, it is well known that there is a tension between the security and usability of passwords [3, 28]. Oftentimes, secure passwords tend to be difficult to memorize (i.e., less usable), whereas passwords that are memorable tend to be predictable. The security community has been trying to come up with password generation strategies that can help users generate secure and usable passwords. Candidate strategies have been suggested by sources ranging from the National Institute of Standards and Technology (NIST) [29] to online comics [2], and from security experts' essays [31, 32] to online help forums. However, these suggestions are often based on intuitions instead of scientific knowledge. Little is actually known about which strategies are effective in helping users create usable and secure passwords.

CCS'16, October 24-28, 2016, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2976749.2978346

Perhaps the most widely recommended and studied strategy is that based on mnemonic sentences: Take a memorable sentence, abbreviate the words, and combine them to form a password. The strategy is generally known as the mnemonic sentence-based strategy (for short, the *mnemonic strategy*). It appears that the general assessment is that this is a good strategy. It is recommended by NIST [29] and by security experts [31, 32]. To our knowledge, three studies on this strategy have been reported, by Yan *et al.* [44, 45], Vu *et al.* [40], and Kuo *et al.* [24]. One standard approach for evaluating the strength of passwords is to use password cracking tools or models to check how many collected passwords can be cracked [21, 22, 23, 27, 34, 37]. Based on this approach, Yan *et al.* [44, 45] claimed that passwords generated using the mnemonic strategy are as strong as random passwords, while the other studies reached a somewhat mixed conclusion regarding its security [24, 40].

These existing studies, however, have limitations. First, they are based on samples of small sizes, with less than 150 passwords under the strategy in each of the three studies. Second, the approach of relying only on checking how many passwords can be cracked to assess the security is flawed. Although such assessment provides useful information about how such passwords fare against today's state of the art cracking methods, the results are often caused by the incompatibility of the cracking techniques and the nature of the mnemonic strategy. Even developing a strategy-specific cracking method, as done in [24], is insufficient. It is always possible that one has overlooked some highly effective attack techniques.

We conducted a much larger study to evaluate 6 variants of the mnemonic strategy and compared them against a control group. When assessing the security of the variants, we went beyond the methods used in existing studies in two ways. First, we adopted the approach of using statistical quantities to measure the distributions of the passwords, as articulated by Bonneau [7]. In particular, we chose to use the β-*guess-rate* (λ_β) [11], which measures the expected success for an attacker limited to β guesses per account. We chose to use $\beta = 1$ and $\beta = 10$, both because they were suggested in [12] as appropriate for defense against online guessing attacks and because a larger β is not very meaningful for our sample sizes (close to 800). Second, we developed a method for attacking passwords resulted from the mnemonic strategy, and demonstrated the effectiveness of this attack.

We chose two of the variants and evaluated their usability in a separate user study, in which password creation time, short-term (*i.e.*, within a few minutes of creation) and long-term (*i.e.*, after 1 week) password recall, and the workload required in both password creation and retention are evaluated.

Our studies were conducted on Amazon Mechanical Turk. They were found to be eligible for exemption from IRB review because

it is research involving survey procedures, and human subjects cannot be identified from the recorded information. Our institution's IRB has also allowed us to share the collected data with other researchers. (The participants were warned not to use their real passwords.)

Contributions. The current paper is the first to investigate the security of password generation strategy variants on a large scale. We recruited a total number of $5,484$ participants, for an average of 783 participants per condition, when evaluating the security of the variants. In addition, we recruited 752 participants for evaluating the usability of two variants against a control condition. Our studies improve the understanding of password generation strategies through the following contributions.

- We show that using the standard cracking-based methodology, password sets obtained under all variants have similar strengths and are all much more secure than the baseline. However, using β-guess-rates, we found that using generic instructions that have been suggested in the literature resulted in 2.5% of the group choosing the same sentence, and the top 10 sentences chosen by 7.8% of the group. We have also found that converting a sentence to a password adds limited entropy. These two facts together suggest that this variant of mnemonic strategy is no more secure than the baseline.

- We show that combining explicit instruction of choosing a personalized sentence that is unlikely to be chosen by others, with the inclusion of such personalized examples, dramatically increases the security of the resulting passwords. Furthermore, using only the explicit instruction or the examples alone results in less secure distributions.

- We show that the instructions for the mnemonic strategy found in the literature and recommended by security experts are not optimal in inducing secure password distributions.

- We found that requiring personalized choice of sentences in mnemonic strategy variants does not reduce the usability of the mnemonic strategy.

To our knowledge, we are the first to observe and experimentally validate the influence of the instructions and the examples accompanying the strategy description on the security of the resulting passwords. It is intuitively understood that precise instructions and demonstrative examples can improve the ease of applying a strategy to generate passwords. However, the relationship between the level of security, the instruction wording, and the examples has not been studied before.

The rest of the paper is organized as follows. We discuss related work in Section 2. We present an overview of the first study and the methodology used for evaluating security of the variants in Section 3, and the evaluation results are presented in Section 4. We then present the study regarding usability of the variants in Section 5. We discuss the consequence of our findings as well as our studies' limitations in Section 6, and conclude with Section 7.

2. RELATED WORK

Evaluation of the mnemonic strategy. Yan *et al.* [44, 45] conducted a study with college students who were given accounts on a central computing facility. The students were randomly assigned to three groups. The control group (95 members) were asked to create a password with at least seven characters long that contained at least one non-letter. The random password group (96 members) received a sheet of paper with the letters A through Z and the numbers 1 through 9 printed repeatedly on it; participants were asked to close their eyes and randomly pick eight characters. (They were also advised to keep a written record until they had memorized the password.) The mnemonic password group (97 members) were told to create a sentence of 8 words and choose letters from the words to make up a password, mixing upper-case and including at least one non-letter. Yan *et al.* [44, 45] found that very few users asked the system administrator to reset their passwords. Responses to an email memorability survey showed that the mnemonic passwords were similar to the control group in terms of difficulty to use, and the random passwords were found to be significantly more difficult. An attack with dictionaries (with permutations with digits) cracked 32% for the control group, 8% for the random password group, and 6% for the mnemonic password group. The authors concluded "*We've debunked another folk belief that random passwords are better than passwords based on mnemonic phrases. In our study, each appeared to be as strong as the other*."

Vu *et al.* [40] studied two variations of the mnemonic strategy: (A) Choose a sentence containing at least 6 words, and use the first letters from each word as the password; (B) strategy A with an additional requirement that users should embed a special character or digit in the password. Forty Psychology students were each asked to create 3 passwords using one of the above strategies. In terms of memorability, they found that participants using strategy B "*took two times longer to recall the passwords, made almost twice as many errors before being able to recall the password, and completely forgot the password twice as often*". Within 12 hours, the L0phtCrack4 (LC4) password cracker cracked all passwords generated with strategy A, whereas only 5% of the passwords from strategy B were cracked.

Kuo *et al.* [24] conducted a study in which 144 subjects were asked to generate mnemonic passwords, with 146 subjects in the control group. For the control group, they used John the Ripper's 1.2 million-word English dictionary, and were able to crack 11% of the 146 passwords. For the mnemonic group, they collected $129,000$ sentences from the Internet and, with some mangling, created a $400,000$-entry mnemonic password dictionary. Using this dictionary, they cracked 4% of the 144 mnemonic passwords. A bruteforce attack cracked an additional 8% in the control group, and an additional 4% in the mnemonic group. Kuo *et al.* also searched the Internet (using Google) for the sentences used by the users to generate passwords, and were able to find 65% of them on the Internet. Based on this evidence, the authors concluded that "*Mnemonic phrase-based passwords are not as strong as people may believe, ...*".

We argue that the fact that a password is generated by a sentence that can be found on the Internet does not necessarily mean that it is weak, given that there are likely billions or tens of billions of sentences on Google-indexed pages. Similarly, that a size-$400,000$ dictionary can crack 4% of password seems more like an indicator of strength to us. Using a list of $400,000$ top passwords from Rockyou, one could crack 32% of the passwords in the Yahoo password dataset [1], and 39% of the passwords in the phpBB dataset [1]. Our interpretation of the data in [24] is that mnemonic sentence-based passwords are significantly stronger than the baseline, as measured by passwords in the Yahoo and phpBB dataset, with two caveats. First, this is based on cracking results obtained by using their particular dictionary. Second, the conclusion may not be statistically significant because the dataset is small.

Other related work. One standard approach to study the strength of password choices under different settings is to use password cracking tools or probabilistic password models to check the num-

ber of passwords cracked [22, 27, 33, 34, 38], e.g., when facing different password policies [21, 23], when presented with different password strength meters [15, 37], when forced to change passwords due to organizational change of password policies [35], when forced to change passwords due to expiration [46], when "persuaded" to include extra randomness in their password choices [17], when allowed to replace some characters from a randomly generated password [20], and when facing different guidance and feedback [33]. The strength of passwords was generally represented by using the guess number graphs, which plot the percentage of passwords cracked in the dataset vs. the number of password guessing attempts. Ma *et al.* [26] proposed the probability threshold graphs which convey the same information as guess number graphs when assessing the quality of passwords. Bonneau [7] proposed metrics for studying the overall level of security in large password datasets, based only on the distribution, and not on the actual password strings.

Schechter *et al.* [30] recommended to strengthen user-selected passwords against statistical guessing attacks by allowing users to choose any passwords they want, so long as it is not already too popular with other users. We follow the method of using statistical quantities to assess strength of distributions, as advocated by [7].

