Testing Computer-Aided Mnemonics and Feedback for Fast Memorization of High-Value Secrets

Sauvik Das and Jason Hong
Carnegie Mellon University
sauvik@cmu.edu and jasonh@cs.cmu.edu

Stuart Schechter
Microsoft Research
stuart.schechter@microsoft.com

***Abstract*—People sometimes require very strong passwords for high-value accounts (e.g., master passwords for password managers and encryption keys), but often cannot create these strong passwords. Assigning them provably strong secrets is one solution, and prior work has shown that people *can* learn these assigned secrets through rote learning, though learning the secrets takes some time and they are quickly forgotten after two weeks of disuse. To improve upon the learning speed and long-term recall of strong, assigned secrets, we introduce, implement and evaluate a set of treatments, inspired by mnemonic devices and real-time feedback tutoring systems, to assist people in learning and remembering their assigned secrets. We encoded strong secrets as a set of six words randomly chosen from a corpus of 676 (∼56 bits of entropy). In a randomized between-subjects experiment, our *story* mnemonic, in which participants wrote two sentences linking their assigned secret words together in a narrative, performed best. Participants who used the *story* mnemonic required significantly fewer training sessions (7.5 versus 12 sessions) and had higher two-week recall when allowing for minor errors (84% vs. 65%) than the rote control from prior work. Additionally, 92% of those who could not recall their full secrets after two weeks were able to recover their secret once they saw their mnemonic hints with the secret words elided. In contrast, our other treatments did not perform as well – providing few, if any, notable improvements over the rote control. Finally, in an exit survey, a large majority of our participants reported that our treatments were quick, helpful and enjoyable.**

I. Introduction

While user-chosen passwords or PINs are sufficiently strong for most applications, some high-stakes applications require people to remember provably strong secrets that will resist a quadrillion ($\approx 2^{50}$) guesses or more.

For example, consider a user carrying a laptop with an encrypted hard drive. If the user is captured, anything the user has (tokens) and all of the users' physical properties (biometrics) will be available to attackers. Alternatively, consider a user who wants to store all of her passwords using password-management software: She'll need to be able to get to the password database from different devices that may not all support additional authentication factors and will thus want her master password to be very strong. Additionally, consider whistle-blowers and reporters who want to protect the private components of their asymmetric keys by encrypting them with strong memorized secrets. For these users, there may be no alternative secure channel that they can trust for two-factor authentication. In all of these cases, users may benefit from learning one or more high-security secrets. All the better if they can do so reliably and with a reasonable amount of effort.

Despite conventional wisdom to the contrary, Bonneau and Sc3hechter recently demonstrated that lay people can indeed reliably learn strong (56-bit) passwords with a reasonable amount of effort [3]. In one treatment, the researchers encoded these secrets as ordered sequences of six words chosen from a dictionary of 676 (26^2) possible words. They had participants memorize these secrets through spaced repetition and without explicitly asking their participants to memorize the secrets. Rather, they simply showed participants their secret and asked them to re-enter it into a text-field. On subsequent attempts, they silently introduced increasing delays (from 0.3 to 10 seconds) before revealing the secret to be copied. Participants could avoid the delay by typing in their secrets from memory and nearly all participants eventually learned all of their secret words (94%). However, learning required a large number of training sessions (median 36) and recall fell sharply after two weeks of disuse (62%).

We hypothesized that by introducing mnemonics and providing feedback to make learning intentional we might reduce the number of training sessions required to memorize secrets and improve the recall and recoverability of the secrets after periods of extended disuse. To that end, in this paper, we designed and evaluated three new training regimens to teach people 56-bit secrets.

Our story mnemonic required participants to create two meaningful sentences, each of which would contain three words of their secret in order. Our peg-word mnemonic required participants to create a separate sentence for each secret word, each also containing a non-secret word to assist users in later recalling the sentence, and, in turn, the secret word. Finally, our feedback treatment did not include a mnemonic, but provided real-time feedback on areas for improvement to make rote learning more intentional.

In a randomized, between-subjects experiment with 351 participants from Amazon's Mechanical Turk, we evaluated these new treatments against two controls: a re-implementation of the rote-learning approach from prior work; and, a dropout-comparison control in which participants were not asked to

USEC '16, 21 February 2016, San Diego, CA, USA

http://dx.doi.org/10.14722/usec.2016.23010

learn a random password, so we could study whether the work incurred to memorize secrets caused participants to drop-out. Participants had to perform a distractor task that required them to login to our website 45 times, with at least one hour between adjacent logins. Each login required participants who were not assigned to the dropout-comparison control to enter a system-assigned password that they would progressively learn through one of the aforementioned methods. We then followed-up with participants to measure recall after 3+ days and 2+ weeks.

Our story treatment performed best. The median number of training sessions required to memorize the full 56-bit password was significantly lower for participants in our story treatment (7.5 sessions) than those in the rote control (12 sessions)—a 38% improvement. Furthermore, allowing for one adjacent word order swap (reducing entropy slightly to 53.8 bits), more participants in our story treatment (43/51, 84%) could recall their secrets after two-weeks of disuse than those in the rote-learning group (28/43, 65%). Also, the mnemonic hints in our story treatment offered a path to password recovery for those who forgot their secret words (12/13, or 92% recovered their secret). Surprisingly, however, our peg-word and feedback treatments showed little benefit. Finally, our participants appeared to find our approach enjoyable, quick and helpful.

Tersely, we offer the following contributions:

- The design and implementation of three novel computer-aided treatments to assist people in learning very strong (∼56-bit) secrets: a *story* mnemonic treatment, a *peg-word* mnemonic treatment and a real-time rote *feedback* treatment.
- A rigorous evaluation of these treatments as compared to the state-of-the-art from prior work [3], in a randomized-controlled experiment.

II. Background and Related Work

Prior work in cognitive psychology suggests that rote learning, as employed by Bonneau and Schechter [3], may not be as effective for long-term recall as techniques that promote the elaborative encoding of information, or actively relating new information to knowledge already in memory or more easily placed in memory [4], [6], [11]. Mnemonics, or strategies that enhance the learning and recall of information [1], are one way to promote the elaborative encoding of information and have been shown to help retain information [1], [5], [13].

Mnemonics work by translating abstract information, such as lists of numbers, into representations that are easier to remember, such as images. Concretely, there are two broad types of mnemonics: chain-type and peg-type [1]. Chain-type mnemonics work by creating logical chains between list items. For example, with story-based mnemonics, learners memorize a list of words by creating a sequential story between list items. Thus, to memorize the words "apple", "beetle", and "crane", one might imagine "a large apple, being eaten by a tiny beetle, is lifted by a crane".

