A Large-scale Analysis of the Mnemonic Password Advice

Johannes Kiesel Benno Stein Stefan Lucks
Bauhaus-Universität Weimar
<first name>.<last name>@uni-weimar.de

***Abstract*—How to choose a strong but still easily memorable password? An often recommended advice is to memorize a random sentence (the mnemonic) and to concatenate the words' initials: a so-called mnemonic password. The paper in hand analyzes the effectiveness of this advice—in terms of the obtained password strength—and sheds light on various related aspects. While it is infeasible to obtain a sufficiently large sample of human-chosen mnemonics, the password strength depends only on the distribution of certain character probabilities. We provide several pieces of evidence that these character probabilities are approximately the same for human-chosen mnemonics and sentences from a web crawl and exploit this connection for our analyses. The presented analyses are independent of cracking software, avoid privacy concerns, and allow full control over the details of how passwords are generated from sentences. In particular, the paper introduces the following original research contributions: (1) construction of one of the largest corpora of human-chosen mnemonics, (2) construction of two web sentence corpora from the 27.3 TB ClueWeb12 web crawl, (3) demonstration of the suitability of web sentences as substitutes for mnemonics in password strength analyses, (4) improved estimation of password probabilities by position-dependent language models, and (5) analysis of the obtained password strength using web sentence samples of different sentence complexity and using 18 generation rules for mnemonic password construction.**

Our findings include both expected and less expected results, among others: mnemonic passwords from lowercase letters only provide comparable strength to mnemonic passwords that exploit the 7-bit visible ASCII character set, less complex mnemonics reduce password strength in offline scenarios by less than expected, and longer mnemonic passwords provide more security in an offline but not necessarily in an online scenario. When compared to passwords generated by uniform sampling from a dictionary, distributions of mnemonic passwords can reach the same strength against offline attacks with less characters.

I. Introduction

Password authentication is widely accepted, has low technical requirements, and hence is expected to stay as a part of authentication systems [4], [16]. Irrespective their popularity, password authentication has always been criticised for the fact that users tend to choose weak passwords—simply to avoid the extra effort of memorizing strong passwords. To animate users to devise stronger passwords, so-called mnemonic passwords are often recommended, which shall provide both strength and memorability [14], [31], [41]. Such advice boils down to the following:

> *Create a sentence. Memorize it. Concatenate the first characters of each word. Use the string as password.*

The strength of mnemonic passwords is based on three assumptions. First, humans can easily remember their mnemonics, a fact that has been shown within several studies [26], [41]. Second, it is infeasible to guess a mnemonic, even if an adversary was able to generate and test millions of guesses per second. This can be assumed, if the user in fact follows the advice and creates the mnemonic himself instead of picking a famous sentence [26]. Third, and most importantly, the derived passwords inherit most of the guessing difficulty of the mnemonic, so that guessing the password remains infeasible as well. To the best of our knowledge, regarding the last point no results have been published in the relevant literature.

This paper contributes various new and interesting results in this regard. Our approach is to generate passwords from a huge sample of human-generated sentences using a generation rule (a variant of "concatenate the first characters of each word"), estimate the resulting password distribution with language models, and calculate common strength estimates from the distribution. The contributions in detail:

- We collect one of the largest available corpora of human-chosen mnemonics (Section III-A).
- We extract a total of 3.1 billion web sentences from the ClueWeb12 crawl [36] with a specialized filter algorithm (Section III-B), show that these sentences are more complex than mnemonics using a standard readability score, and take a sample with appropriate sentence complexity (Section III-C).
- We use the corpus of mnemonics to provide evidence that the distributions of the character probabilities which are used by common password strength measures are approximately the same for mnemonics and web sentences (both all and the less complex sample). This allows us to substitute web sentences for mnemonics (Section III-D).
- To model mnemonic password distributions we optimize language models. For this, we introduce position-dependent language models to password modeling, for which we show that they improve the estimation over regular language models (Section IV-B).

NDSS '17, 26 February - 1 March 2017, San Diego, CA, USA
Copyright 2017 Internet Society, ISBN 1-1891562-46-0
http://dx.doi.org/10.14722/ndss.2017.23077

- Using common password strength measures that cover both online and offline attack scenarios (Section IV-C), we compare the strengths of password distributions from all and only the simpler sentences under 18 different password-generation rules (Section V).

Our approach comes along with a number of important advantages. It is fully reproducible since it uses a static web crawl. It exploits the knowledge of the password generation to its full effect, which makes the strength estimates more reliable compared to estimates obtained from dictionary-based cracking attempts. It causes no privacy concerns since no private authentication data is involved. It allows to precisely compare password generation rules, such as concatenating the words' last characters instead of the first.

II. Related Work

Different to existing studies we do not analyze a password corpus, but put a well-known[1] generation principle for passwords to the test.

Mnemonic password strength analyses have previously focused on cracking them by using dictionary or brute-force attacks [41] or a collection of quotes, lyrics, and similar known phrases [26]. However, these analyses are based on very small sample sizes (see Table I), the results depend largely on the employed cracking dictionaries, and they leave the exact generation process to the participants. Also, the used mnemonics are not available. It is interesting to note that Kuo et al. find that, if not explicitly forbidden, users tend to choose famous sentences as mnemonics, with the—expected—negative impact on security.

Very recently, Yang et al. [42] published a strength analysis on what they call mnemonic-based strategy variants, which are variations of the "*create a sentence*" part of the mnemonic password advice. They find that the security against online attacks can be increased when suggesting to the users to use personalized mnemonics and providing them an example mnemonic and password. In contrast, we analyze the security for different variants of generating the password from the sentence. Furthermore, since we use a much larger sample of passwords, we can also estimate the strength of mnemonic passwords against offline attacks and our estimates against online attacks are more robust.

Password strength analysis in general used way larger password samples (up to 70 million [3]), but do not distinguish between mnemonic passwords and others. Especially interesting is the analysis by Bonneau, who found differences in password strength between different user groups (determined by account settings) [3]. Our current data does not provide this kind of meta information.

An overview of the cracking methods used in these analyses is presented by Dell'Amico et al. [9]. Language models, which we use for our analysis, are also used in password cracking [9], [27], [30], [34]. Presumably, these password crackers would also benefit from our contribution of position-dependent language models.

Table I. Number of mnemonics and passwords in the corpora of this and other studies. For the corpora of this study, the number of passwords is averaged over generation rules.