Some have suggested that users should simply use password managers and remember just one password. Password managers, however, create their own security, reliability, and convenience problems [14, 25, 36, 47, 48]. Perhaps the biggest concern is that a password manager software takes the security of all critical websites out of the hand of the user and puts it in one piece of software, creating a single point of failure and an attractive target for attackers at the same time. Recently, Xing *et al.* [43] showed that Unauthorized Cross-App Resource Access (XARA) vulnerabilities on Apple OS X and iOS enable malicious applications to read passwords saved into Apple Keychain and passwords saved in the popular 1Password password manager. These results demonstrate the risk of relying on one password manager for all critical websites.

3. STUDY 1: SECURITY

We studied 6 variants of mnemonic strategy. In such a strategy, a participant is asked to first select an easy-to-remember sentence, and then convert the sentence into a password.

Table 1 gives the detailed descriptions of the 6 variants in our study. We urge readers of this paper to read Table 1 before proceeding, as the differences in the strategy descriptions are important parts of the study. Below is a summary.

- MneGenEx (Mnemonic-Generic-Example, with generic instruction and a generic example, similar to what used in Kuo *et al.* [24]),
- MnePerEx (Mnemonic-Personalized-Example, with emphasis on using personalized choices of sentences that other people are unlikely to use and a personalized example),
- MnePer (Mnemonic-Personalized, with emphasis on personalized choice of sentences, but no example),
- MneEx (Mnemonic-Example, with multiple personalized examples, but no emphasis on personalized choices of sentences),
- MneSchEx (Mnemonic-Schneier-Example, with some emphasis on personalized choices and mixed examples, suggested by Schneier in [31, 32]),
- MneYanEx (Mnemonic-Yan-Example, with some emphasis on personalized choices in some examples, used by Yan *et al.* [44, 45] in their studies).

In addition to the 6 variants, a control group Control, in which we ask for passwords containing at least 8 characters without any extra restriction, was included in the study as well.

3.1 Study Design

We conducted the study through Amazon Mechanical Turk (MTurk), and all participants were at least 18 years old. We limited our data collection to participants from the United State because the strategy variants were constructed using the English language. The study was divided into 7 rounds, one for each of the 7 conditions. Participants were allowed to participate in only one round. If a participant was in more than one rounds of the study, we kept only the data from the first time that the participant was in.

Participants were asked to type the sentence used in the intermediate step. After that, participants were asked to enter the password they created twice, and the password typed in each time had to match each other before they could proceed.

We warned participants not to use their actual passwords. In the study, we forbade passwords that were the same as the examples and that did not appear to be generated following the instructions. In MneGenEx, MnePerEx, MnePer, and MneEx, we required the length of the password to be identical to the number of words, and further checked if a letter in a password can be found in the corresponding word in the sentence; no check was performed for digits and special symbols in the password. In MneSchEx and MneYanEx, a participant was allowed to keep a complete word in the resulting password; thus we cannot use the above approach. Instead, we required that the sequence of letters (ignoring special symbols or digits) in a password was a subsequence of the sequence of letters in the sentence.

3.2 Methodology

Our goal in this study is to assess the security of the passwords generated by the different password generation strategy variants. The traditional approach to assess the strength of passwords generated under a given setting is to use password cracking tools or probabilistic password models to plot guess number graphs [21, 22, 24, 23, 27, 34, 37] or probability threshold graphs [26].

The above approach can assess the security of passwords against current password cracking tools and probabilistic password models, which are adapted to today's password distributions; however, it cannot adequately assess the strength of password creation strategies against attacks targeting these strategies. If a strategy is widely used, then attackers may develop strategy-specific methods which can efficiently guess the passwords. For any password creation strategy, one attack strategy is to conduct a dictionary attack which use password datasets created using the strategy as the dictionary. For the mnemonic strategy, an adversary can also create a dictionary of sentences that people are likely to use, and then generate guesses from the sentences.

An alternative approach, as articulated by Bonneau [7], is to measure the probability distribution induced by the strategy. A number of metrics on the strength of password distributions have been proposed by Bonneau [7]. In the case of evaluating passwords obtained from user subject studies, the datasets are quite small (on the scale of several hundreds in our case). One metric that is appropriate for small datasets is the β-*guess-rate* (λ_β) [11], which is the total probability of the most common β passwords with some small β. λ_β measures the expected success for an attacker limited to β guesses per account. Brostoff and Sasse [12] suggested 10 as the allowed failure counts before the account is locked.

Given a sample set S, we use $\mathsf{top}(S)$ to denote the number of times that the most frequent password appears in S, and $\mathsf{top}_{10}(S)$

Table 1: Mnemonic-based Strategy Variants

Variant	Short Description	Exact instruction given to the users in the study
MneGenEx	Mnemonic with generic example, used in Kuo *et al.* [24]	1. Think of a memorable sentence or phrase containing at least seven or eight words. For example, "*Four score and seven years ago our fathers brought forth on this continent*". 2. Select a letter, number, or a special character to represent each word. A common method is to use the first letter of every word.For example: four ⇒ 4, score ⇒ s, and ⇒ &. Combine them into a password: 4s&7yaofb4otc.
MnePerEx	Mnemonic with emphasis on personalization, with an example.	1. Think of a memorable sentence or phrase that is meaningful to you, and other people are unlikely to use. The sentence or phrase should contain at least eight words. For example, "*I went to London four and a half years ago*". 2. Select a letter, number, or a special character to represent each word. A common method is to use the first letter of every word. For example: went ⇒ w, four ⇒ 4, and ⇒ &. Combine them into a password: iwtl4&ahya.
MnePer	Mnemonic with emphasis on personalization, without giving an concrete example.	1. Think of a memorable sentence or phrase that is meaningful to you, and other people are unlikely to use. The sentence or phrase should contain at least eight words. 2. Select a letter, number, or a special character to represent each word, and combine them to create the password.
MneEx	Mnemonic with several personalized phrases as examples.	1. Think of a memorable sentence or phrase containing at least eight words. 2. Select a letter, number, or a special character to represent each word, and combine them to create the password. The following are some examples: "*In June 2013, my wife and I visited Tokyo, Kyoto, and Sapporo*" might become "i63mw&ivTk&$". "*Run 5 miles per week for my first half marathon*" might become "r5mpw4mfhm". "*My high school classmates had a reunion in July 2014*" might become "Mhscharij2". "*I sold my gold Toyota corolla when it had close to 120000 miles*" might become "i$mgtcwIhc21m". "*Danny bought the book The Razor's Edge from me for five dollars*" might become "Dbtb-trefm45d". "*Save money for traveling with my parents to Germany*" might become "S$4twmp2G".
MneSchEx	Mnemonic with mixed examples, used in [32]	1. First create a personally memorable sentence (choose your own sentence – something personal). 2. Then use some personally memorable tricks to modify that sentence into a password. The following are some examples: "*This little piggy went to market*" might become "tlpWENT2m". "*When I was seven, my sister threw my stuffed rabbit in the toilet*" might become "WIw7,mstmsritt". "*Wow, does that couch smell terrible*" might become "Wow...doestcst". "*Long time ago in a galaxy not far away at all*" might become "Ltime@go-inag faaa!". "*Until this very moment, these passwords were still secure*" might become "utvm,tpwstillsecure".
MneYanEx	Mnemonic with mixed examples, used in [44, 45].	1. Please create a simple sentence of 8 words and choose letters from the words to make up a password. You should put some letters in upper case to make the password harder to guess; and at least one number and/or special character should be inserted as well. 2. Use this method to generate a password of 7 or 8 characters. An example of such a composition might be using the phrase is "*It's 12 noon I am hungry*" to create the password "I's12&Iah" which is hard for anyone else to guess but easy for you to remember. By all means use a foreign language if you know one: the password "AwKdk.Md" from the phrase "*Anata wa Kyuuketsuki desu ka ... Miyu desu*" would be an example. You could even mix words from several languages. However, do not just use a word or a name from a foreign language.

to denote the total number of the times the 10 most frequent items occur in S. The estimated density of top 1 and top 10 passwords are calculated by $\tilde{\lambda}_1 = \frac{\mathsf{top(S)}}{|S|}$ and $\tilde{\lambda}_{10} = \frac{\mathsf{top_{10}(S)}}{|S|}$, where $|S|$ is the size of S.

We want to tell whether the differences in these metrics between two datasets are statistically meaningful or not, given that we have small datasets. To address the issue, for a given β, we test the null hypothesis that the total density of top β passwords in the two datasets are the same using the *two proportion z-test*, calculated by:

$$z = \frac{p_1 - p_2}{\sqrt{p(1-p)(\frac{1}{n_1} + \frac{1}{n_2})}}$$

where $p_1 = \frac{x_1}{n_1}$ and $p_2 = \frac{x_2}{n_2}$ are the two proportions from the two samples, i.e., total density of top β passwords in two datasets; n_1 and n_2 are the size of the two datasets; p is pooled sample proportion, which is estimated by $\frac{x_1+x_2}{n_1+n_2}$; and x_1 and x_2 are the total frequencies of top β passwords in the two datasets.

We also apply these metrics to the sentences used in generating passwords, because passwords based on the same sentence are not independent of each other.

4. RESULTS FROM STUDY 1

We recruited 864, 793, 797, 795, 982, and 870 participants for the 6 variants of the mnemonic strategy. After removing duplicate participants, the number of participants we accepted was 864, 777, 753, 745, 868, and 799, respectively. The number of participants recruited in the control group (Control) is 678. In total, 5,484 (3,205 female) participants were involved. The participants' ages ranged from 18 to over 50, with about 70% between 23 to 50 years. Most of the participants were college students or professionals who had bachelor or higher degrees. The demographic distributions in the 7 groups were similar.