Peg-type mnemonics provide the learner with a *cognitive cuing structure*, known as "pegs", to which the learner can associate the list items to be memorized. For example, with the peg-word mnemonic, learners first memorize an ordered set of "pegs", often generated through rhyme—such as one is nun, two is shoe, three is tree. When memorizing "apple", "beetle" and "crane", then, learners can picture a nun eating an apple, a shoe stomping on a beetle, and a crane lifting up a tree. Later, learners can recall the list words by remembering the peg rhymes—*e.g.*, "one is nun" should elicit the vision of a nun eating an apple, and, in turn, help people remember the list word: "apple".

Some prior work has looked at using mnemonics to help people memorize strong passwords. Curiously, however, much of this work offers users advice on *generating* a strong password with a mnemonic device, rather than *retaining* a provably strong password. Research on these co-generation strategies, in which passwords and mnemonics are created together, has outlined that many users incorrectly apply mnemonics in a manner that prevents any added security gains [7] or outright do not comply with using these strategies at all [21]. Indeed, past work has documented that, unsurprisingly, people tend to choose famous phrases that are well known and easily guessable in co-generating mnemonics and secrets to create mnemonic passwords (*e.g.*, "4s&7ya", coming from "Four scores and seven years ago") [12]. It is, thus, not surprising that passwords generated mnemonically are vulnerable to guessing attacks [12], [21].

Outsider of these co-generation strategies for passwords, however, mnemonics are typically used to learn existing information that existed prior to the creation of the mnemonic [1]. Accordingly, we should be able to maximize the retention of random secrets by encoding them into forms amenable to mnemonic construction—*i.e.*, generating the mnemonic from the secret to ensure that the secret is uninfluenced by the mnemonic. Pronounceable passwords [9] are an example of prior work that seeks to encode strong secrets into formats that are easier to remember. Fastwords [10], in which an ordered sequence of user-chosen dictionary words make up a user's random secret, is another example of prior work that tries to encode secrets into a memorable form. In addition, Blocki [2] explored using person-action-object three-tuples to assist users in remembering passwords (i.e., a person performing an action with an object, where the person, action and object are the password). However, Blocki does not provide security guarantees for these tuples and their memorability benefits remain unclear with less than a $\frac{1}{3}$ of participants, in some conditions, recalling their tuples after just one week.

In general, prior work on applying mnemonics to help users remember strong passwords has focused on usability, without quantifying security improvements and often without strictly assessing memorability improvements. In contrast, we designed and rigorously evaluated mnemonic treatments that quickly and reliably teach lay people secrets with provably strong security guarantees.

III. Method

A. Bonneau and Schechter's Experiment

To facilitate comparison with Bonneau and Schechter's state-of-the-art [3], our experimental design builds on theirs. They recruited participants to login and complete an attention test, designed to appear similar to tests of the Stroop effect [19], 90 times over 15 days. Participants in an experimental treatment were assigned a secret of six words, chosen randomly

from a corpus of 676. The researchers added a phase to this login flow in which participants had to enter at least part of their assigned secret.

The researchers divided the six-word secrets into three chunks of two words. Initially, they presented participants with only the first chunk (two words) of the code, which they rendered directly above the text-entry field into which participants were asked to type these words. Thus, participants simply had to copy the two-word code. At each subsequent login, the researchers added a third of a second *revelation delay* before they revealed the two words of the chunk above the input field, up to a maximum of 10 seconds. Participants could enter the words regardless of whether they had been rendered on screen. Each time a participant typed a new character of the secret correctly, the researchers reset the revelation delay. Resetting the revelation delay gave participants who recalled the chunks more time to type them. Once a participant entered the words of a chunk before the words were revealed, and did so three times in succession, the researchers included the next two-word chunk on every subsequent login. Each chunk had its own revelation delay based on the number of times it was exposed to users.

Three days after participants completed the 90 trials, the researchers asked participants to return to recall their secrets. They did so again after another two weeks.

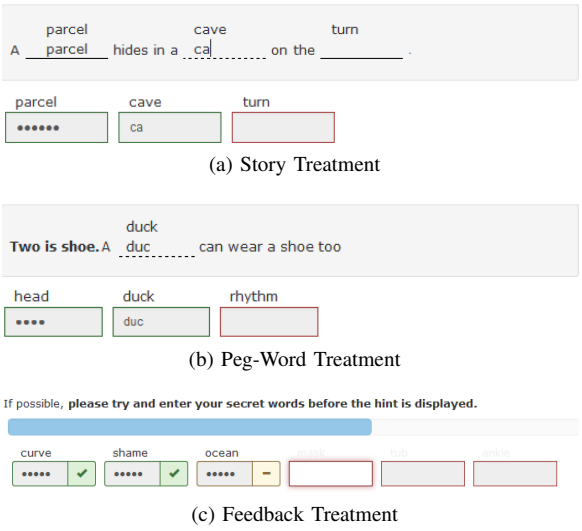
B. Changes to accommodate new hypotheses

While we were able to re-use the JavaScript for the attention test, testing our new hypotheses required us to partially change the experimental design and implement our own testing infrastructure.

First, to facilitate the use of mnemonics, we created a new 676-word corpus with only nouns, whereas the corpus used in prior work did not restrict parts of speech. We also divided participants' randomly generated passwords into two chunks of three secret words, instead of three chunks of two words, because Bonneau and Schechter found that some of their participants avoided learning their second chunk to avoid being assigned a third. Upon revealing the second chunk, we assured participants that they would not be assigned more.

As prior work had shown that most participants learned their full secret within 45 rehearsals, we reduced the total number of rehearsals from 90 to 45, increased the minimum interval between rehearsals from 30-minutes to an hour, and decreased the number of days in which to complete all sessions from 15 to 10. We also changed the revelation delay increments from $\frac{1}{3}$ of a second to $\frac{1}{2}$ of a second per exposure because we suspected that participants were learning in fewer sessions than prior work could detect—a suspicion supported by Schechter and Bonneau's follow-up work [17].

Whereas prior work asked participants to enter their full chunk in a single text field, this would have caused problems for our peg-word treatment, in which participants focused on one word per sentence. We thus created a single text field for each word, but auto-tabbed to the next field after a participant completed a word so that they could type the words as if they were entering all their secret words into a single text field.