Corpus	#Mnemonics	#Passwords
Webis-Sentences-17	3 369 618 811	1 381 862 722
Webis-Simple-Sentences-17	471 085 690	234 106 405
Webis-Mnemonics-17	1 048	1 035
Obfuscated Yahoo! passwords [3]	-	70 000 000
Leaked from RockYou (e.g., [39])	-	32 000 000
University passwords [28]	-	44 000
Phished from MySpace (e.g., [9])	-	34 000
Survey by Kelley et al. [21]	-	12 000
Survey by Yang et al. [42]	5 334	6 236
Survey by Kuo et al. [26]	140	290
Creation advised by Yan et al. [41]	97	290
Received from Passware [30]	-	140
Survey by Vu et al. [38]	40	40

Extending the usual mnemonic password advice, Topkara et al. [37] suggest complex generation rules to create passwords very different to the mnemonic. This allows to produce from the same mnemonic somewhat independent passwords with different generation rules, which aims at reducing password-reuse between services. Our estimates could also be calculated for such rules.

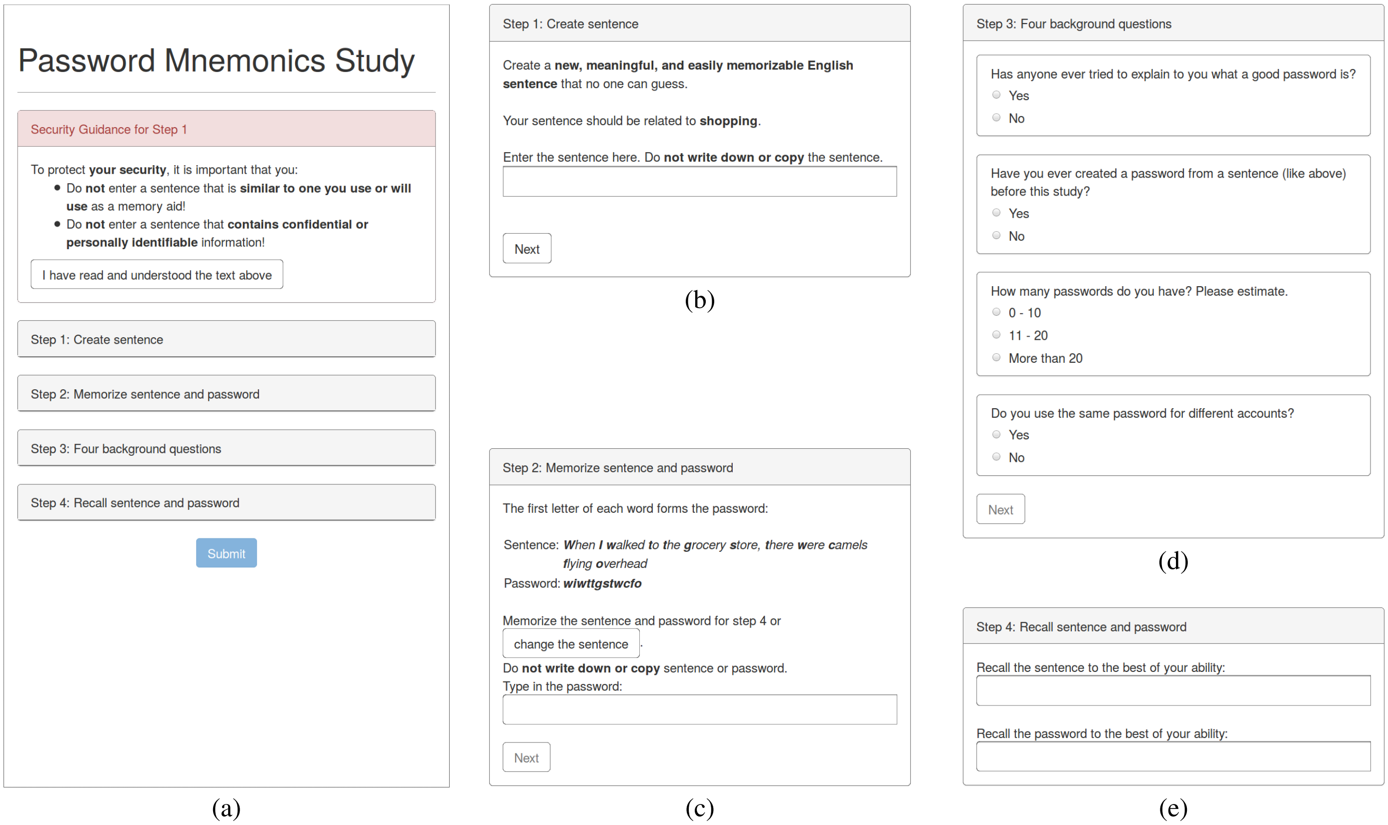
The good memorability of human-chosen mnemonics has been shown by previous studies. For example, Yan et al. found that mnemonic passwords are about as memorable as passwords selected freely but with at least one non-letter [41]. As memorability measure, they used the time needed until the passwords—which the 290 participants had to use frequently—are memorized. Random passwords, on the other hand, took about 8 times as long to remember.

A different approach to mnemonic passwords is to generate the mnemonics for the users using either sentence templates and dictionaries [1], [20], linguistic transformations [19], or language models [13]. While this removes the problem of humans choosing weak mnemonics, it is unclear how this changes the memorability compared to human-chosen mnemonics.

III. Sentence Corpora Acquisition

The analysis of a password advice requires a huge sample of the random element of that advice. In the case of the mnemonic password advice, the random element is the mnemonic. Section III-A introduces the new Webis-Mnemonics-17 corpus, which now is the largest corpus of human-chosen password mnemonics, but which is still far too small for a well-founded statistical analysis. Hence this section introduces also two new corpora of web sentences: the Webis-Sentences-17 corpus (Section III-B), as well as a subset called the Webis-Simple-Sentences-17 corpus whose overall sentence complexity better fits that of password mnemonics (Section III-C). Section III-D demonstrates that mnemonics and web sentences, though different, are very similar in the distributions of character probabilities which are relevant for estimating the password strength. With this knowledge, we can then estimate the strength of mnemonic passwords using the web sentence corpora.

[1] For example in a 2011 survey of 195 university people, about 40% had already used a mnemonic password [25].



Password Mnemonics Study

Security Guidance for Step 1

To protect **your security**, it is important that you:

- Do **not** enter a sentence that is **similar to one you use or will use** as a memory aid!
- Do **not** enter a sentence that **contains confidential or personally identifiable** information!

I have read and understood the text above

Step 1: Create sentence

Step 2: Memorize sentence and password

Step 3: Four background questions

Step 4: Recall sentence and password

Submit

(a)

Step 1: Create sentence

Create a **new, meaningful, and easily memorizable English sentence** that no one can guess.

Your sentence should be related to **shopping**.

Enter the sentence here. Do **not write down or copy** the sentence.

Next

(b)

Step 2: Memorize sentence and password

The first letter of each word forms the password:

Sentence: ***W**hen **I** **w**alked **t**o **t**he **g**rocery **s**tore, **t**here **w**ere **c**amels **f**lying **o**verhead*

Password: ***iwttgstwcfo***

Memorize the sentence and password for step 4 or change the sentence.