4.1 Analyzing Passwords Using Probability Models and Password Strength Meters

We first evaluate the strength of passwords generated using the variants as well as two commonly used datasets Yahoo and phpBB against today's attacks, utilizing (1) the 5-order Markov Model trained on Rockyou dataset, (2) Google password strength API[1], and (3) Zxcvbn [41] deployed by Dropbox. Google password strength API produces an integer score from 1 to 4 for a password. Passwords with score 1 are considered too weak and are forbidden by Google, and passwords with score 4 are considered strong. Zxcvbn gives an estimation of minimum entropy for a password. The entropy is calculated by first dividing the password into chunks, and then combining the entropy estimated for each chunk. Different ways of dividing the password results in different estimated entropy, and Zxcvbn uses the smallest entropy as its output.

In Fig 1(a), each curve conveys the strength of a password dataset evaluated by a 5-order Markov Model trained on Rockyou dataset. A point (x, y) on a curve means that in the corresponding dataset, y percentage of passwords have probability no less than 2^{-x}. Curves in Fig 1(b) and Fig 1(c) illustrate the evaluation based on Google password strength API and zxcvbn, respectively. A point (x, y) on a curve means y percentage passwords in the corresponding dataset has a score no higher than x. In the graphs, a lower curve means passwords from the corresponding variant are considered stronger.

In all three graphs, the curve for the control group (Control) is below the curves for Yahoo and phpBB, indicating that passwords created in the study are stronger than that in real-world datasets. Therefore, the security of passwords created in the study can serve as a lower-bound measurement. On the other hand, curves for Yahoo, phpBB, and Control are significantly higher than the other curves. This indicates that according to the metrics, passwords created without any specific strategy are significantly weaker than those following the mnemonic strategy. When guessing according to the order suggested by the metrics, more passwords in Yahoo, phpBB, and Control will be cracked than passwords from any mnemonic strategy variant. For example, if all passwords with score less than 25 measured by Zxcvbn are attempted, more than 50% of passwords from Yahoo, phpBB, and Control will be covered, while in the 6 mnemonic strategy variants, the percentage of passwords cracked is less than 15%. However, this conclusion is due to the fact that the model or the meters are designed to evaluate generally selected passwords, which is broadly similar to passwords in Yahoo and phpBB, and passwords generated from the mnemonic strategy result in quite different distributions.

Table 2 shows the average lengths of passwords generated from the 6 variants and the control group (Control) as well as passwords in Yahoo and phpBB datasets. From the table, we can observe that passwords generated by using MneSchEx is longer than passwords generated by using other variants. This is because the instructions for MneSchEx use examples in which some whole words (instead of just one character) are included when converting a sentence into a password, and some participants followed the same practice. We also observe that the average length of passwords from MneGenEx is longer than those from variants that require personalized choice of sentences (MnePerEx, MnePer, MneEx). This is because the length of personalized sentences (and the resulting passwords) is relatively easy to control, and people generally prefer short passwords. On the other hand, many users selected well-known quotes in MneGenEx; these quotes can be quite long. From the table, we can also observe that passwords created in Control is longer than passwords in Yahoo and phpBB datasets. This likely contributes to the conclusion that passwords from Control are stronger than those from Yahoo and phpBB according to the metrics in Fig 1. However, although the average length of passwords in Control is longer than passwords from some mnemonic variants, passwords in Control are weaker than them according to Fig 1. This suggests that character sequences in passwords from the mnemonic strategy appear relatively infrequently in dictionaries and common passwords.

[1]https://accounts.google.com/RatePassword

4.2 Strength of Mnemonic Sentences

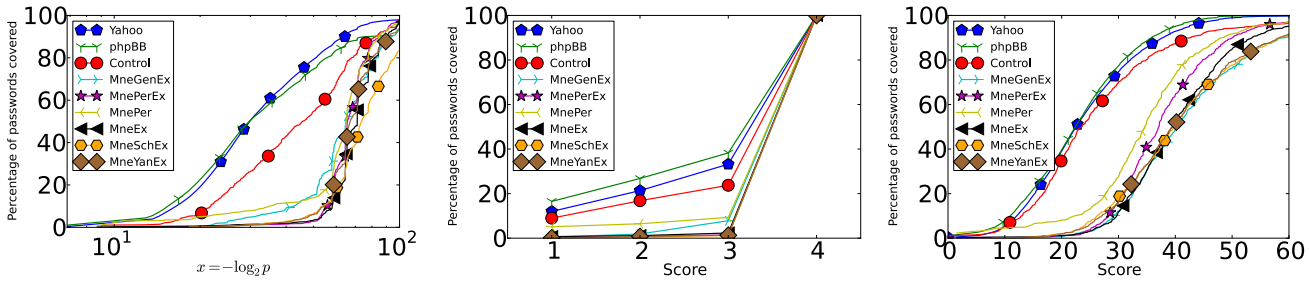
We evaluate the strength of sentences used in the mnemonic strategy as well as the resulted passwords utilizing $\tilde{\lambda}_1$ and $\tilde{\lambda}_{10}$ metrics. Table 3 shows the $\tilde{\lambda}_1$ (top) and $\tilde{\lambda}_{10}$ (top_{10}) values of passwords generated by the control group (Control). Also shown in the table are the quantities evaluated on sets of 800 passwords randomly sampled from three commonly used password datasets Rockyou, phpBB and Yahoo. Table 4 gives the $\tilde{\lambda}_1$ (top) and $\tilde{\lambda}_{10}$ (top_{10}) values of sentences and the resulted passwords for all mnemonic sentence-based variants.

The control group (Control)**.** In Control, the $\tilde{\lambda}_1$ and $\tilde{\lambda}_{10}$ are 0.9% and 2.9%, respectively, which are close to the quantities from the real-world datasets. For instance, the Rockyou dataset has $\tilde{\lambda}_1 = 0.9\%$ and $\tilde{\lambda}_{10} = 3.1\%$. Although the passwords created in the study are stronger than those in real-world datasets, according to the existing strength metrics, as illustrated in Fig 1, the strengths of the weakest passwords are similar. If an adversary is limited to try 10 passwords per account (e.g., by rate limiting), a similar number of accounts will be compromised.

Finding 1: Using generic instructions and examples results in weak passwords. MneGenEx uses the instructions as in [24] and one of the examples used in [24]. We were truly surprised by the high frequencies of the most common sentences and passwords. Among the 864 participants, there were 57 sentences chosen more than once, for a total of 179 times, and the 10 most popular sentences (top_{10}) were picked 68 times. 22 participants chose the famous quote "*to be or not to be, that is the question*". This yielded $\tilde{\lambda}_1 = 2.5\%$, $\tilde{\lambda}_{10} = 7.8\%$. See Table 5 for other commonly chosen sentences, which are also well-known quotes in general, and the resulting passwords.

In terms of those passwords, if passwords are case-insensitive, 36 passwords generated by following the MneGenEx variant appeared more than once, and the most common password was chosen 8 times, with $\tilde{\lambda}_{10} = 5.3\%$. Even taking case-sensitivity into account, there were still 27 non-unique passwords, with the top count number to be 7, and $\tilde{\lambda}_{10} = 3.1\%$, as the majority of the participants did not use upper-case letters. Comparing $\tilde{\lambda}_{10}$ resulted from Control (2.9%) and MneGenEx (4.1%), it appears that the password distribution resulted from MneGenEx is likely to be weaker than Control.

Finding 2: Instructions specifically requesting personalized sentences and containing appropriate examples lead to strong



(a) 5-order Markov Model (b) Google API (c) Zxcvbn

Figure 1: Comparison of strength of passwords resulted from different variants and datasets using probabilistic models and passwords strength meters.

Table 2: Average length of passwords in each variant as well as Yahoo and phpBB datasets.

Variant/Dataset	Yahoo	phpBB	Control	MneGenEx	MnePerEx	MnePer	MneEx	MneSchEx	MneYanEx
Avg. Length	7.6	8.3	10.4	10.1	9.2	9.1	9.4	11.4	9.6

Table 3: $\tilde{\lambda}_1$ (top) and $\tilde{\lambda}_{10}$ (top$_{10}$) in Control as well as samples with size 800 from Rockyou, phpBB, and Yahoo. E_{SD} means the average if E and the standard deviation is SD.

Variant	Count	$\tilde{\lambda}_1$ (top)		$\tilde{\lambda}_{10}$ (top$_{10}$)	
		Case Insensitive	Case Sensitive	Case Insensitive	Case Sensitive
Control	678	1.2%(8)	0.9%(6)	3.4%(23)	2.9%(20)
Rockyou^s	800	$1.0\%_{0.3\%}$(7.7)	$0.9\%_{0.4\%}$(7.5)	$3.1\%_{0.5\%}$(25.0)	$3.1\%_{0.5\%}$(24.4)
phpBB^s	800	$1.2\%_{0.4\%}$(9.5)	$1.2\%_{0.4\%}$(9.5)	$3.8\%_{0.6\%}$(30.2)	$3.8\%_{0.5\%}$(30.2)
Yahoo^s	800	$0.4\%_{0.2\%}$(3.5)	$0.4\%_{0.2\%}$(3.5)	$2.1\%_{0.3\%}$(16.5)	$2.0\%_{0.3\%}$(16.3)

Table 4: $\tilde{\lambda}_1$ (top) and $\tilde{\lambda}_{10}$ (top$_{10}$) in mnemonic strategy variants.