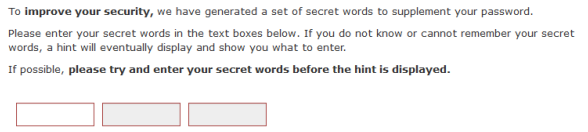


(a) Story Treatment

(b) Peg-Word Treatment

(c) Feedback Treatment

Fig. 1: Our three treatments. (a) Story: Users wrote a sentence linking three randomly assigned secret words; (b) Peg-Word: users wrote a sentence linking each secret word to a public peg that could later assist in recall (c) Feedback: Users were given feedback about progress.



(a) Participants had to enter their secret words into the empty text boxes.

(b) Input fields were highlighted green when participants entered the correct character.

Fig. 2: Rote control condition.

Finally, Bonneau and Schechter had presented this attention test as the sole focus of their study. However, they reported that many participants became wise to their ruse. Since we would introduce treatments that required conscious training, we chose to disclose that we were studying the process of learning the secret words in addition to the attention test. We retained the cognitively demanding attention test, however, to distract participants who would otherwise focus too intently on memorizing their secret words—an unrealistic situation not reflective of a real deployment.

C. Conditions

In addition to a dropout control in which participants did not have to learn any secret words, we implemented one rote memorization control emulating prior work and three treatments to inspire more active learning of the assigned secret words. The treatments varied in (i) mnemonic creation, (ii) rehearsal, and (iii) hint progression.

1) Rote [Baseline control]: First is our reimplementation of Bonneau and Schechter's word-based rote memorization

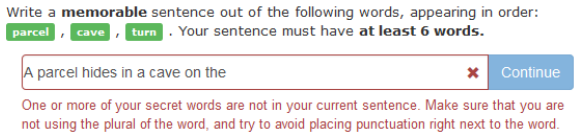
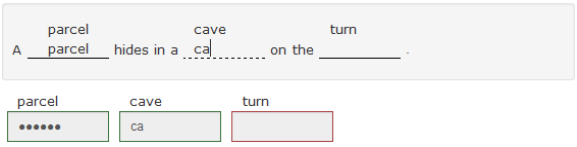


Fig. 3: The mnemonic creation phase of our story treatment. Participants had to write a sentence connecting the three secret words of their first chunk.



(a) The full-assistance rehearsal format allowed participants at an early stage of learning to enter their secret words inline.

(b) The reduced-assistance format obfuscated the hint sentence, but still allowed inline text entry, to allow participants at a later stage to rely more heavily on their own memory.

Fig. 4: For the story treatment, we used different presentation formats depending on the stage of learning

treatment. As participants did not have to create a mnemonic, they progressed directly to rehearsal.

Chunk rehearsal. The rehearsal initially consisted of three text boxes for entering the first three-word chunk, as shown in Figure 2a. We prevented participants from entering words out of order. After participants learned their first chunk (defined as successfully entering the first chunk three times in succession without the hints being revealed), we revealed a second set of three text boxes corresponding to their second and final chunk of secret words.

Hint Progression. The revelation delay increased at $\frac{1}{2}$ second increments with each login, up to a maximum of 10 seconds. In other words, for a participant's i^{th} attempt on a chunk, the revelation delay was $\min\left(\frac{i}{2}, 10\right)$ seconds.

2) Story [Treatment]: The story treatment, based on the story chain-type mnemonic [1], required participants to create a sentence for each three-word chunk, with the words in each chunk appearing within the sentence in order.

Mnemonic creation. On their first login session, we asked participants to write a memorable sentence by stringing together, in order, the three words of their first chunk. We also provided an example—for the words "cat", "leaf", and "wind", we offered the example "A pink cat is chasing a giant leaf blowing in the wind". We asked participants to create a visual sentence by embellishing upon and exaggerating words, guided by the understanding that exaggerated visual imagery is easier to remember [20]. We also provided feedback to participants as they wrote a sentence, informing them if they were missing a word or had not yet written six words (see Figure 3). After they completed writing their sentence, we asked participants to visualize the sentence for 10 seconds. To discourage them from ignoring this request, we presented a 10-second timer and did not allow them to continue until it expired. After participants learned their first chunk, we repeated this process for their second chunk.

Chunk rehearsal. We added an additional user-interface element to the rehearsal screen to facilitate the retrieval and entry of the secret words – the hint well – which can be seen just above the input fields in Figure 4. Our full-assistance rehearsal format rendered the user-created story sentence, with underlines in place of the secret words, in the hint well. Participants could type their secret words into the underlined blanks, as shown in Figure 4a. The inline text fields in the hint sentence were linked to the standard text fields shown below the hint well, so anything the user wrote in the inline text fields were copied into the standard text fields as well. When we revealed the secret words of a chunk, we placed them both above the inline text field and above a standard text field.

Our reduced-assistance format obfuscated the words of the story sentence by replacing them with solid black bars. We rendered the bars to be the same width as the words they replaced, as illustrated in Figure 4b. We still required participants to type their secret words inline. Our no-assistance format left the hint well empty, leaving only the three text fields from the rote treatment's rehearsal. The reason these fields appeared in the other two formats was to make the transition between formats less jarring.

Hint Progression. As reading the hint sentence takes time, we added an additional second to the revelation delay, so that for the i^{th} login, the delay was $\min\left(\frac{i+1}{2}, 10\right)$ seconds. So, for a participant's 10^{th} attempt on the chunk, the delay would be $\min\left(\frac{10+1}{2}, 10\right) = 5.5$ seconds.

When first asking the user to enter the words of a chunk, we used the full-assistance format. After a user first entered the chunk from memory, we transitioned to the reduced-assistance format and created a format-transition timer. The format-transition timer was set to be half the value, in seconds, of the number of exposures to the chunk in the reduced-assistance format. So, a participant would be in the reduced-assistance format for 0.5 seconds on her first exposure to the format, 1 second for her second exposure, 1.5 seconds for her third exposure and so on. If the format-transition timer expired before the participant could enter her secret words, we regressed her back to the full-assistance format and started the revelation delay timer.

Similarly, once a participant entered the chunk from memory with only reduced-assistance, we transitioned her to the no-assistance format. Again, we used a format-assistance timer set to be half the value, in seconds, of the number of exposures to the chunk using this format. If a participant could not enter the correct words before the timer expired, we regressed her back to the reduced-assistance format and started the format-transition timer for the reduced-assistance format. Notably, the cumulative sum of all three timers could not exceed 10 seconds, so participants would never need to wait more than 10 seconds before the secret words of a chunk were revealed.