Do **not write down or copy** sentence or password.

Type in the password:

Next

(c)

Step 3: Four background questions

Has anyone ever tried to explain to you what a good password is?

○ Yes

○ No

Have you ever created a password from a sentence (like above) before this study?

○ Yes

○ No

How many passwords do you have? Please estimate.

○ 0 - 10

○ 11 - 20

○ More than 20

Do you use the same password for different accounts?

○ Yes

○ No

Next

(d)

Step 4: Recall sentence and password

Recall the sentence to the best of your ability:

Recall the password to the best of your ability:

(e)

Figure 1. The HTML interface used to collect the Webis-Mnemonics-17 corpus (Section III-A). (a) Complete interface at the survey start. Participants have to read a security guidance. After that, the steps (b-e) are shown one at a time. (b) Participants have to enter a sentence that fulfills our requirements (automatically checked, Section III-A). (c) Participants see their sentence and the corresponding password and are told to memorize both. They have to type in the password. Should they try to paste the password, the pasting fails and they are told not to do so. They can go back to step 1 to choose another sentence. (d) Participants have to select one option for each question. (e) Participants are asked to recall sentence and password.

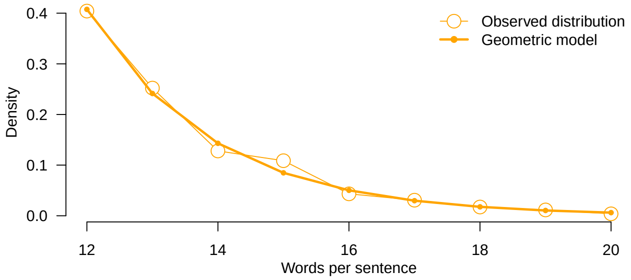


Figure 2. Distribution of sentence lengths in the Mnemonics Survey corpus and fitted geometric model.

A. The Webis-Mnemonics-17 Corpus

With the aid of the crowd-sourcing platform Amazon Mechanical Turk, 1 117 mnemonics were collected in a short survey, each from a different worker. Figure 1 shows the study interface. The workers are told to chose a mnemonic and remember it (without writing or copying) while answering password-related multiple-choice questions. The study has been designed to fulfill best practices for Mechanical Turk user studies [23]. For example, encouraging a participation in good faith by disabling copy-and-paste. The workers took on average 3 minutes and 35 seconds to complete the study.[2]

Instead of trying to reproduce the memorability results of previous research (cf. Section II), we opted for a shorter study with more participants.

In detail, the workers were asked to “create a new, meaningful, and easily memorable English sentence that no one can guess.” To resemble the advice of choosing the mnemonic related to the web page for which it is used (e.g., [14]), we randomly showed one topic suggestion (*money*, *shopping*, *mail*, *talking with friends*, or no suggestion) to the workers. The survey interface automatically enforced certain constraints to mirror plausible password requirements: The mnemonic must contain (1) only 7-bit ASCII characters; (2) at least 12 words; (3) at least 9 different words from an English dictionary (to ensure English mnemonics); and (4) no sequence of 6 or more words that also occurs in the Webis-Sentences-17 (detailed below, like a blacklist of known phrases).

After manual cleaning, 1 048 mnemonics remain. In detail, we rejected 17 workers that submitted grammatically incorrect mnemonics, and filtered mnemonics that were inherently meaningless (10), contained several phrases (40) or a known phrase missed by our filter (1 mnemonic), and where the interface did not record correctly (1 mnemonic). As Figure 2 shows, the length of the remaining mnemonics follows a geometric distribution, which is similar to password length distributions in general [27]. Table III gives a few examples from the corpus for each suggested topic.

[2] The corpus with detailed interaction logs of the workers is available at www.uni-weimar.de/en/media/chairs/webis/corpora/webis-mnemonics-17

Despite being one of the largest available corpora of human-chosen mnemonics for password generation, the Webis-Mnemonics-17 corpus is still too small for the calculation of theoretical password strength estimates. Such strength estimates rely on the probability distribution of the passwords, which can not be estimated for corpora of such a small size: Every sentence from the Webis-Mnemonics-17 corpus leads to a different password, which makes it impossible to infer the probability distribution from the data. Because of this, most previous work on strength estimates for mnemonic password strengths [26], [41] were restricted to reporting the percentage of cracked passwords when using cracking software, with the usual drawbacks [3]: results are hard to compare, hard to repeat, and rely on the specific cracking method. For example, because they use different cracking methods, Yan et al. and Kuo et al. come to different conclusions regarding the password strengths. In order to resolve these problems, we use a web crawl to collect a huge amount of sentences that are sufficiently similar to human-chosen mnemonics like those in the Webis-Mnemonics-17 corpus. We then use these web sentences in place of the mnemonics (see below).

B. *The Webis-Sentences-17 Corpus*

To analyze natural language sentences at huge scale we specifically designed the new Webis-Sentences-17 corpus,[3] which is based on the ClueWeb12 web page crawl (version 1.1) [36]. The ClueWeb12 is a 27.3 TB collection of 733 million English web pages crawled in 2012. It covers authors from a wide range of age, education, and English-speaking countries. The ClueWeb12 is distributed as HTML, making an automatic sentence extraction method necessary.

Since we are interested in content sentences only, we design an automatic extraction algorithm and test it by comparing it to human extraction capabilities. For this purpose, 924 sentences were manually extracted by copy-and-pasting all fitting sentences from 100 random ClueWeb12 web pages. Out of the passwords from automatic sentence extraction, 81% match those from the human extraction.[4] As Section III-D shows, this quality is sufficient for the purposes of this paper.

We use an own open source extraction method with optimized parameters:[5] The method renders the web page text[6] and removes non-English paragraphs [15], paragraphs with less than 400 characters, sentences with less than 50% letter-only tokens,[7] and sentences without an English function word. We found that some domains use a small set of sentences frequently and filtered such sentences by removing re-occurrences within 1 000 extracted sentences. Further excluding spam pages [8] could not improve the method. We also tried the standard Boilerpipe ArticleSentenceExtractor [24], but found that it performed worse in our tests.

The final Webis-Sentences-17 corpus contains 3.4 billion sentences. From these, we generate on average 1.4 billion passwords of length 8 to 20 per generation rule. We chose this range based on length limits in popular web pages [18]. Table IV gives a few examples from this corpus.