Variant	Count	$\tilde{\lambda}_1$ (top)			$\tilde{\lambda}_{10}$ (top$_{10}$)		
		Sentence	Password		Sentence	Password	
			Case Insensitive	Case Sensitive		Case Insensitive	Case Sensitive
MneGenEx	864	2.5%(22)	0.9%(8)	0.8%(7)	7.8%(68)	5.3%(46)	4.1%(36)
MnePerEx	777	0.1%(1)	0.1%(1)	0.1%(1)	1.3%(10)	1.3%(10)	1.3%(10)
MnePer	745	0.7%(5)	2.3%(17)	2.3%(17)	2.8%(21)	5.8%(43)	5.6%(42)
MneEx	868	0.7%(6)	0.2%(2)	0.2%(2)	2.2%(19)	1.7%(15)	1.3%(11)
MneSchEx	753	0.4%(3)	0.5%(4)	0.3%(2)	2.8%(21)	1.7%(13)	1.5%(11)
MneYanEx	799	0.3%(2)	0.3%(2)	0.3%(2)	1.6%(13)	1.5%(12)	1.4%(11)

passwords. MnePerEx explicitly asked users to choose personalized sentences that other people are unlikely to choose with an example "I went to London four and a half years ago". Among the 777 participants, there was no sentence or password selected more than once. We observed that 536 sentences start with "I" or "my", suggesting a personalized choice. In comparison, such sentences appeared only 125 times in MneGenEx. We noted that not all participants chose personalized sentences. Common sentences such as "to be or not to be, that is the question" still occur in the dataset. Because they occur with much lower frequencies, we did not observe any collision in the dataset. With larger datasets, collisions are bound to occur. As a result, the $\tilde{\lambda}_{10}$ value (1.3%) in sentences selected in MnePerEx was significantly smaller than that from MneGenEx ($z = 6.26, p < 0.001$), and the comparison of the resulted passwords between MneGenEx and MnePerEx leads to similar results. This indicates that in terms of security, MnePerEx is significantly better than MneGenEx based on the $\tilde{\lambda}_1$ and $\tilde{\lambda}_{10}$ metrics.

Finding 3: Commonly suggested instantiations are worse than MnePerEx**.** Seeing results from MneGenEx and MnePerEx, it was clear to us that the instructions played a critical role in the level of security. We then tried to evaluate the precise instructions suggested in Bruce Schneier's two blog posts [31, 32]. We noted that the instructions in the two posts were slightly different. Our version, MneSchEx, was based on the version in [32], which was the more elaborated one. MneSchEx had several differences from MnePerEx. First, it gave 4 examples, some of which are popular, e.g., "Long time ago in a galaxy not far away at all", others are more personalized "When I was seven, my sister threw my stuffed rabbit in the toilet". Second, in the examples, some words are completely kept in the resulting passwords. Third, while the instructions said "Choose your own sentence – something personal"; it did not include the phrase "other people are unlikely to use".

The results came back at somewhere in between MneGenEx and MnePerEx. Among 753 participants, 9 different sentences were not uniquely chosen, with the most common sentence appearing 3 times and the $\tilde{\lambda}_1$ was 0.4%. The $\tilde{\lambda}_{10}$ of sentence selected was 2.8%. There was only a single password selected twice. The $\tilde{\lambda}_{10}$ from MneSchEx was significantly larger than that from MneGenEx ($z = 4.47, p < 0.001$), and was significantly smaller than MnePerEx

Table 5: Popular passwords and probability for top 5 frequently chosen sentences in mnemonic strategy variants.

Rank	Sentences	Passwords	Frequency
MneGenEx (864)			
1	to be or not to be, that is the question (22)	2bon2btit? (7); 2bon2btitq (6); tbontbtitq (1); 2Bon2Btit? (1); 2B0n2bt1tq (1); 2bontbtitq (1); 2brn2btstq (1); 2brn2btit? (1)	2.55%
2	the quick brown fox jumped over the lazy dog (9)	tqbfjotld (2); Tqbfjotld (2); t@bfj0tld (1); tqb4j0tld (1); TQ35j#TLd (1); tqbfjt̂ld (1); Tq8fj0tld (1)	1.04%
3	one small step for man, one giant leap for mankind (6)	1ssfm1glfm (3); 1ss4m1gl4m (1); 1$$4m1gl4m (1); ossfmoglfm (1)	0.69%
4	a penny saved is a penny earned (5)	apsiape (3); @p$i@p3 (1); apsiApe (1)	0.58%
5	in the beginning, god created the heavens and the earth (5)	itbGcth&te (1); 1t8GctH&t3 (1); itbGcth&tE (1); ItbGctHatE (1); NtbGcth (1)	0.58%
MnePerEx (777) No collisions found.			
MnePer (745)			
1	I love you to the moon and back (4)	12345678 (1); !l0t7m@b (1); ily2tmnb (1); !@#$%^&* (1)	0.67%
2	it was the best of times it was the worst of times (3)	iwtb0*iwtw0* (1); Iwtbotiwtwot (1); 233425233525 (1)	0.40%
3	the quick brown fox jumped over the lazy dog (2)	tqbfjotld (2)	0.27%
4	don't look a gifthorse in the mouth (2)	dlaghitm (1); d*1gh0t% (1);	0.27%
5	down by the bay where the watermelons grow (2)	dbhaw!rg (1); DBTBWTWG (1)	0.27%
MneEx (868)			
1	the quick brown fox jumped over the lazy dog (6)	tqbfjotld (1); 7qbxj07ld (1); tQbfj0tld (1); +qbfj0+ld (1); tqbfj0tld (1); tQbfôtzd (1)	0.69%
2	to be or not to be that is the question (3)	2Bon2Btit? (1); 2bontbtitq (1); 2bon2b,it? (1)	0.35%
3	my very educated mother just served us nine pizzas (2)	Mvemj$u9p (1); mvemjsu9p (1)	0.23%
4	I like big butts and I cannot lie (2)	1lbba1cl (1); 1lbb&1cnl (1);	0.23%
MneSchEx (753)			
1	four score and seven years ago (3)	4score7yo (1); foscanseyeag (1); fscrn7yrg (1)	0.40%
2	the quick brown fox jumps over the lazy dog (3)	tqbFOXjotlDOG (1); tqbfjotld (1); Tqbfjotld (1)	0.40%
3	once upon a time (2)	O345$&on@tim8 (1); 1ceupontme (1)	0.27%
4	I love to eat pizza (2)	eyeL2EZa (1); ILtePi&&a (1);	0.27%
5	I love dark chocolate (2)	eyeluvdrkchoco (1); heartDlate (1);	0.27%
MneYanEx (799)			
1	the quick brown fox jumped over the lazy dog (2)	tqbfjotld (2);	0.25%
2	i like big butts and i cannot lie (2)	ilbbaicl (1); Ilbbaicl (1);	0.25%
3	the quick brown fox jumped over the dog (2)	Tqbf&jotD (1); TQbfdreg (1);	0.25%

($z = 2.08, p = 0.019$). One might notice that if passwords are not case-sensitive, the frequency of the most common password was more than the max count of sentences. The four repeated passwords actually came from 3 variations of the same sentence "*the quick brown fox jumps over the lazy dog*", "*the quick brown fox jumped over the lazy dogs*", and "*the quick brown fox jumped over the lazy dog*".

We also studied the effect of the instructions and examples used in Yan et al. [44, 45] (MneYanEx). In MneYanEx, the instructions for creating passwords was relatively generic. However, "hard for anyone else to guess" was explicitly mentioned in the examples. As a result, both the $\tilde{\lambda}_1$ (0.3%) and $\tilde{\lambda}_{10}$ (1.6%) in sentence choices from MneYanEx was less than those from MneSchEx, but is more than that from MnePerEx. The difference in $\tilde{\lambda}_{10}$ between MneYanEx and MneGenEx was significant ($z = 5.91, p < 0.001$).

Finding 4: Both personalized sentences and high-quality examples are needed to achieve better security. Another question is whether the instructions or the examples have more influence on the unpredictability of the chosen sentences and consequently the generated passwords. This led us to study the two variants MnePer and MneEx. MnePer asked for personalized sentences in instructions, but did not provide any example; while MneEx did not explicitly ask for personalized sentence in the instructions, but provided a list of personal sentences as examples. For MnePer, the most popular one was chosen 5 times ($\tilde{\lambda}_1 = 0.7\%$), and $\tilde{\lambda}_{10}$ was 2.8%. $\tilde{\lambda}_{10}$ from MnePer was significantly smaller than that from MneGenEx ($z = 4.42, p < 0.001$), and was significantly larger than that from MnePerEx ($z = 2.11, p = 0.017$). For MneEx, the most popular one was chosen 6 times ($\tilde{\lambda}_{10} = 0.7\%$), and $\tilde{\lambda}_{10}$ was 2.2%. $\tilde{\lambda}_{10}$ from MnePer was significantly smaller than that from MneGenEx ($z = 5.41, p < 0.001$), and was larger than that from MnePerEx ($z = 1.39, p = 0.083$). There was no significant difference between the MnePer and MneEx.

An unexpected finding is that in MnePer, while the 10 most popular sentences were chosen only 19 times, $\tilde{\lambda}_{10}$ in password choices was 5.6% ($\mathsf{top}_{10} = 42$). In all other variants, the $\tilde{\lambda}_{10}$ in password selections was less than that in sentence selections, since the same sentence can result in different passwords. Why do we have higher frequency in popular passwords than in popular sentences? Examining the dataset we found that a significant fraction of users chose pure digit sequence passwords (such as 12345678, 233425233525)

that did not appear to match the sentences. (Since we allow letters to be replaced with digits, we did not check for such situations.) It appears that when users are not shown any examples, some users do not know how to follow the instruction.