Index	Choices
One	Nun, Bun, Gun, Sun
Two	Shoe, Shrew, Zoo, Screw
Three	Tree, Bee, Key, Sea
Four	Thor, Boar, Door, Shore
Five	Hive, Chive, Knives, Wives
Six	Bricks, Chicks, Sticks, Flicks

TABLE I: Peg-Word choices for all indices.

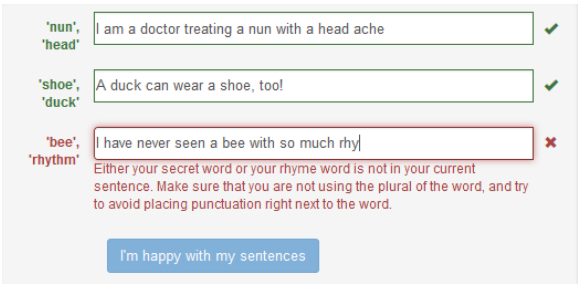


Fig. 5: The peg-word treatment required users to write sentences associating their peg and secret words.

3) Peg-Word [Treatment]: The peg-word treatment, based on the peg-word mnemonic [1], required participants to create three sentences for each chunk. Each sentence would contain one of the secret words from their chunk along with a peg-word: a word they would choose in advance of learning their secret words that they would try to later associate with it. Making participants select their pegs in advance ensured that the peg-words could be made public (to later assist with recall) without compromising the secret.

Mnemonic creation. We first asked participants to choose one of four possible peg-rhymes for each secret word index. The options are shown in Table I. After participants chose three peg words for the chunk, we showed them the three secret words in their chunk. We asked them to write three simple sentences containing both a peg word and secret word, in sequence, to help them associate the (public) peg word with the secret word. As with the story treatment, we instructed participants to make the sentence visual and memorable, and gave them the following example with the secret word "cat" and the peg "bun": "A pink cat is trapped in a giant hamburger bun". The process of entering sentences is captured in Figure 5.

Chunk rehearsal. The rehearsal process was similar to that for the story treatment, with three differences: (1) we showed only the sentence for one word, rather than one chunk, in the hint-well at any one time; (2) we showed the public peg-rhyme (e.g., "one is nun") regardless of format; and, (3) we provided peg-word specific instructions on how to retrieve one's secret words (i.e., we told them to visualize their peg-word in the context of the hint sentence they earlier created). As in the

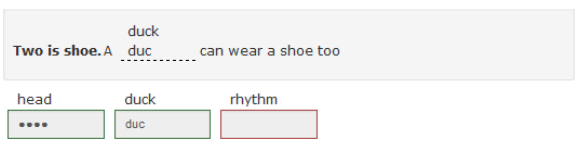


Fig. 6: Full-assistance rehearsal format for peg-word.

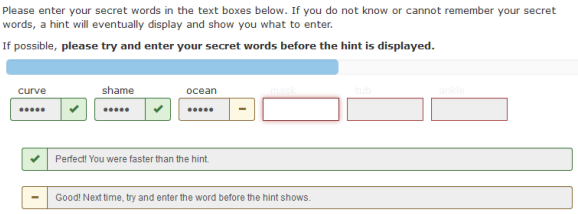


Fig. 7: Participants in the feedback treatment were shown the chunk hint timer and word-by-word feedback.

story treatment, participants in the full or reduced assistance format had to "fill in the blank" of their hint sentence by entering their secret words into the inline text-fields (Figure 6), but had to enter their secret words into the standard text-entries in the no-assistance format.

Hint Progression. We transitioned participants through the same three assistance formats with the same timers as the story treatment. While there were three sentences per chunk, timers continued to operate per chunk.

4) Feedback [Treatment]: Given the background literature on the potential benefits of real-time feedback on progress and areas for improvement on learning [14], [15], we decided test whether a less subtle conditioning process would impact learning speed and recall. This treatment was a variant of the rote baseline with additional feedback cues in the rehearsal phase. The training and hint progression remained exactly the same as the rote baseline.

Chunk rehearsal. We mirrored the rote baseline, but also rendered a progress slider indicating the time remaining until the secret words in a chunk would be revealed (see Figure 7). If a participant entered a correct character, they would see the slider start again from zero. In addition, when a participant entered a word before time ran out, we rendered a green check mark on the right edge of the input field for that word. If time ran out before a participant entered a secret word correctly, we placed a yellow minus sign on the right side of the input field. At the bottom of the page, we provided more specific feedback as to the meaning of these indicators. For the green check mark, we wrote: "Perfect! You were faster than the hint"; and, for the yellow minus sign we wrote: "Good! Next time, try and enter the word before the hint shows."

IV. Experimental Evaluation

Mirroring Bonneau and Schechter's study, we recruited workers on Mturk to complete the one-minute attention test for $0.40 and, upon completion, offered them the opportunity to join our extended study for a $19.00 bonus. The extended study required participants to create an account with our website. We paid participants for completing the HIT regardless of whether they joined the extended study, thus conforming to Amazon's policies which forbid requiring workers to create a website account to complete a HIT. We paid participants by adding a bonus to their HIT, thus ensuring we conformed to the spirit of Amazon's policies (Amazon could continue to charge us for facilitating the transaction).

This extended study required participants to complete the attention test 45 times over the course of 10 days, with at least

60 minutes in between neighboring sessions. As participants signed up for an account with our website, we randomly assigned them to either the dropout-control (10%) that required no additional authentication, the rote control (baseline, 22.5%), or one of the three treatments: story (22.5%), peg-word (22.5%), and feedback (22.5%). For all but the dropout-control condition, we explained that the extra step of entering secret words had been added to improve account security.

A. Completion Survey and Follow-Ups

Once participants finished their 45th and final session, we redirected them to a completion survey. For brevity, we omit the specific questions asked. Notably, we asked participants for demographic information such as age, gender, educational history and occupation.

After completing this survey, we asked participants if we could contact them in case we had any additional questions. If they agreed, we sent them an e-mail three days and two weeks after the end of the study for two follow-up experiments. In each follow-up experiment we asked participants to recall their six secret words. We did not render the secret words—if participants had forgotten one of their secret words, we instructed them to enter a "*" in the text field. However, we did show those in the peg-word group their public peg-word rhymes (Figure 8a), as these peg-words were generated independently and thus revealed nothing about the secret.