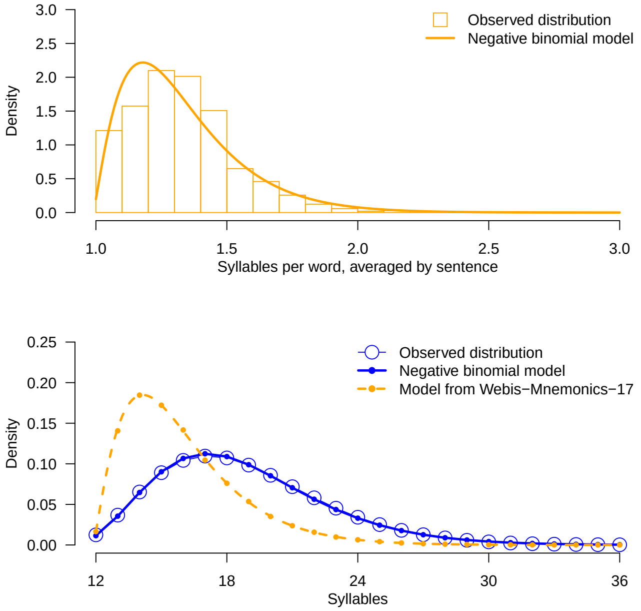


Figure 3. Distribution and fitted model of syllable counts per word for the Webis-Mnemonics-17 corpus (top) and per sentence of length 12 in the Webis-Sentences-17 corpus (bottom)

C. *The Webis-Simple-Sentences-17 Corpus*

As the Webis-Sentences-17 corpus intuitively contains more complex sentences than can be expected for mnemonics, we created the Webis-Simple-Sentences-17 sub-corpus with a sentence complexity like in the Webis-Mnemonics-17 corpus.[3] For measuring sentence complexity, we use the standard Flesch reading ease test [11] (higher F means more readable):

$$\text{F} = 206.835 - 84.6 \cdot \frac{\#\text{syllables}}{\#\text{words}} - 1.015 \cdot \frac{\#\text{words}}{\#\text{sentences}} . \quad (1)$$

For the Webis-Simple-Sentences-17 corpus, we sample sentences from the Webis-Sentences-17 corpus such that, for each sentence length, the syllable distribution of the sampled sentences matches the syllable distribution in the Webis-Mnemonics-17 corpus. Since this requires to compare only Flesch values for single sentences of the same length, Equation 1 essentially reduces to the number of syllables, where less syllables correspond to simpler sentences. For the sampling probabilities, we fit negative binomial models—which are usual for syllable counts of English sentences [17]—to the observed syllable counts of the Webis-Mnemonics-17 and Webis-Sentences-17 corpora.[8] Figure 3 shows these models. When sampling sentences, the appropriate sampling probability for each sentence length and syllable count follows directly from these models. Also, the Figure shows that the web sentences are indeed significantly more complex than human-chosen mnemonics. Hence, the Webis-Simple-Sentences-17 corpus is more similar to mnemonics than the Webis-Sentences-17 corpus. The final corpus consists of 0.5 billion sentences. From these sentences, we generate on average 0.23 billion passwords of length 8 to 20 per generation rule. Table V gives a few examples from this corpus.

[3] www.uni-weimar.de/en/media/chairs/webis/corpora/webis-sentences-17

[4] Tested for the lowercase letter word initials password generation rule

[5] Source: github.com/webis-de/aitools4-aq-web-page-content-extraction

[6] Rendering by Jericho HTML: jericho.htmlparser.net v. 3.2

[7] Tokenization by ICU4J: site.icu-project.org/home v. 53.1

[8] As the negative binomial distribution is a discrete distribution, the model for the Webis-Mnemonics-17 syllable counts is first fit to a transformed value of (syllables-per-word $-$ 1) $\cdot$ 100 and then transformed inversely

Table II. CHARACTER-WISE CROSS ENTROPY ESTIMATES FOR PASSWORDS FROM THE WEBIS-MNEMONICS-17 CORPUS OF LENGTH 12. MODEL CORPORA: WEBIS-MNEMONICS-17 (WM), WEBIS-SENTENCES-17 (WS), WEBIS-SIMPLE-SENTENCES-17 (WSS).

Character set	Model corpus	Model order					
		0	1	2	3	4	5
ASCII	WS	4.95	4.64	4.58	**4.56**	4.62	4.75
	WSS	4.94	4.63	4.56	**4.55**	4.62	4.76
	WM	4.59	**4.51**	4.54	4.55	4.55	4.55
Lowercase letters	WS	4.17	4.11	4.08	**4.06**	4.07	4.14
	WSS	4.16	4.09	4.06	**4.04**	4.06	4.14
	WM	4.14	**4.10**	4.18	4.20	4.20	4.20

D. Web Sentence and Mnemonic Similarity

We will now argue why password strength estimates will be approximately the same for passwords from mnemonics and from web sentences. (a) Strength estimates for password distributions depend on the distribution of password probabilities and not on the literal passwords (cf. Section IV-C). (b) Password probabilities can be estimated well from a sample using language models, as successfully exploited for password cracking [9], [27], [30], [34]. (c) Language models estimate password probabilities using only the conditional probabilities of the characters given their preceding characters [7].
Hence, given passwords from two different password sources in which these conditional character probabilities follow approximately the same distributions, the password probabilities of these two sources will also follow approximately the same distribution (from b+c), and the strength estimates will therefore be approximately the same for both password sources (from a). It is important to note that the above reasoning does *not* require that both sources contain the same passwords.
Moreover, algorithmic successes suggest that these conditional character probabilities from mnemonics and web sentences follow approximately the same distributions: (1) Automatic language identification based on related conditional character probabilities works robustly on short texts from various sources [15]; (2) Human-chosen password phrases—a similar setting to that of mnemonics—can be cracked using language models from a few million web sentences [34].

In order to provide further evidence for the similarity, we show that, while complete passwords from mnemonics and web sentences are likely different, they are composed from a very similar set of common substrings. This suggests that the difference between mnemonics and web sentences is more of a topical than a linguistic kind, and has therefore not much impact on the strength estimates. To show that both kind of sentences are composed from a very similar set of common substrings, we compare the cross-entropy—a standard similarity measure of distributions—of different sentence corpora to the Webis-Mnemonics-17 corpus using language models with specific model orders (cf. Section IV-A for details). A model of order o only considers substrings up to $o+1$ characters. As Table II shows, the cross entropy from the web sentences corpora to the mnemonic corpus gets about as low as the cross entropy between different subsets of the mnemonic corpus. Therefore, the substrings up to length 4 or 5 in passwords from the web sentences corpora are very similar to those in human-chosen mnemonics.

Table III. EXAMPLE SENTENCES DRAWN RANDOMLY FROM THE WEBIS-MNEMONICS-17 CORPUS FOR EACH OF THE TOPIC SUGGESTIONS FROM THE USER STUDY (CF. SECTION III-C).