Overal, our results suggest that neither explicit request for personalized sentences nor high-quality examples by itself suffice (in fact, neither appears to be more important than the other), and one needs both to get high security.

4.3 Cracking Mnemonic Passwords

We now develop a method for cracking passwords generated using the mnemonic strategy. Our goal is to demonstrate that the step of converting sentences to passwords provides only limited extra entropy. Given the sentences, we can crack more than half of the passwords selected by the users in between 5 and 10 guesses.

For ease of exposition, we first explain our method for case-insensitive passwords. When generating passwords by following the mnemonic strategy, a word can theoretically be mapped to any character; however, given a word, the number of characters that are chosen by users is limited in practice. People generally just pick the first letter of each word. When ignoring case differences, on average, each word is converted to 4 possible characters. From Table 6, we can see that on average 81.2% of the words are converted into their first character; furthermore, about 3.3% of the time, an additional leet substitution is applied. For mappings not using the first letters, the characters chosen are almost fixed for a given word; most of them are based on pronunciation or the meaning of the word. For instance, "*to*" is mapped to "2", "*question*" is mapped to "?", and "*first*" is mapped to "1".

Given a training dataset which contains pairs of sentences and passwords, we first learn the probability distribution of the word-to-character mappings. We classify words into *normal words* and *special words*. Normal words are typically mapped to the first character, with a possible leet substitution. For each letter, we maintain a probability distribution of how that letter is likely to mapped into. Special words are often not mapped to its first letter. For each special word, we maintain a probability distribution for its mappings.

The classification of words is an iterative process. At the beginning, we assume that all words which appear at least 5 times are normal words. In each iteration, we first calculate the probability distribution of each character by averaging the converted character distribution of all corresponding words. Then, we find the `L1` distance between the converted character distribution of each word and the probability distribution of its first character. If the `L1` distance is larger than a certain threshold, we say that the word is a special word. In our experiment, the threshold value we use is 0.6. We repeat the process until no words are removed from normal words.

For password cracking, given a sentence, we first generate a guess by taking the first character of all the words. Then, we generate the passwords by converting words into characters. We assume that in a sentence, the same words are always converted in the same way, and different words are converted into characters independently. Therefore, the probability of each generated password is the product of the probabilities of the transitions from all unique words to characters. We generate passwords in the descending order of probability.

We evaluate the method on the sentences and passwords we collected from MneGenEx, MnePerEx, MnePer, MneEx by cross validation, *i.e.*, we train the model on data from three variants and attempt to crack passwords in the other variant. MneSchEx and MneYanEx are excluded in the evaluation, as in the two variants, a word is not always converted into one character. The percentage of passwords cracked when varying the number of guesses is illustrated in Figure 2(a). For all of the variants, we can crack 60% of the passwords within 10 guesses, where most of them are in the first 5 attempts.

The method performs less effective on MnePer and MneEx. From Table 6, we can observe that the percentages of unique conversions from a word to a character contribute to 32.8% and 31.3% of all such conversions in MnePer and MneEx. The quantities are much higher than those from MneGenEx (22.1%) and MnePerEx (24.1%). The more unique conversions leads to more character mappings that are never guessed. Table 6 also shows that participants in MneEx and MnePer are more likely to use digits, symbols, and upper-case letters than participants in MneGenEx and MnePerEx. One likely explanation is that just a single example is presented in MneGenEx and MnePerEx; while no example is presented in MnePer and six examples are given for MneEx. It is possible that both no example and lots of examples cause people to be more creative in mapping words to characters.

We adapt our method to be case-sensitive when guessing as follows. The training process is identical to the case-insensitive condition. When generating password guesses, every time a password (with the highest probability) is generated, instead of 1 guess, 4 guesses are made. We try the original password, capitalize all letters, capitalize the first letter, and capitalize all letters whose corresponding words are capitalized. The performance of the method on case-sensitive passwords is shown in Figure 2(b). More than 50% of passwords in all the 4 variants can be guessed in 20 attempts, with most successes from the first 10 guesses.

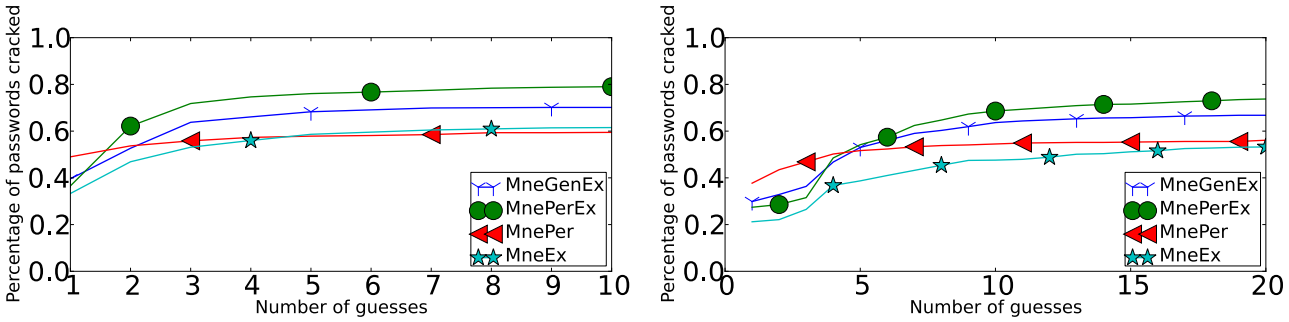
Cracking from scratch. Now, we apply the cracking method described above to a real-world scenario, in which sentence selection in the testing dataset is unknown. Given a training dataset, we generate candidate passwords as follows. We first order the sentences selected in the training dataset by the descending order of their frequencies. Then, starting from the most popular sentence, for each sentence, we generate 20 case-sensitive guesses.

Fig 3 shows the effect of our method evaluated on the four datasets by cross validation, i.e., for each testing dataset, the training dataset is the union of the other three datasets. In the graphs, each curve represents a cracking method, and a point (x, y) on the curve means y percentage of passwords in the testing dataset are cracked within x attempts. We also plot the curves of 5-order Markov Model (MC_5), PCFG method (PCFG) trained on Rockyou dataset, and two blacklist-based methods, which use Rockyou (Rockyou) dataset and the passwords in the training datasets (Train), respectively. Because of the limited number of sentences in the training dataset (less than 2400), we are able to generate less than 50,000 candidate passwords using the new method, and used 50,000 as the number of passwords generated for all methods.

The evaluation on MneGenEx is illustrated in Fig 3(a). From the figure, we can observe that all the generic cracking methods perform poorly, and can crack no more than 0.4% passwords within 50,000 guesses. In fact, the only passwords covered by the methods are "!@#$%^" and "!@#$%^&*", which are apparently created without following the strategy. On the other hand, 3.2% of the passwords in MneGenEx are covered in the 211 passwords in the training datasets, which confirm the need to using strategy-specific methods. Our proposed method can crack 6.4% passwords with 50,000 guesses. We expect performance of the method will increase with the size of training data. The performance of our method as well as the dictionary obtained from the training dataset drops significantly on the other datasets, and passwords from MnePerEx appears to the strongest. Less than 1% passwords in MnePerEx are cracked with 50,000 guesses. The reduced per-

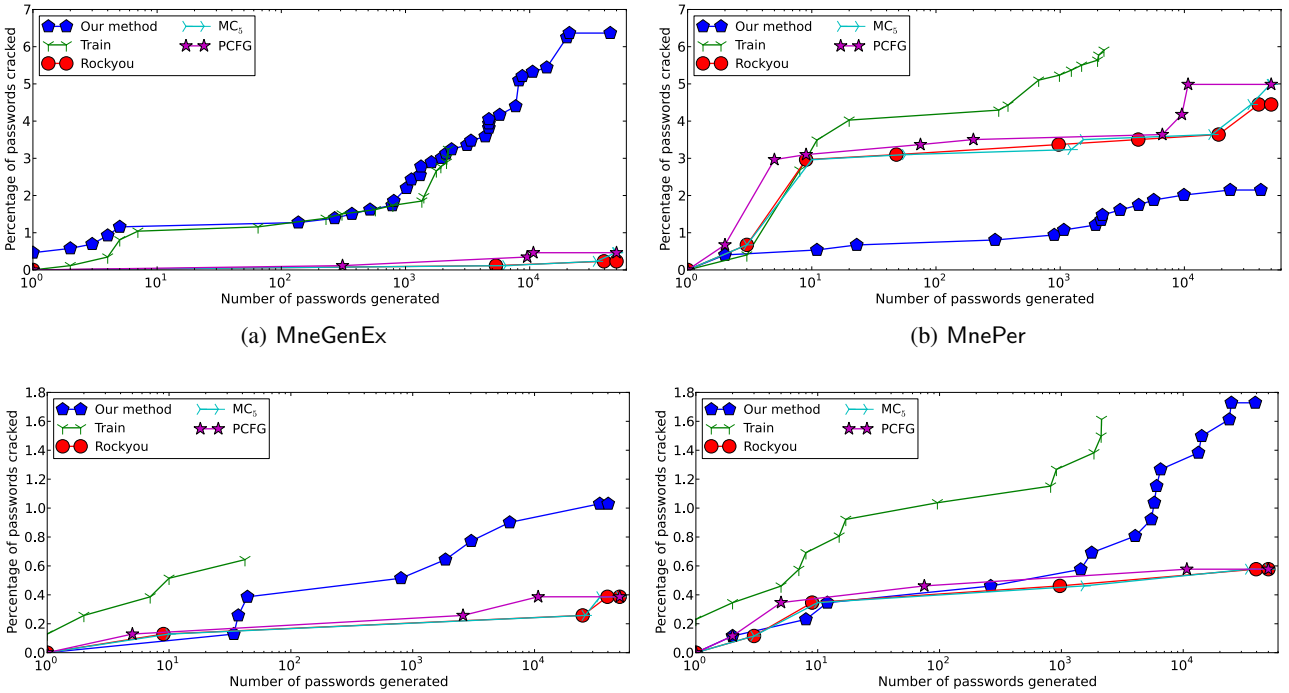
Table 6: Character usage in mnemonic strategy variants. Upper, Lower, Digit and Symbol means the number of corresponding type of character used in passwords. First means the number of words whose first character is directly used in the password. First + Leet means the number of words whose first character or the Leet substitution of the first character is used in the password. Total Trans means the total number of pairs of word and resulted characters. Unique Trans means the number of word-character pairs that only appear once in the dataset. Distinct Words are the number of distinct words used in all sentences.