If a participant in the peg-word or story treatments failed to remember all of their secret words, we gave them a second opportunity with their hint sentences rendered (Figure 8b). Offering this hint reduces security, as the sentences were generated based on the secret words and thus may provide attackers a guessing advantage. Still, these hints may be an effective method for recovering a partially-forgotten secret that does not require users to remember several separate secrets, as is the case for using a password as the primary means of authentication and a challenge question for recovery.

V. Results

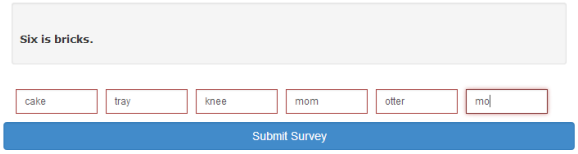
A. Descriptive Statistics

Out of the 450 people who completed our attention-test HIT, 351 (∼78%) created an account to be part of our study. Of those 351 participants, 242 (∼70%) completed the initial study—the 45 sessions and demographic survey. As we asked participants for demographic information at the end of our study (to reduce the barriers to joining), our demographic statistics reflect only those 242. They ranged in age from 18 to 68 (mean=31, sd=10). Reported genders were 127 (∼52%) female, 113 (∼47%) male, and two undisclosed (1%). Most participants (204) reported having some form of post-secondary education, and the two most frequent reported non-student occupations were service (27) and IT professionals (24). In addition, all but six participants reported English as their primary language.

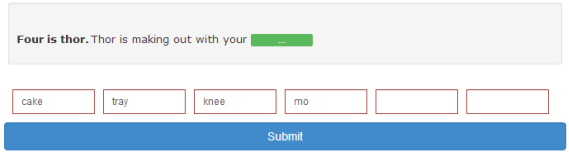
In Table II we present the number of participants who were assigned to each condition, completed the experiment, and completed each of the follow-up requests. Our control group had the highest dropout rate, but by a one-participant margin in a small sample. A Chi-Square test did not find

	Story	Peg	Feedback	Rote	Ctrl.
Participants	81	75	85	71	39
Completed	57	48	62	51	24
3-day return	54	45	61	48	N/A
2-week return	48	43	55	42	N/A

TABLE II: Participant assignment, completion, and follow-up returns across conditions.



(a) Without hint, (except for peg-words for those in peg-word treatment.



(b) With full hint except for the secret-words.

Fig. 8: Follow-up survey recall test results

significant differences in drop-outs across conditions [χ^2(4, N=51)=2.8, p=0.58]. Thus, we found no evidence to suggest that participants found our treatments too much of a burden to stop continuing with the study.

B. Learning Sessions & Learning Time

We hypothesized that participants in our story, peg-words, and feedback treatments should learn their secret words with fewer training sessions than those in the rote baseline. To test this hypothesis, we needed to define a concrete metric for learning. When a participant entered all three words of a chunk before the words were revealed for them to copy, we concluded that they did so from memory. When they first did so three times in succession, we concluded that they had learned their secret words prior to this three-session sequence (i.e., they demonstrated what they had learned by entering the words without seeing them). We refer to sessions preceding the three consecutive hint-free sessions as the learning sessions for that chunk and the remaining sessions as reinforcement sessions.

We then modeled the number of learning sessions required with a log-link Poisson regression [8], including a participant's treatment condition and age as covariates. We included participants' age as a control variable because prior work has shown that age correlates strongly with learning and memory [16]. Figure 9 shows the distribution of learning sessions across treatments and Table III shows the coefficients for our Poisson model. Coefficients represent the expected rate of change in a participant's predicted learning sessions for a one-unit increase in a numeric covariate, or a change from the baseline level of a categorical covariate to another level. Positive coefficients imply a positive correlation between covariate and response.

For example, the coefficient for the story treatment is $-0.36(p < 0.001)$, meaning that the expected difference in

the number of sessions required to learn both chunks for the story condition, relative to the rote control, is $e^{-0.36} = 0.70x$. Thus, a participant of average age assigned to story treatment is expected to learn his secret words in 30% fewer sessions than the same assigned to rote. Similarly, the coefficient for the peg-words treatment is $-0.19 (p = 0.003)$, suggesting that those in peg-words are expected to learn their treatment in $e^{-0.17} = 0.83x$ as many sessions as those in rote (i.e., 17% fewer). Thus, as we hypothesized, those in the story (median: 7.5) and peg-word (median: 9) treatments required fewer sessions to learn their assigned password than those in the rote condition (median: 12). However, we did not find convincing evidence to suggest that people in the feedback (median: 11) treatment required fewer learning sessions than the rote baseline (median: 12).

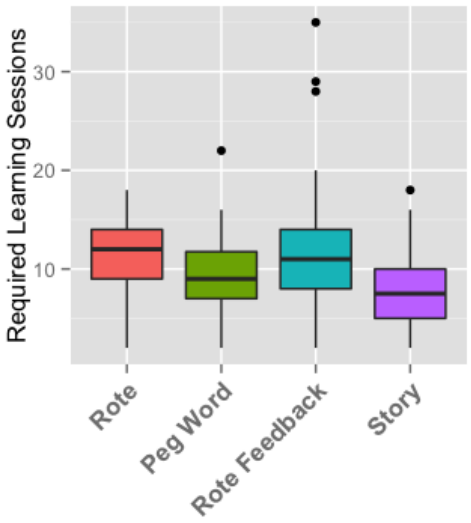
A related concern is the amount of learning time a participant requires—or the amounts of time participants have to spend in their learning sessions. To examine the magnitude of this learning time difference, we ran a post-hoc analysis. We calculated a participant's learning time by aggregating the amount of time a participant spent in their learning sessions. Figure 10 shows the average of this required learning time across all participants within a condition. A Kruskal-Wallis test revealed a significant effect of experimental condition on learning time $(\chi^2(3) = 53.4, p < 0.001)$. A series of post-hoc pairwise Mann-Whitney tests with Bonferroni correction showed significant differences between the learning times required for those in the story and rote conditions (means: 5.6 vs. 4.4 minutes, $p < 0.01$, $r = 0.30$) and between the peg word and rote conditions (means: 8.9 minutes vs. 4.4 minutes, $p < 0.001$, $r = 0.59$), but not between the feedback and rote conditions (means: 4.5 vs. 4.4 minutes, $p = 1.0$, $r = 0.04$).