No suggestion

- What was the color of your car when you were twenty years old?
- The order of my favorite colors followed by my cousin's pets is the password that I use.
- The five green ships docked at the west yellow arrow pointing south.
- i have an upside down kayak that floats on air without wings
- Three birds are sitting on a hibiscus tree driving their cars fast
- my very eager mother just served us pickles, never eat shredded wheat
- My parents are driving here from Michigan to visit for a week.

Your sentence should be related to mail

- beautiful mails require a touch of golden heart and brave minds that also pray
- Savings under the floorboards are safer than inside a big bank vault.
- Boy, you must be Fedex because you look like a hot mail.
- Is it all junk today, or is there anything worthwhile for a change?
- I like talking with my friends about current events and things that will happen in the near time coming.
- i can remember very well what i try to keep as a secret
- I pick up the mail at noon from the mailbox in the lobby of the building
- I want to become a successful teacher as well as a lovable mother

Your sentence should be related to shopping

- While shopping i usually purchase meaningless items that i wrap up in shinny paper.
- when I don't have money I want it, if I have money I want more.
- the cat liked to shop for cookies and bananas at the store in france
- I go shopping in the spring only when it's raining in Paris.
- There is a little girl shopping for a blue dress for her sister.
- When I go shopping, I always buy at least two bunches of bananas.
- Warehouse savings can multiply with money deposited into my account every day.
- My three sons bought the faith of the king with a robe.

Your sentence should be related to money

- Cash is king of the hill and worth every penny and cent.
- The crisp green bill did not leave the frugal boy's pocket until the day he died.
- The community i was born and raised in until I turned legal age.
- I like to bathe in a vat of crisp tens and twenties.
- Just like my inventory in Dragon Age Origins I am hella loaded
- My wife and I are often worried we will have enough money.
- She will get a new apron on her 3rd birthday next year.
- i have huge amount of money and have kept all of my money in savings banks

Your sentence should be related to talking with friends

- How do you know that carrots are good for the eye sight?
- I told my friend a secret and told her not to tell anyone
- Hey tell me what friends usually talk when they meet or call?
- it is important to wash your hands through out the day to keep proper hygine.
- I like chat with friends because they are so funny and I am happy I have them.
- My dear friend how are you and do you know the secret about our teacher mallika
- Talking to friends can be fun and sometimes we learn new things.
- My friends make me feel confident about myself and my work skills

Table IV. Example sentences drawn randomly from the Webis-Sentences-17 corpus (cf. Section III-B).

- There are also other retail outparcel developments on the other side of the interchange as well as some industrial development in the immediate area, so the center promises to have a strong regional draw.
- The ADA recommends that the costs associated with postexposure prophylaxis and exposure sequelae be a benefit of Workers' Compensation insurance coverage.
- Your agents will come away with the knowledge of how service level and quality go hand-in-hand and how that affects the entire contact center.
- This distance, the 'local loop', helps determine which of the providers in Manhattan will be the best options to provide service to your location.
- The arena act was the product of gate keeping & was only ever important from a commercial standpoint.
- And when it comes to painting, throw out your color charts because rural Pennsylvanians use an array of hues not found in nature or in any hardware stores looking to remain on the right side of the Better Business Bureau.
- Nominations are called for Vice-president and two Director positions on the Board of Directors of ALIA, as incorporated under Corporations Law.
- The lack of initiative in this case seemed puzzling due to nearly all Americans' faith at the time in the strength and reliability of the constitutional machinery of due process.
- It will be better for you if you renounce meat & masalas.

Table V. Example sentences drawn randomly from the Webis-Simple-Sentences-17 corpus (cf. Section III-C).

- Please do not ask to return an item after 7 days of when you received the item.
- This guide has a lot of nuggets, and I could only stop when I was finished with it.
- She acted as a student leader during her primary school, high school, college and graduate studies.
- As mentioned, some gyms also have a daycare program so that you can drop the kids off there while you work out.
- How much you lose depends on the compression level, but it happens with all saves.
- So far it looks to top the current king of the hill (Radeon 4870X2) in most but not all benchmarks.
- Your dog will be well behaved and all your friends will want to know how you did it.
- And if that is what we want, then talking about "attraction" and "bonding" is a good place to begin.
- The ramps vary in size and height and you will want to look around to find the best one for your ATV needs.
- That's blatant right there, you should have seen how wroth Bela Karolyi was about that.
- Some of the more commonly known herbs to avoid during pregnancy include:
- Additional cost and energy savings are realized by reducing or eliminating the need for hot water, detergent, labor costs, and capital costs.
- You can structure it and then restructure it as per your needs.

IV. Password Strength Estimation

Password strength is measured on the password distribution, which is unknown for mnemonic passwords but which can be estimated from huge password samples using language models. Language models are detailed in general in Section IV-A and optimized for mnemonic passwords in Section IV-B. After this, Section IV-C details the common strength measures we use in our analysis.

For a formal discussion, this section uses the following notations. X is a random variable distributed over a set of n passwords $\{x_1, \ldots, x_n\}$ according to the password distribution $\mathcal{X}$. We use $p_i = \Pr[X = x_i]$ to denote the probability that a password X drawn from $\mathcal{X}$ is equal to x_i. We enumerate passwords in descending order of their associated probability in $\mathcal{X}$, that means $p_1 \geq \ldots \geq p_n$. Furthermore, $x_i^1 \cdots x_i^{\ell_i}$ denotes the ℓ_i characters of password x_i and X^j denotes the random variable of the j-th character of a password. Finally, L denotes a random variable distributed according to the password lengths in $\mathcal{X}$.

A. Language Models

Even password corpora several orders of magnitude larger than the Webis-Sentences-17 corpus would not suffice to calculate reliable maximum-likelihood estimates for the probabilities of very rare passwords. The maximum-likelihood estimate of a password probability is the number of its occurrences divided by the size of the entire password sample. However, even in the Webis-Sentences-17 corpus, the often-used Good-Turing method[9] [12] estimates that about 75% of the probability mass of the unknown distribution corresponds to passwords that occur not a single time in the corpus. Therefore, using the maximum-likelihood estimate for each password is unsuitable.