Variant	Upper	Lower	Digit	Symbol	First	First+Leet	Total Trans	Unique Trans	Distinct Words
MneGenEx	$683_{8.6\%}$	$6073_{76.4\%}$	$792_{10.0\%}$	$399_{5.0\%}$	$6633_{83.5\%}$	$194_{2.4\%}$	7947	$1758_{22.1\%}$	2034
MnePerEx	$559_{7.8\%}$	$5465_{76.7\%}$	$741_{10.4\%}$	$364_{5.1\%}$	$6080_{85.3\%}$	$174_{2.4\%}$	7129	$1718_{24.1\%}$	1954
MnePer	$817_{12.0\%}$	$4290_{63.2\%}$	$1123_{16.6\%}$	$554_{8.2\%}$	$5027_{74.1\%}$	$332_{4.9\%}$	6784	$2228_{32.8\%}$	2334
MneEx	$994_{12.2\%}$	$5539_{68.1\%}$	$1046_{12.9\%}$	$553_{6.8\%}$	$6651_{81.8\%}$	$277_{3.4\%}$	8132	$2547_{31.3\%}$	2207



(a) case-insensitive (b) case-sensitive

Figure 2: Percentage of passwords cracked within 10 attempts for case-insensitive passwords, and 20 attempts for case-sensitive passwords.



(a) MneGenEx (b) MnePer

(c) MnePerEx (d) MneEx

Figure 3: Guess number graph on passwords created by using the mnemonic strategy.

formance of the methods is mainly due to the requirement of personalized sentence choice and the resulted increasing number of unique sentences. The result is consistent with the findings from $\tilde{\lambda}_1$ and $\tilde{\lambda}_{10}$ analysis described in Section 4.2.

One may also notice that in MnePer, the relative order of the methods is quite different. This is because of high frequency of passwords generated not following the strategy, such as *12345678* (17), *!@#$%^&** (6), *123456789* (5). These passwords

are hard to predict based on our method, but are easy to guess based on the other methods. As a result, our method performs worse than all the other methods in the graph.

5. STUDY 2: USABILITY

We conducted another user study evaluating the usability of the mnemonic strategy from two aspects: (I) Creation usability: Time and effort required from the user to generate a password by following the given strategy; (II) Memorability: recall of the password generated with the given strategy about 1 week later. In this study, three variants were evaluated, MneGenEx, MnePerEx, which were evaluated as the most and the least secure mnemonic variant in the previous analysis, and Control, which serves as a baseline.

Time used for password generation, the success rate of password recall, and password recall time were measured. We also examined the effort that participants spent during password generation and recall utilizing the NASA-Task Load Index (TLX) [18], which has been widely used in human factors research to assess the perceived workload during a task. In the NASA-TLX, workload is rated on 6 subscales: mental demand, physical demand, temporal demand, performance, effort and frustration. Participants rated the workload of each subscale ranging on a linear scale from 0 to 20, where 0 means very low workload and 20 means very high workload.

5.1 Study Overview

This study was conducted on MTurk in two phases. The first phase was similar to the previous experiment except as noted. At the beginning, we explicitly told participants that they would be asked to return and use the password in about one week, and they could take whatever measures they would normally take to remember and protect the passwords. Also, a concrete creation scenario was provided to simulate a real-world password generation context. Specifically, each participant was asked to create an online account for a bank named "Provident Citizens Bank". One variant randomly selected from Control, MneGenEx, and MnePerEx was assigned to each participant for password generation. For participants using MneGenEx and MnePerEx, the sentence creation and password generation were separated into two pages, such that the created sentence was not visible to participants during password generation, in order to mimic the password generation environment in practice. Participants were allowed to arbitrarily switch back and forth between the two pages. After the password generation, each participant was asked to measure the workload spent on creating the passwords utilizing the NASA-Task Load Index. About half of the participants were randomly selected to recall the password that they had just created at the end of the study, to evaluate the impact of short-term retrieval on password recall 1 week later.

Participants were invited back for the second phase by email. We sent the invitation emails through MTurk starting from the 6th day after participants finished the first phase. For those participants who did not come back to the study, we re-sent the same invitation email for another two days. In the second phase, participants were instructed to login to "Provident Citizens Bank" with the password they created and then to update the password. Each participant was allowed up to 4 attempts until failure. If a participant could not recall the password within the first 2 attempts, the strategy was displayed as a hint. Regardless of the performance in the login process, all participants were asked to evaluate the workload during password recall by using the NASA-TLX afterwards.

Table 7 lists the general statistics of the study. In the first phase, for each condition, we list the number of participants, average password creation time, and statistics for short-term password recall (if applicable) including success rate before and after seeing the strategy as a hint, failure rate after 4 attempts, and time used in password recall. In the second phase, we list the number (percentage) of participants that returned to the study; statistics about long-term password recall, including the number (percentage) of participants who did not write down passwords; the success/failure rate and average time used in password recall for those who did not write passwords down; the number (percentage) of participants who used the strategy provided to update their passwords.

5.2 First Phase Results

We recruited 224, 250, and 278 participants for Control, MneGenEx, and MnePerEx, accordingly, with a total of 752 (346 females). The participants' ages ranged from 18 to over 50, with 76% between 23 to 50 years. Most participants were college students or professionals who had bachelor or higher degrees.

Password creation time. The average time used in password creation for each variant is listed in Table 7. The password creation time was significantly different among the three conditions. As expected, participants spent the least time when there was no restriction (Control), and time spent in Control is significantly less than that MneGenEx and MnePerEx ($p < 0.001$). Compared with MneGenEx, password generation time was shorter in MnePerEx ($t = 2.45, p = 0.014$). That's mainly due to the additional requirement for personalized choice that narrowed down the search space of sentences and resulted in a faster decision.

Workload. Fig 4(a) shows the average ratings in each subscale of NASA-TLX for the three variants. Overall, the perceived workload was relatively low, with the average ratings for all subscales being below or close to 10. The workloads required in Control were lower than that from the two mnemonic strategy variants ($p < 0.001$). There was no significant difference between MneGenEx and MnePerEx ($p = 0.101$).

Short-term recall. About half of the participants in each variant were asked to recall the password at the end of the first phase. From Table 7, we can observe that regardless of the strategy used, almost all participants could enter the correct password.

5.3 Second Phase Results

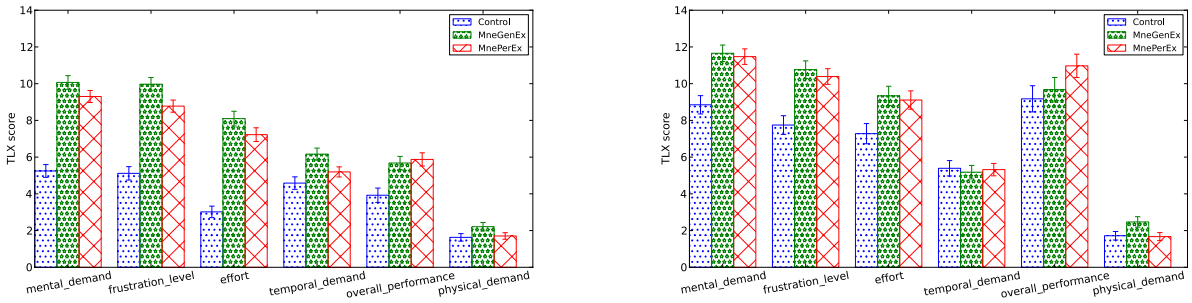
Approximately around 70% of participants from each condition returned to the second phase of the study.

Long-term recall. For participants who came back for the study after 1 week, we asked them whether they had written down the password after password recall process and explicitly told them that the answer did not affect payment to reduce any intention of deceiving. Approximately 80% indicated that they did not write down the password or the sentence (in MneGenEx and MnePerEx) and the ratio from the three conditions is similar, indicating that participants using mnemonic strategy variants were as confident as those in the control group that they could remember the passwords. We analyzed the password memorability from the participants who claimed that they did not write down passwords or sentences.

About 38% of participants recalled the passwords successfully within first two attempts, and an extra 7% of participants were able to recall the passwords when the strategy was displayed as a hint. The final successful recall rate did not differ significantly among the conditions ($\chi^2_{(2)} = 3.237, p = .198$). When there was a short-term recall, the long-term recall success rates were generally increased for each condition, which is in agreement with previous

Table 7: Statistics for usability study. Succ1 means the the number of participants who successfully recall the password within 2 attempts. Succ2 means the the number of participants who successfully recall the password in the third or fourth attempts, and the strategy was displayed as a hint. No WDP means the number of participants who did not write down passwords. Time is measured in second.