Thus, participants in the story and peg-word treatments required 1.2 and 4.3 more minutes of learning time than those in the rote baseline, respectively. This result is expected given that the mnemonic treatments required an upfront time investment for creating the mnemonics. In a real-world implementation, reducing the number of learning sessions should justify a one-minute increase in learning time because the training sessions need to be private. Thus, the fewer training sessions needed, the quicker a user can use their strong secret words as their true password. Furthermore, spreading out an extra 1 minute of training over 7.5 training sessions results in an increase of only ∼10 seconds per session.

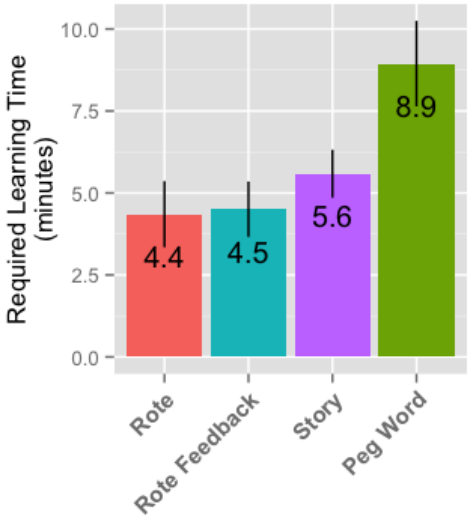
To summarize, relative to the rote baseline, we found that participants in the story and peg-word treatments required significantly fewer sessions to learn their secret words, though this expectedly came at the expense of some additional learning time. Participants in the feedback treatment, however, did not significantly differ from those in the rote baseline in learning sessions or learning time.

C. 2-Week Follow-Up Recall Rates

While we sent out a three-day follow-up to maintain methodological consistency with Bonneau and Schechter's original design, we were primarily interested in testing how our treatments affected recall after two weeks of disuse. We hypothesized that our new treatments should outperform the rote baseline in recall rate at the 2-week followup. More



(a) Number of training sessions required



(b) Mean training time required

Fig. 9: Cross-treatment boxplots of training sessions and training time to learn each chunk (with 95% confidence intervals).

Variable	Coefficient	p-value	
Condition: Peg Word	−0.19	0.003	*
Condition: Feedback	0.07	0.207	
Condition: Story	−0.36	< 0.001	**
Age	0.05	0.021	*
Intercept	2.43	< 0.001	**

Conditions are vs. baseline (Rote), ** $p < 0.001$, * $p < 0.05$

TABLE III: Coefficients for the log-link Poisson Regression modeling the effect of our experimental treatments on the number of training sessions required to learn the assigned password. Negative coefficients imply faster learning.

	Story			Peg Word			Feedback			Rote		
Recall Model	*Success*	*Fail*	*Rate*	*Success*	*Fail*	*Rate*	*Success*	*Fail*	*Rate*	*Success*	*Fail*	*Rate*
Perfect	38	13	75%	18	25	42%	27	29	48%	26	17	60%
Single Swap	43	8	84%	20	23	47%	31	25	55%	28	15	65%
Forgot One Word	43	8	84%	29	14	67%	37	19	66%	32	11	74%
Relaxed Order	44	7	86%	20	23	47%	31	25	55%	28	15	65%

TABLE IV: Follow-up 2-week recall rates across recall models and experimental condition. The story condition performed best.

specifically, we were interested in several levels of recall that each provided strong security guarantees, with the weaker levels accounting for small, predictable errors in memory:

Perfect recall (56.4 bits). One should remember all six secret words in the correct order. This level provides the strongest security guarantee at $\log_2\left(676^6\right) = 56.4$ bits of entropy.

Single swap (53.8 bits). One may swap the order of a single pair of two adjacent words. For example, one participant entered "cart", "hen", "fang" instead of the correct "cart", "fang", "hen". This is a relatively innocuous error that can be fixed at cost of $\log_2(6) = 2.6$ bits of entropy, as there are six valid orderings of the six secret words where any two adjacent words can be swapped. Allowing for these errors reduces password strength to 53.8 bits—still very strong.

Forgot one word (49.6 bits). One may forget a single secret word, but must remember the other five in the correct order. We can fix this error at a cost of 6.4 bits of entropy, as there are 676^5 combinations of 5 correct words times 6 places to insert an incorrect word, yielding $\log_2\left(676^5 \cdot 6\right) = 49.6$ bits.

Relaxed order (46.9 bits). One may enter her secret words in any order, but must remember all of them. We can fix this error at a cost of $\log_2(6!) = 9.5$ bits of entropy, reducing the strength of the learned secret words to a still strong 46.9 bits.

Table IV shows how participants in each condition performed at the 2-week follow-up. Notably, the story treatment performed best with 75% (38/51) perfect recall, and up to 86% (44/51) recall if we relax order constraints. For the rote baseline treatment, perfect recall rates (26/43, or 60%) were similar to those observed in Bonneau and Schechter's prior work (62%). Surprisingly, the peg-word and feedback treatments performed worse than the baseline rote treatment at all recall levels.

To analyze whether these differences in 2-week recall were significant, we modeled whether participants' could remember their secret words using a logistic regression. As with the previous analysis, we included a participant's experimental condition and age as covariates. In addition, since participants were free to complete the follow-ups at any time after we sent them the invitation, we included the number of days since a participant's last training session as an additional control covariate. In practice, most participants returned immediately after the invitation (the median return time after we sent out the invite was 5 hours). We also included a control variable for whether or not participants were shown their full hint-sentences in the three-day follow-up because it is possible that revealing their hint sentences afforded these participants an unfair advantage in reinforcing their secret words. Only ten participants were shown their hint sentences at the 3-day follow-up, nine of who returned for the 2-week follow-up.

We constructed a separate model for each recall level. We included only participant's first attempts at recalling their secret words in which we provided no hints—just six empty text boxes. Table V shows the coefficients for these models. The coefficients in Table 5 represent a change in *log-odds*, or $\ln\frac{P}{1-P}$, where P represents the probability that a follow-up response was correct (i.e., that the participant correctly entered her six secret words at a particular level of recall). A positive coefficient implies that P increases with the covariate. A negative coefficient implies the opposite.

The story condition did best. Allowing for order errors either single swap ($b = 1.09, p = 0.04$) or relaxed order ($b = 1.25, p = 0.02$) errors participants in the story treatment had significantly higher recall than the rote baseline. For the perfect ($b = 0.65, p = 0.165$) and forgot one word ($b = 0.65, p = 0.22$) recall levels, however, the improvement of the story treatment was not significant. Given the large effect sizes of the regression coefficients and the modest sample size, however, we would certainly not conclude they were insignificant without further study. Nevertheless, in practice, allowing for order errors does not substantially reduce security—indeed, allowing for single swaps still affords a very strong 53.8 bit secret that 84% of participants in the story treatment could remember after a formidable two weeks of disuse.