The most widespread language models for passwords, often referred to as Markov chains or n-gram models, employ the chain-rule of probability to describe a password probability by its length and character probabilities.[10] Let the probability of password x_i be

$$p_i = \Pr[L = \ell_i] \cdot \prod_{j=1}^{\ell_i} p_{i,j}\ , \qquad \text{where}$$

$$p_{i,j} = \Pr\Big[X^j = x_i^j \,\Big|\, X^1 \cdots X^{j-1} = x^1 \cdots x_i^{j-1}, L = \ell_i\Big]\ . \qquad (2)$$

Instead of the exact probabilities in Equation 2, language models approximate the character probabilities by conditioning on only the o preceding characters [7], and thus require much less passwords. Therefore, they reduce the model complexity by assuming that

$$x_i^{j-o} \cdots x_i^j = x_k^{t-o} \cdots x_k^t \quad \rightarrow \quad p_{i,j} = p_{k,t}\ , \qquad (3)$$

which leads to robust models used successfully in various natural language tasks [7]. Applying Equation 3 to Equation 2,

$$p_{i,j} \approx \Pr\Big[X^j = x_i^j \,\Big|\, X^{j-o} \cdots X^{j-1} = x_i^{j-o} \cdots x_i^{j-1}, L = \ell_i\Big]\ ,$$

where a special start-of-password symbol is used to cope with characters preceding x_i^1:

$$\Pr\left[X^j = \text{start-of-password symbol}\right] = \begin{cases} 1 & \text{if } j \leq 0 \\ 0 & \text{if } j > 0 \end{cases}$$

[9] The estimate is calculated as the number of passwords occurring only once divided by the number of different passwords in the corpus

[10] An alternative method introduces an end-of-password symbol that is treated like a normal character by the language model [7]. However, the effect of this choice is usually negligible [27].

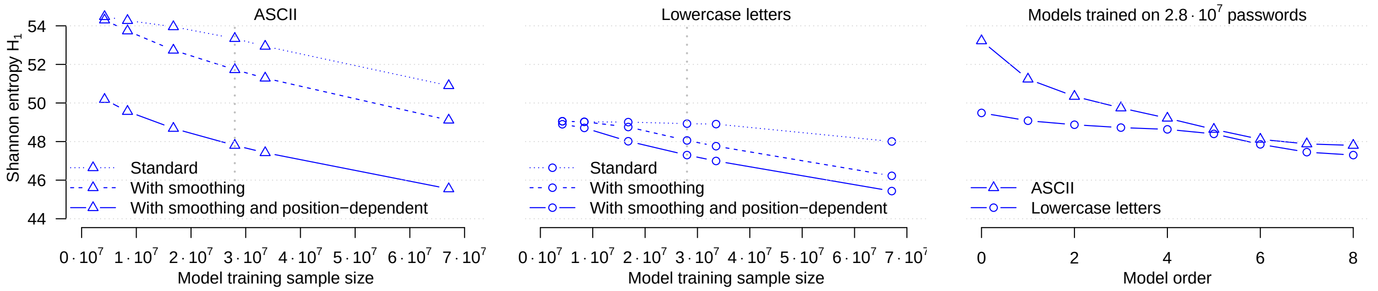


Figure 4. Effect of the sample size and model order in model training on estimated cross entropy for passwords of length 12 using the ASCII (triangles) and lowercase letters (circles) character sets. The left and center plots show the effect of the sample size for different model settings and optimal order. The right plot shows the effect of the model order for the selected sample size.

B. *Empirical Language Model Optimization*

Language models have several parameter, which are commonly optimized for a given task using the cross entropy on an independent password sample [7]. The cross entropy is

$$H_1(\mathcal{X}, \mathcal{X}') = -\sum_{i=1}^{n} p_i \cdot \log p'_i \, ,$$

where p_i and p'_i are the probabilities of x_i under $\mathcal{X}$ and $\mathcal{X}'$ respectively. In our case, $\mathcal{X}$ is the correct password distribution (approximated by the independent password sample) and $\mathcal{X}'$ is the distribution as estimated by the language model. Note that, when the language model is perfect, that means $\mathcal{X} = \mathcal{X}'$, the cross entropy is minimal and equal to $H_1(\mathcal{X})$. Conversely, because a lower cross entropy corresponds to a better language model, it is safe to optimize language models for cross entropy.

Model Order. The model order o governs the strength of the assumption in Equation 3. For example, $o = \ell_i$ gives the unreliable maximum-likelihood estimate of password probabilities. On the other hand, $o = 0$ assumes that character probabilities are independent of preceding characters, which leads to robust but heavily biased estimates.[11] In general, the best value for o depends on the amount of passwords in the sample.

Smoothing. Smoothing methods use prior-assumptions to improve the unreliable probability estimates for rarely occurring sequences [7]. We use the interpolated Witten-Bell smoothing method [7], [40] for our experiments, which is suggested for character-based models [35]. This method blends unreliable higher-order estimates with more-reliable lower-order ones.

Position-dependency. For the special case of password distributions, we propose to use position-dependent language model. Position-dependent models account for the different character distributions at the start, middle, and end of sentences.[12] This is done by estimating the conditional character probabilities for each character position in a password separately. Formally, this corresponds to adding the requirement $j = t$ to Equation 3. To the best of our knowledge, we are the first to apply position-dependent models to passwords. As the results below show, position-dependent models are superior for estimating mnemonic password distributions.

Since the different sentence corpora and generation rules lead to password corpora of different sizes, we optimize language models for two scenarios: using all available passwords (for the best strength estimates) and using only a sample of a specific size that is reached by most password corpora (for a fair strength comparison). In order to ensure a safe optimization without overfitting to the data, we create the language models[13] from passwords from 19 of the 20 ClueWeb12 parts and evaluate them on the last part that contains mostly web pages from different domains. Therefore, a smaller entropy estimate directly corresponds to a better model. Figure 4 (left, center) shows how the entropy estimates decrease with increasing sample size. In to ensure a fair comparison between generation rules for which we have different sample sizes, we use only $2.8 \cdot 10^7$ passwords per password length and rule when comparing rules (Sections V-A,V-B). We chose this size so that it is reached for most generation rules.

Furthermore, Figure 4 shows that smoothed position-dependent models of the highest order perform best, and we therefore use these models in our experiments in Section V. As the Figure demonstrates, position-dependent models are especially advantageous for ASCII passwords, probably due to the included punctuation that occurs mostly as last characters.

C. *Password Distribution Strength Measures*

A password-generation rule is stronger when the passwords it generates are more difficult to guess. However, this difficulty depends largely on the knowledge of the guesser. We employ the common Kerckhoffs' principle [22]: since we cannot estimate the knowledge of the adversary, we use the worst-case scenario that she knows the full distribution. Even if the adversary would not know the generation rule, related results suggest that users employ only very few different rules [42]. The adversary tries to guess by choosing one password, verifying it, and repeating to choose and verify until the correct one is found. Since she knows the full password distribution, she guesses passwords ordered by their probability.

We follow related work on password security and distinguish two scenarios: online, where adversaries have a small number of guesses until the authentication system blocks them, and offline, where they are limited only by their time [3].

[11] This and similar choices between too complex ($o = \ell_i$) and too simple ($o = 0$) are known as bias-variance trade-off in machine learning [2].

[12] For example, in web sentences of length 8, a total of 21% of the first words start with "t", but only 8% of the last words do so, too.

[13] We use SRILM v. 1.7.1 (www.speech.sri.com/projects/srilm/) to generate language models and a custom implementation based on KenLM (kheafield.com/code/kenlm/) to get probabilities.