Variant	Short Term Recall	Phase 1						Phase 2						
		Count	Creation Time	Short-term Recall				Number Returned	Long-term Recall					Update
				Succ1	Succ2	Failed	Time		No WDP	Succ1	Succ2	Failed	Time	Use Strategy
Control	Yes	111	38.4	108(97%)	0(0%)	3(3%)	27.0	84(76%)	66(79%)	25(38%)	5(8%)	36(55%)	41.1	N/A
	No	113	41.1	N/A	N/A	N/A	N/A	82(73%)	63(77%)	23(37%)	4(6%)	36(57%)	47.5	N/A
	All	224	39.8	N/A	N/A	N/A	N/A	166(74%)	129(78%)	48(37%)	9(7%)	72(56%)	44.3	N/A
MneGenEx	Yes	114	170.0	107(94%)	5(4%)	2(2%)	31.0	91(80%)	72(79%)	39(54%)	2(3%)	31(43%)	70.2	67(69%)
	No	136	143.1	N/A	N/A	N/A	N/A	91(67%)	80(88%)	26(32%)	9(11%)	45(56%)	105.3	59(65%)
	All	250	155.4	N/A	N/A	N/A	N/A	182(73%)	152(84%)	65(43%)	11(7%)	76(50%)	88.7	126(69%)
MnePerEx	Yes	146	126.0	140(96%)	1(1%)	5(3%)	31.5	107(73%)	90(84%)	30(33%)	6(7%)	54(60%)	139.3	78(73%)
	No	132	139.7	N/A	N/A	N/A	N/A	94(71%)	75(80%)	25(33%)	5(7%)	45(60%)	91.8	67(71%)
	All	278	132.5	N/A	N/A	N/A	N/A	201(72%)	165(82%)	55(33%)	11(7%)	99(60%)	117.7	145(72%)



(a) Password Generation

(b) Password Recall

Figure 4: Mean scores of TLX as a function of strategy and subscale for the three conditions. Error bars represent standard errors of the scores.

finding [39]. And the increase in rates was larger for the mnemonic strategy, especially for the MneGenEx variant.

Long-term recall time. The password recall time in Control was shorter than that for MneGenEx ($t = 3.24, p = 0.001$) or MnePerEx ($t = 1.80, p = 0.073$), whereas there was no significant difference between MneGenEx and MnePerEx ($t = 0.73, p = 0.46$). Whether or not short-term recall had been required did not have any significant impacts.

Workload. The workload of password recall evaluated by TLX is illustrated in Fig 4(b). Comparing Fig 4(a) and Fig 4(b), perhaps the most noticeable difference is that the subscale of performance in Fig 4(b) is almost double that in Fig 4(a), which was mainly due to large portion of failed recall. Except physical demand and temporal demand, which are not directly related to the task, the average rates of all the other 3 subscales also increased dramatically, suggesting that password recall was more difficult than password generation.

For the subscales, mental workload and frustration ratings of mnemonics strategy variants were higher than those of the Control, and no significant difference was observed between MneGenEx and MnePerEx, which is consistent with the first phase results, suggesting the overhead from the extra requirements of the mnemonic strategy.

Password update. At the end of the task, we asked participants to update the password, without any restriction except that the password could not be the same as the old one. For MneGenEx and MnePerEx, after the password was created, we asked participants whether they used the strategy we provided. About 70% of participants said "yes" to the question, and the percentage for MnePerEx was slightly higher. The results indicated that most of the participants were willing to use the instructed strategy even if not forced to do so.

Overall, the study suggests that although workload required for the mnemonic strategy variants is significantly larger than that for Control, no significant difference in password recall between mnemonic strategy variants and the control group is observed, which is consistent with the previous literature [44, 45]. MnePerEx, which shows advantage over MneGenEx in terms of security, performs similar to MneGenEx in all the measurements regarding usability.

6. DISCUSSION

In this section, we discuss the consequence of our security evaluation findings and also present our study limitations.

6.1 Impact of Security Assessment

We observed that the security of the mnemonic strategy is highly sensitive to the exact instructions and examples presented to the user. We showed that the generic and commonly suggested instructions and examples resulted in high β-guess-rates in both sentence choices and the resulted passwords. As a result, if the mnemonic strategy with generic instructions and examples is widely adopted by a large population, the resulting passwords are not likely to be stronger than the baseline passwords (*i.e.*, passwords created by users without following any particular strategy), which are considered to be weak and predictable. If an adversary is equipped with a specially-designed cracking mechanism, such as the one presented in Section 4.3, he will still be able to break into a large number of password-protected user accounts within a limited number of attempts. We also observed that explicitly requiring personalized

choices of sentences as well as providing good personalized examples significantly enhance the security of the strategy. Notably, in our study, we witnessed that no sentences or passwords were chosen more than twice in MnePerEx. Even though with a larger dataset we may observe duplicated passwords, the number of password repetition, however, is likely to be significantly lower compared to the baseline.

Our findings suggest the following recommendation: *when offering instructions for teaching the mnemonic sentence based password generation strategy, we recommend including the additional requirement of personalized choice of sentences as well as concrete example(s).* More generally, one should pay attention to the exact wording of instructions for describing other password generation strategies, and for other messages aiming to communicate security-relevant messages. Furthermore, using examples is an important aspect of such communication, as apparently a portion of the users do not follow what appear to be straightforward instructions without concrete examples.

The fact that passwords generated under MneGenEx, using the generic version of the instructions, are considered to be strong under standard cracking methods, even though they contain a high level of collisions, suggests that previous password studies that use cracking as the only method to assess strength of passwords created under different conditions (e.g., under different composition policies) are limited. It would be interesting to revisit some of the studies using statistical metrics.

6.2 Study Limitations

Ecological validity. We conducted our study with the MTurk population, which is more diverse than the participants in typical laboratory studies [13]. The use of MTurk also allowed us to recruit more participants and collect data from a larger and diverse population, which is hard to achieve in a traditional in-person data collection study. The downside is that the participants' behavior in the simulated study setting may be different from real-world scenarios. This is a concern shared by all studies that use MTurk.

Why two separate studies. In this paper, the evaluation of the security and memorability of the mnemonic strategy were carried out in two separate studies. When we were designing Study 1, we were principally interested in the security of the resulting passwords, in part because we believed that the usability of the mnemonic strategy had been convincingly established by previous studies, e.g., Yan et al.'s [44, 45] study, which have participants actually using passwords created under the strategy. We thus did not ask participants to come back after the study to assess the longer-term memorability of the passwords. Only after Study 1 had been underway did we realize that the precise instructions play an important role in the level of security, and then the natural research question of whether different instructions would also affect the usability comes up. We thus carried out Study 2 to compare MnePerEx, the condition best for security, with MneGenEx and a control condition.

To make the situation in Study 2 to be as close to real-world situations as possible, we made a few changes in Study 2 compared with Study 1. First, in Study 2 participants were presented with information about a fictitious bank and asked to create a password for that bank. Second, in Study 2, while participants were still asked to type in the sentence they were using for creating the passwords (in part to help ensure that they were following the strategy), when they entered the password, the sentence was hidden from view. Third, participants were told that they would be asked to return in one week for another study during which they would be asked to recall the password. Due to these differences, we believe that these two studies should be viewed as separate from each other, and data should be compared only with those from the same study. Study 1 assessed the security of 6 variants, and Study 2 assessed the usability, including creation mental workload and memorability, of 2 of them. Furthermore, any difference should affect all groups in one study equally, and should not affect comparisons between groups within each study.

One potential concern is that because password creation in Study 1 was not set up in a way that exactly reflects a real-world password creation scenario, the created passwords might not reflect what the users would have used in real-world scenarios. We acknowledge that such concerns are valid, although we would argue that the main conclusions comparing the security under different variants remain meaningful. The instructions are explicit and appear to be new to most of the participants. Thus what passwords the participants would create should be mostly affected by the instructions, and the additional influence of whether they were told that they would come back in one week is likely to be limited, especially because the instructions already explicitly ask participants to choose memorable sentences.

Low password recall rates. In Study 2, we observed a seemingly high failure rates in password recall in all groups (about 56% in the control group). We believe that several factors contributed to this failure rate. First, we asked participants not to use their current passwords because we will record them, and it is likely that the passwords generated in the studies were freshly created. Second, participants did not use this password for one week, before they were asked to recall it. It is unclear to us whether the failure rate we have observed is indeed excessively high under such conditions. Imagine that a user created a fresh password that is unrelated to her existing passwords for a website, and was then asked to login at the site after one week without any usage of the password in between. What would be the failure rate in such a situation? It is likely to be quite high, although how high it is is an open question. We are unaware of documented failure rates in such situations.

Intuitively, most password-based authentication systems in real world do not have such a high failure rate as observed in our study. Some possible reasons are (a) most users use existing passwords (or simple variants) for a new account, (b) some users rely on password managers (e.g., those provided by browsers), (c) some sites are visited more frequently immediately after the accounts are created (thus users have more frequent rehearsal).

It is well known that memorability of passwords is highly correlated with the frequency of password use. For instance, the positive impact of repeated login of secrets (*e.g.*, passwords) has been demonstrated in a study by Bonneau and Schechter [10]: Following a rigorous but carefully designed login schedule (*i.e.*, spaced repetition), a large number of study participants (*i.e.*, 88%) were able to effectively recall an encoded random 56-bit binary string. Furthermore, to promote successful recall of passwords, some password management strategies have built-in rehearsal schedules [5, 6].