Surprisingly, only a minority of participants in both the feedback (27/56, or 48%) and peg-words treatments (18/43, or 42%) could perfectly recall their secret words after 2-weeks of disuse. Even at other levels of recall, these treatments underperformed the rote baseline. However, as the non-significant coefficient values in Table V indicate, these differences fell short of significance. The poor performance of the feedback condition suggests that the feedback mechanisms we chose might have been more distracting than helpful. If exploring other forms of feedback that might be more successful, we would likely strive to make them unobtrusive.

The poor performance of the peg-word mnemonic suggests that the memorization of six mnemonic sentences, even with a public cue, may be too much for people to reliably internalize with little training on internalizing the mnemonic hint sentences themselves.

D. Secret Word Recovery With Hints

If participants in the story and peg-word treatments could not perfectly recall their secret words in the follow-ups, we examined whether they could remember their secret words if shown their mnemonic hint sentences (Figure 8b). After two weeks of disuse, 46% (6/13) of hint-assisted retries in the story treatment and 44% (11/25) of hint-assisted retries in the peg-word treatment were successful. Including these retries yields a cumulative perfect recall rate of 86% (44/51) for the story treatment and 67% (29/43) for the peg-word treatment. In other words, it appears that many participants in our mnemonic treatments who made recall errors in the

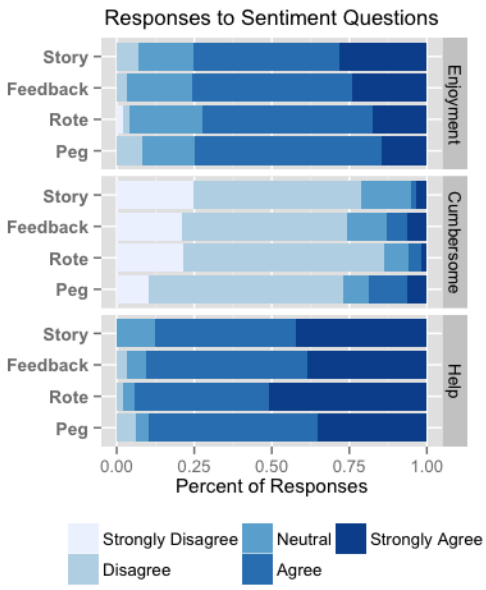


Fig. 10: Responses to sentiment questions in exit survey.

2-week follow-up were able to recover their secret words if shown their mnemonic hints. This result has two implications.

First, it is likely that we can improve the 2-week perfect recall rates of our mnemonic treatments if we help participants internalize their mnemonic hints *in addition to* their secret words. Indeed, in the story condition, the median number of exposures to either the first or second-chunk mnemonics was 7. For the peg-word treatment, the median number of exposures was 9.5 and 8 for the first and second chunk mnemonics, respectively. Participants who could not recall their secret words after 45 exposures are unlikely to recall a mnemonic to which they had so few exposures.

Second, it appears as though our mnemonic treatments offer a natural path towards automated password recovery. Showing participants their mnemonic hints and allowing for the aforementioned errors (order errors or forgetting one word), 12/13 (92%) participants in our story treatment and 20/25 (80%) participants in our peg-word treatment were able to recover their secret words after initially forgetting them. Of course, revealing the mnemonic sentences has its risks—users choose sentences to fit the words they are assigned and attackers may be able to guess the secret words from the sentence. However, given that other recovery mechanisms (e.g., password hints and challenge questions) also impose great risks [18], retaining hint sentences for automated password recovery may be a viable use case.

E. Post-hoc Sentiment Analysis

In all of our use cases, user creation of mnemonics to assist the memory of a strong secret would be optional. Most use cases are those in which users are invested in choosing a strong secret. Still, even in situations such as ours in which users are given a strong incentive to learn a secret (a gratuity), they still might not enjoy or appreciate doing so—especially if they

	Relaxed Order		Forgot One		Single Swap		Perfect
Condition: Peg Word	**-0.76**		**-0.27**		**-0.77**		**-0.71**
Condition: Feedback	**-0.49**		**-0.45**		**-0.49**		**-0.57**
Condition: Story	**1.26**	*	**0.65**		**1.09**	*	**0.65**
Age	-0.29		-0.20		-0.29		-0.23
Days Since Last Training	-0.07		-0.04		-0.06		0.01
Shown 3-day Hint?	-0.90		-0.98		-0.85		-1.06
Intercept	0.78	*	1.17	*	0.77	*	0.48

Conditions are vs. baseline (Rote), * p ¡ 0.05

TABLE V: Coefficients for the logistic regressions predicting whether or not a recall to the 2-week follow-up was successful.

perceive the training as boring or cumbersome. We asked our participants a number of questions to gauge their sentiments towards the learning process. Prior to asking the questions, we reminded participants that their answers would have no impact on their payment, so they should be frank. For brevity, and because these results are post-hoc and not key results, we keep this discussion short and omit some details. A summary of the results can be seen in Figure 10.

First, we asked participants to rate their agreement, on a five-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree", with the statement: *"I enjoyed the training provided to help me learn my secret words."* A large majority of participants in all conditions (over 75% of all participants in every condition) answered either "Agree" or "Strongly Agree" to this statement. Second, we asked participants to rate their agreement, on the same 5-point Likert scale, with the statement: *"The training provided to help me learn my secret words was too slow or cumbersome. (note: not including the attention test)"*. A large majority of participants (around 75% for each condition) answered "Strongly Disagree" or "Disagree" to this question. Finally, we asked participants to rate their agreement, on the same 5-point Likert scale, with the statement: *"The training provided for the secret words (e.g., the delayed hints, incremental chunks) helped me learn them."* We again found that the overwhelming majority of participants believed the training was helpful—not a single participant in the story treatment disagreed, and around 90% of all participants answered "Agree" or "Strongly Agree".

In sum, while these responses should be viewed with some skepticism (as participants may not want to state negative opinions), we have some empirical evidence to suggest that participants found our system to be enjoyable and quick.

VI. Discussion

To summarize, we ran a randomized, controlled experiment with 351 participants to test if mnemonic construction and real-time feedback would be more effective than rote memorization in teaching people strong secrets.

We found that participants in our story and peg-word treatments required significantly fewer learning sessions to learn their secrets. While this improvement came at the cost of requiring slightly more training time, in an actual deployment, reducing the number of learning sessions is more important as these sessions should be done privately.