For all the measures detailed below, a higher value corresponds to a stronger password distribution.

Min-entropy. The min-entropy models the very extreme case where the adversary guesses only a single password [3]. The min-entropy H_∞ is a widespread measure to assess distributions, not only of passwords. It is defined by

$$H_\infty(\mathcal{X}) = -\log p_1$$

Failure Probability. The failure probability is a measure for the online scenario. The failure probability λ_β reflects the average probability of not guessing a password with β guesses [5].

$$\lambda_\beta(\mathcal{X}) = 1 - \sum_{i=1}^{\beta} p_i$$

We report on $\beta = 10$ and $\beta = 100$ (like [3], [6]).

Work-factor. The α-work-factor is a measure for the offline scenario. It models the case where adversaries guess until they have guessed a fraction α of passwords. The α-work-factor μ_α gives the expected number of guesses [32].

$$\mu_\alpha(\mathcal{X}) = \min\left\{\beta \,|\, 1 - \lambda_\beta(\mathcal{X}) \geq \alpha\right\}$$

We report on $\alpha = 0.5$ (like [3], [5]).

Shannon Entropy. The Shannon entropy H_1 measures the bits needed to encode events from a distribution. Unlike the other strength measures, H_1 considers the full distribution. For a uniform distribution, $H_1 = H_\infty$, and $H_1 > H_\infty$ otherwise.

$$H_1(\mathcal{X}) = -\sum_{i=1}^{n} p_i \cdot \log p_i \tag{4}$$

Shannon entropy is usually approximated by the cross entropy on a held-out password sample (cf. Section IV-B).

The computational cost of the work-factor $\mu_{0.5}$ makes it infeasible already for passwords of length 9 or 10, but we find that it strongly correlates with the Shannon entropy H_1 in our case (Figure 5, Pearson's r = 0.71). H_1 has been criticized as a strength measure for password distributions as it does not clearly model the offline scenario [5], [32]. However, due the observed strong correlation, we see it as a meaningful strength measure in the case of mnemonic passwords.

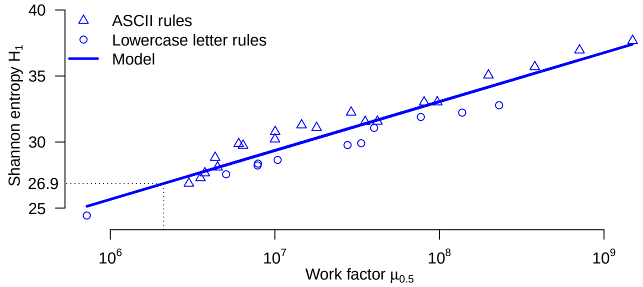


Figure 5. Scatter plot of strength estimates for different password generation rules and sentence corpora by work-factor (logarithmic scale) and Shannon entropy for passwords of length 8. All language models are trained on $2.8 \cdot 10^7$ passwords. The dotted line shows an estimated $\mu_{0.5}$ for real-world passwords [3] and the corresponding H_1 according to the model.

V. Experiments

This section analyzes the strength of mnemonic password distributions. It addresses the following research questions:

- Which of the password generation rules generates the strongest password distribution? (Section V-A)
- What effect does sentence complexity have on password distribution strength? (Section V-B)
- Does password distribution strength increase linearly with password length? (Section V-C)
- Security-wise, how far are mnemonic passwords from uniformly sampled character strings? (Section V-D).
- How strong are mnemonic passwords compared to other password approaches? (Section V-E, V-F)

A. *Estimates by Generation Rules*

This experiment compares the strength of password distributions from 18 generation rules in terms of common strength measures (Section IV-C). A password generation rule is an algorithm which a human can apply to transform a short text into a password. For this evaluation, we selected rules that vary by the employed character set, replacement rules, and the chosen words from the sentence and characters from the words. The selected rules follow the standard rule of word initials (no replacement, every word, first character) [14], [26], [31], [41] with some variations to test the effect of such variations on the reached security level. If not said otherwise, other experiments use this standard rule. Our implementation of the generation rules is available open source.[14]

Character set. The generated passwords consist of either lowercase letters (26 characters) or 7-bit visible ASCII characters (94 characters). Each sentence is processed by a Unicode compatibility and canonical decomposition and stripped of diacritical marks. For lowercase passwords, all letters are converted to lowercase. Then, remaining unfitting characters are removed. Punctuation is treated as an own "word" for ASCII passwords.[15] While a larger character set theoretically leads to stronger passwords, especially users of on-screen keyboards are tempted to use only lowercase letters as switching to uppercase or special characters is an extra effort.

Replacement. Sometimes the mnemonic password advice includes to replace words by similar-sounding characters. To analyze this advice, we include deterministically replacing word prefixes (like "towards" → "2wards") as a variant.[16]

Word. We use either every word or every second word in the sentence for generating the password. Theoretically, omitting words increases the difficulty of guessing the next character.

Character position. Besides concatenating the first characters, we analyze using the last or both characters as variants. For one-character words, all three variants use this character once.

[14] https://github.com/webis-de/password-generation-rules

[15] We use the ICU4J BreakIterator: site.icu-project.org v. 53.1

[16] The employed replacements are based on a list of "pronunciation rules" with the two additional rules of "to" → "2" and "for" → "4": blog.codinghorror.com/ascii-pronunciation-rules-for-programmers

Table VI. MIN-ENTROPY (H_∞), FAILURE PROBABILITY (λ_β), AND SHANNON ENTROPY (H_1) FOR DIFFERENT PASSWORD-GENERATION RULES, SORTED BY H_1. THE VALUES ARE FOR PASSWORDS OF LENGTH 12 FROM THE WEBIS-SENTENCES-17 (WS) AND WEBIS-SIMPLE-SENTENCES-17 (WSS) CORPORA WITH MODELS FROM AT MOST $2.8 \cdot 10^7$ PASSWORDS. VALUES FROM FEWER THAN $2.8 \cdot 10^7$ PASSWORDS ARE SHOWN GRAY.