In an online study conducted on MTurk by Shay *et al.* [34], the password retention rate was about 80%. However, their study has two differences from our Study 2. First, in [34] participants were not explicitly asked to create fresh passwords, and were simply asked to create passwords that satisfy certain composition policies. Also, the temporal distance between password recall and password creation in [34] was 2 days, while ours is about 1 week. Empirical evidence from memory research robustly suggests that long-term memory declines over time [42].

In summary, the absolute numbers of the recall failure rates may not be very informative. However, since the conditions are equiv-

alent for all groups, our results regarding the between-group comparisons should still be valid.

Writing passwords down. In Study 2, whether passwords had been written down was based on participants' response to our verification question, and we are relying on the honesty of the participants. It is unclear whether more reliable verification methods exist. Although we cannot verify users' responses, we asked the question after the password recall phase and explicitly pointed out that their answer would not affect payment to remove any incentive of deceiving. Of course we can never rule out the possibility that some users wrote down their passwords and then lied, which would mean that the real password recall rates are even lower than what we have observed. Again, any impact of deceiving should be similar for all groups.

Management of passwords for multiple accounts. In a real-life setting, a user is likely required to create and recall passwords for multiple accounts. Password reuse is often considered bad practice and discouraged in advices for passwords. Creating and managing multiple strong passwords, however, yield a phenomenon for the users called the *password overload*. Password overload alludes to the users' inherent inability to successfully recall passwords for multiple accounts due to *memory interference* (*i.e.*, failure to recall an item that is similar to items stored in the memory) [4]. The effect of memory interference has also been observed in the context of graphical passwords [16]. In Study 2, we asked users to create and recall only one password. The study did not address the issue of memory interference. It is an intriguing open research question whether using the mnemonic sentence-based strategy would make remembering multiple passwords easier or harder.

7. CONCLUSION

In this paper, we investigated the security of 6 variants of the mnemonic password generation strategy. For two of them, we also evaluated memorability after 1 week. We showed that using the standard cracking-based methodology, password sets obtained under all variants have similar strengths and are all much more secure than the baseline. However, using β-guess-rates, we found that different instructions have a tremendous impact on the security level of the resulting passwords. In particular, the instructions for the mnemonic strategy found in the literature and recommended by security experts are not optimal at inducing secure password distributions. However, combining explicit instructions of choosing a personalized sentence that is unlikely to be chosen by others, with the inclusion of corresponding examples, dramatically increased the security of the resulting passwords, without observable negative impacts on usability.

8. ACKNOWLEDGEMENT

This paper is based upon work supported by the United States National Science Foundation under Grant No. 1314688.

We would also like to thank Yinqian Zhang, shepherd for this paper, and other reviewers for their helpful comments, which guided us revise and improve the paper.

9. REFERENCES

[1] Passwords, 2009. http://wiki.skullsecurity.org/Passwords.

[2] xkcd password generator, 2011. http://preshing.com/20110811/xkcd-password-generator.

[3] A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[4] M. C. Anderson and J. H. Neely. *Memory*, chapter Interference and inhibition in memory retrieval, pages 237–313. Academic Press, 1996.

[5] J. Blocki, M. Blum, and A. Datta. *Naturally Rehearsing Passwords*, pages 361–380. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] J. Blocki, S. Komanduri, L. F. Cranor, and A. Datta. Spaced repetition and mnemonics enable recall of multiple strong passwords. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, 2015.

[7] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 538–552. IEEE, 2012.

[8] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567. IEEE, 2012.

[9] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. Passwords and the evolution of imperfect authentication. *Commun. ACM*, 58(7):78–87, June 2015.

[10] J. Bonneau and S. Schechter. Towards reliable storage of 56-bit secrets in human memory. In *Proceedings of the 23rd USENIX Security Symposium*, August 2014.

[11] S. Boztas. Entropies, guessing, and cryptography. Technical Report 6, Department of Mathematics, Royal Melbourne Institute of Technology, 1999.

[12] S. Brostoff and M. A. Sasse. "ten strikes and you're out": Increasing the number of login attempts can improve password usability. In *HCISEC Workshop*, 2003.

[13] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

[14] S. Chiasson, P. C. van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, 2006.

[15] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley. Does my password go up to eleven?: The impact of password meters on password selection. In *Proceedings of CHI*, pages 2379–2388, 2013.

[16] K. M. Everitt, T. Bragin, J. Fogarty, and T. Kohno. A comprehensive study of frequency, interference, and training of multiple graphical passwords. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 889–898, New York, NY, USA, 2009. ACM.

[17] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle. Improving text passwords through persuasion. In *Proceedings of SOUPS*, pages 1–12, 2008.

[18] S. G. Hart and L. E. Staveland. *Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research*, volume 52, pages 139–183. Elsevier, 1988.

[19] C. Herley and P. C. van Oorschot. A research agenda acknowledging the persistence of passwords. *IEEE Security & Privacy*, 10(1):28–36, 2012.

[20] J. H. Huh, S. Oh, H. Kim, K. Beznosov, A. Mohan, and S. R. Rajagopalan. Surpass: System-initiated user-replaceable

passwords. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 170–181. ACM, 2015.

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

[22] S. Komanduri, R. Shay, L. F. Cranor, C. Herley, and S. Schechter. Telepathwords: Preventing weak passwords by reading users' minds. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 591–606, San Diego, CA, Aug. 2014. USENIX Association.

[23] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. Of passwords and people: measuring the effect of password-composition policies. In *CHI*, pages 2595–2604, 2011.

[24] C. Kuo, S. Romanosky, and L. F. Cranor. Human selection of mnemonic phrase-based passwords. In *Proceedings of the second symposium on Usable privacy and security*, pages 67–78. ACM, 2006.

[25] Z. Li, W. He, D. Akhawe, and D. Song. The emperor's new password manager: Security analysis of web-based password managers. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 465–479, Aug. 2014.

[26] J. Ma, W. Yang, M. Luo, and N. Li. A study of probabilistic password models. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 689–704. IEEE, 2014.

[27] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur. Measuring password guessability for an entire university. In *Proceedings of ACM CCS*, pages 173–186, Berlin, Germany, 2013. ACM.

[28] R. Morris and K. Thompson. Password security: A case history. *Communications of the ACM*, 22(11):594–597, 1979.

[29] K. Scarfone and M. Souppaya. Guide to enterprise password management (draft), Apr. 2009. NIST Special Publication 800-118 (Draft).

[30] S. Schechter, C. Herley, and M. Mitzenmacher. Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks. In *Proceedings of HotSec*, pages 1–8, 2010.

[31] B. Schneier. Passwords are not broken, but how we choose them sure is, Nov. 2008. The Guardian.

[32] B. Schneier. Choosing secure passwords, 2014. https://www.schneier.com/ blog/ archives/ 2014/ 03/ choosing _secure _1.html.

[33] R. Shay, L. Bauer, N. Christin, L. F. Cranor, A. Forget, S. Komanduri, M. L. Mazurek, W. Melicher, S. M. Segreti, and B. Ur. A spoonful of sugar? The impact of guidance and feedback on password-creation behavior. In *CHI'15: 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Apr. 2015.

[34] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor. Can long passwords be secure and usable? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2927–2936, New York, NY, USA, 2014. ACM.

[35] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor. Encountering stronger password requirements: user attitudes and behaviors. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, pages 2:1–2:20, New York, NY, USA, 2010. ACM.

[36] D. Silver, S. Jana, D. Boneh, E. Chen, and C. Jackson. Password managers: Attacks and defenses. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 449–464, Aug. 2014.

[37] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. How does your password measure up? the effect of strength meters on password creation. In *Proceedings of USENIX Security Symposium*, 2012.

[38] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay. Measuring real-world accuracies and biases in modeling password guessability. In *Proceedings of the 24th USENIX Security Symposium*. USENIX, Aug. 2015.

[39] K.-P. L. Vu, R. W. Proctor, A. Bhargav-Spantzel, B. Tai, J. Cook, and E. Eugene Schultz. Improving password security and memorability to protect personal and organizational information. *International Journal of Human-Computer Studies*, 65(8):744–757, 2007.

[40] K.-P. L. Vu, B.-L. B. Tai, A. Bhargav, E. E. Schultz, and R. W. Proctor. Promoting memorability and security of passwords through sentence generation. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 48, pages 1478–1482. SAGE Publications, 2004.

[41] D. Wheeler. zxcvbn: realistic password strength estimation. dropbox blog article (apr. 10, 2012).

[42] J. T. Wixted and E. B. Ebbesen. On the form of forgetting. *Psychological Science*, 2(6):409–515, 1991.

[43] L. Xing, X. Bai, T. Li, X. Wang, K. Chen, X. Liao, S.-M. Hu, and X. Han. Cracking app isolation on apple: Unauthorized cross-app resource access on mac os x and ios. In *Proceedings of the 22Nd ACM CCS*, pages 31–43, 2015.

[44] J. Yan, A. Blackwell, R. Anderson, and A. Grant. The memorability and security of passwords: some empirical results. *Technical Report-University Of Cambridge Computer Laboratory*, page 1, 2000.

[45] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, 2(5):25–31, Sept. 2004.

[46] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. In *Proceedings of ACM CCS*, pages 176–186, 2010.

[47] R. Zhao and C. Yue. All your browser-saved passwords could belong to us: A security analysis and a cloud-based new design. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*, pages 333–340, 2013.

[48] R. Zhao and C. Yue. Toward a secure and usable cloud-based password manager for web browsers. *Computers & Security*, 46:32–47, 2014.