In terms of long-term recall, we found that participants in our story treatment were most successful at recalling their se-

cret words after two weeks of disuse. Specifically, allowing for single swap errors (marginally reducing entropy from 56.4 bits to 53.8 bits), participants in the story condition significantly outperformed the rote baseline (84% vs. 65%) in two-week recall. Furthermore, 12/13 participants in the story treatment who initially forgot their secret words successfully recovering them when shown their hint sentences. Thus, only 1 of 51 participants in our story treatment could not recall or recover his secret words. Surprisingly, we found that our peg-word and feedback treatments provided little benefit in two-week recall. The peg-word treatment, however, also did provide a stable path to secret word recovery with hint sentences.

Finally, we found that participants found our approach enjoyable and helpful without being cumbersome. Indeed, many expressed a desire to use our approach in practice.

A. Limitations and Future Work

Real-world implementation: One open question is how should an incremental training scheme be implemented in practice? When using high-value secrets, such as a master password for a password manager, training should occur on a single machine before the database is encrypted to be shared between devices. Likewise, when using high-value secrets to encrypt storage drives, users will again want to be sure to complete training before putting high-value information onto the device or drive. Finally, some services may not accept passwords in the form of six secret words. However, it may still be possible to use our approach if learning a master password for a password manager that handles all other credentials.

Multiple-password interference: It remains unclear if our approach can be used to learn multiple passwords simultaneously–users may confuse one set of mnemonics with another. Future work should focus on answering this question as well as the question of teaching participants their mnemonic hints. For now, we envision our approach as a way to train users to learn one or a small set of strong passwords for especially important applications like a password manager.

Mnemonic reinforcement: It should be possible to improve the story treatment by reinforcing the mnemonic sentence during rehearsal. For example, on each of the first 15 exposures to a chunk, we might require the user to not only enter the three secret words in each chunk, but also one other randomly selected word from elsewhere in the sentence. This might cause the user to learn the whole mnemonic sentence during the mnemonic-reinforcement period.

VII. Conclusion

When users need to learn a high-security secret while minimizing the period (number of logins) until mastery and maximizing retention of the secret through extended periods of disuse, guiding users through the process of building a *story* mnemonic achieves significant improvements over subconscious rote learning. This improvement does not appear to be caused by the increased intentionality that results when users invest cognitive effort to create an mnemonic, as the more-intentional feedback treatment showed no improvement over rote learning. Story mnemonics can also create hints that may be used to help users recover secrets—but that may weaken security if used. Finally, many participants reported enjoying the learning process and expressed interest in using their learned secret words for high-value applications.

Acknowledgements

This work was done as a part of a summer internship at Microsoft Research. We would like to thank Jaeyeon Jung, Mona Haraty and Erik Harpstead for their helpful feedback.

References

[1] F. S. Bellezza, "Mnemonic Devices: Classification, Characteristics, and Criteria," *Rev. Educational Research*, no. 2, pp. 247–275, jan.

[2] J. Blocki, "Usable Human Authentication : A Quantitative Treatment Thesis Committee :," Ph.D. dissertation, Carnegie Mellon University, 2014.

[3] J. Bonneau and S. Schechter, "Towards reliable storage of 56-bit secrets in human memory," in *USENIX Security*, Aug. 2014.

[4] G. Bradshaw and J. Anderson, "Elaborative encoding as an explanation of levels of processing," *Journal of Verbal Learning and Verbal Behavior*, pp. 165–174.

[5] D. P. Bryant, M. Goodwin, B. R. Bryant, and K. Higgins, "Vocabulary Instruction for Students with Learning Disabilities: A Review of the Research," *Learning Disability Quarterly*, no. 2, p. 117.

[6] J. H. Coane, "Retrieval practice and elaborative encoding benefit memory in younger and older adults," *Journal of Applied Research in Memory and Cognition*, no. 2, pp. 95–100, jun.

[7] A. Forget, S. Chiasson, and R. Biddle, "Helping Users Create Better Passwords : Is this the right approach ?" in *Proc. SOUPS'07*, 2007, pp. 151–152.

[8] W. Gardner, E. P. Mulvey, and E. C. Shaw, "Regression analyses of counts and rates: Poisson, overdispersed Poisson, and negative binomial models." *Psychological Bulletin*, no. 3, pp. 392–404.

[9] M. Gasser, "A random word generator for pronounceable passwords," in *Mitre Corp Tech Report MTR-3006*, 1975.

[10] M. Jakobsson and R. Akavipat, "Rethinking Passwords to Adapt to Constrained Keyboards," in *Proc. IEEE MoST*, 2012.

[11] J. D. Karpicke and M. a. Smith, "Separate mnemonic effects of retrieval practice and elaborative encoding," *Journal of Memory and Language*, no. 1, pp. 17–29, jul.

[12] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in *Proc. SOUPS'06*. New York, New York, USA: ACM Press, p. 67.

[13] D. R. Lyon, "Individual Differences in Immediate Serial Recall: A Matter of Mnemonics?" *Cognitive Psychology*, vol. 9, pp. 403–411, 1977.

[14] M. J. Nathan, "Knowledge and Situational Feedback in a Learning Environment For Algebra Story Problem Solving," *Interactive Learning Environments*, vol. 5, pp. 135–159, 1998.

[15] J. Psotka, L. D. Massey, and S. Mutter, *Intelligent Tutoring Systems: Lessons Learned*. Psychology Press, 1988.

[16] T. A. Salthouse, "When does age-related cognitive decline begin?" *Neurobiology of aging*, no. 4, pp. 507–14, apr.

[17] S. Schechter and J. Bonneau, "Learning assigned secrets for unlocking mobile devices," in *Proceedings of the Symposium On Usable Privacy and Security (SOUPS)*, July 2015.

[18] S. Schechter, A. Brush, and S. Egelman, "It's no Secret: Measuring the Security and Reliability of authentication via 'secret' questions," in *Proc. S&P'09*. Ieee, pp. 375–390.

[19] J. R. Stroop, "Studies of interference in serial verbal reactions," *Journal of Experimental Psychology*, no. 6, pp. 643–662.

[20] H. von Restorff, "Über die Wirkung von Bereichsbildungen im Spurenfeld," *Psychologische Forschung*, no. 1, pp. 299–342, dec.

[21] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password Memorability and Security: Empirical Results," *IEEE Security & Privacy Magazine*, pp. 25–31, 2004.