Password generation rule				H_∞		λ_{10}		λ_{100}		H_1	
Character set	Replacement	Word	Char. pos.	WS	WSS	WS	WSS	WS	WSS	WS	WSS
ASCII	✓	every 2nd	1st	13.8	13.2	0.99958	0.99940	0.99827	0.99753	**56.7**	55.8
ASCII	-	every 2nd	1st	13.8	13.2	0.99959	0.99940	0.99827	0.99752	53.6	52.9
ASCII	✓	every	1st	13.8	12.5	0.99949	0.99925	0.99760	0.99689	49.9	**48.0**
ASCII	✓	every 2nd	last	13.8	13.3	**0.99960**	0.99939	0.99825	0.99745	49.8	50.0
Lowercase letters	-	every 2nd	1st	13.1	12.8	0.99956	0.99938	**0.99840**	0.99739	48.5	48.1
ASCII	-	every	1st	13.8	12.4	0.99948	0.99925	0.99759	0.99688	47.8	46.2
ASCII	-	every 2nd	last	13.9	13.3	**0.99960**	0.99939	0.99824	0.99744	47.6	47.8
Lowercase letters	-	every	1st	11.4	12.8	0.99912	0.99928	0.99738	0.99739	47.3	45.7
ASCII	✓	every 2nd	1st+last	12.4	12.8	0.99940	0.99925	0.99775	0.99724	46.5	44.9
Lowercase letters	-	every 2nd	last	13.1	12.8	0.99955	0.99938	0.99833	0.99735	44.6	44.7
ASCII	-	every 2nd	1st+last	13.1	12.5	0.99948	0.99929	0.99767	0.99725	44.6	43.0
ASCII	✓	every	last	**14.0**	12.4	0.99951	0.99925	0.99759	0.99690	43.4	42.6
Lowercase letters	-	every	last	11.4	12.8	0.99912	0.99928	0.99734	0.99738	42.7	41.8
Lowercase letters	-	every 2nd	1st+last	12.0	**13.3**	0.99933	**0.99941**	0.99803	**0.99785**	42.6	41.2
ASCII	-	every	last	**14.0**	12.5	0.99950	0.99925	0.99757	0.99689	42.0	41.3
Lowercase letters	-	every	1st+last	10.3	9.6	0.99708	0.99650	0.99225	0.99098	36.8	35.3
ASCII	✓	every	1st+last	10.8	9.7	0.99772	0.99715	0.99108	0.98826	35.8	36.1
ASCII	-	every	1st+last	8.5	11.5	0.99400	0.99775	0.98634	0.98936	34.8	35.2

Table VI shows the estimated strength measures for passwords of length 12 from the 18 employed generation rules. The discussion below focuses on the results for the Webis-Sentences-17 corpus. While mnemonic password distributions in the real world contain passwords from different lengths, we restrict the analysis here to passwords from one length in order to make the comparison easier to understand, as it removes the influence of the length distribution. Especially generation rules that use two characters per word have very different length distributions. Strength estimates based on a natural distribution of password lengths are discussed from Section V-C onwards. For a fair comparison, we use the same number of passwords for all estimates, and mark estimates for rules for which our data has less passwords in gray. These estimates in gray are less reliable and biased to higher values for H_1.

For the online scenario measures min-entropy H_∞ and failure probability λ_β, comparable strengths are achieved by all generation rules but those that use multiple characters and every word, which are considerably weaker. For H_∞, a further factor is the character set where ASCII has about 1 bit advantage. For λ_{100}, generation rules that use every second word are stronger than other rules.

For the offline scenario measure H_1, passwords from ASCII achieve a similar strength to passwords with only lowercase letters when every word is used, but better strength when every second word is used. In total, using every second word and only the first character with the ASCII character set leads to the strongest of the tested password distributions. Also, word prefix replacements can increase the entropy by 2–3 bit. Moreover, using the first character of a word is preferable.

The strongest distribution is arguably using the ASCII character set, every second word, and only the first characters, which achieves best or nearly-best values for all measures. Word prefix replacement considerably increase the strength for H_1, but not for the online scenario. However, both using only every second word and word prefix replacements come with additional memorization and processing costs, a discussion of which lies outside the scope of this publication.

B. Estimates by Sentence Complexity

Table VI also shows that strength estimates for the Webis-Simple-Sentences-17 corpus are most times a bit weaker, but still very similar, to those from the Webis-Sentences-17 corpus for all distributions with sufficient training passwords. The maximum difference for one generation rule between the corpora are 1.6 bit for H_∞, 0.00026 for λ_{10}, 0.00071 for λ_{100}, and 1.9 bit for H_1. This corresponds to a large difference for H_∞ and a still noticeable difference for H_1, but smaller than one could expect.

Therefore, mnemonics with lower complexity do indeed lead to passwords that are easier to guess. This is likely due to the reduced vocabulary of the mnemonics, which is biased towards words with less syllables.

The effect of mnemonic complexity is especially strong for the min-entropy H_∞, which considers the most probable password only. A possible explanation for this is that the most probable password stems from simple sentences, even for the Webis-Sentences-17 corpus. Then, the probability of this password increases naturally when more complex sentences are filtered out.

On the other hand, the effect of mnemonic complexity is still noticeable for the Shannon entropy H_1, which considers the entire password distribution. Therefore, reducing the complexity skews the entire password distribution farther away from the uniform distribution. However, the effect is much weaker than for min-entropy. An estimate of the effect can be the maximum difference in Table VI between Webis-Sentences-17 and Webis-Simple-Sentences-17 for generation rules with sufficient training passwords, divided by the password size. This estimates the effect to 0.16 bit per character.

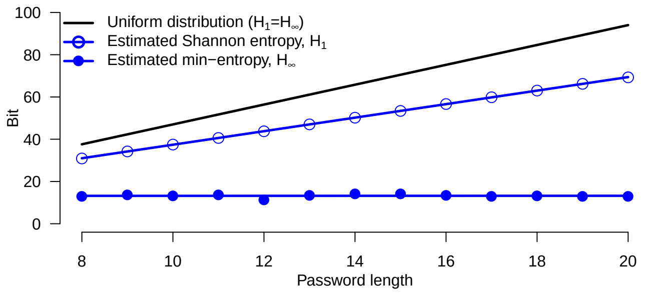


Figure 6. Shannon entropy and min-entropy estimates compared to the optimal uniform password distribution by password length. Passwords are from the Webis-Sentences-17 corpus using lowercase letters and the first character.

C. Estimates by Password Length

This section analyzes how the strength of password distributions increases with password length. The number of possible passwords increases exponentially with the password length, theoretically leading to stronger password distributions. Using the Webis-Sentences-17 corpus, we analyzed all rules to very similar results. As an example, Figure 6 shows the result for the standard generation rule using lowercase letters only.

Figure 6 shows that the resistance against offline attacks (H_1) increases as expected with password length, but that the resistance against online attacks (H_∞) stays rather constant.[17] We also found λ_{10} and λ_{100} to be rather constant.

The approximately constant resistance against online attacks shown in Figure 6 suggests that, for each length, there are a few sentences with a high probability irrespective the length. Only after these high-probability sentences, a spreading of the probability mass over the possible sentences occurs. This spreading is shown by the steady increase of the Shannon entropy. Unfortunately, the Webis-Mnemonics-17 corpus is far too small to reproduce this effect on human-chosen mnemonics. It thus remains unclear to which extent this effect also appears for human-chosen mnemonics. However, based on our analysis it is reasonable to assume that the resistance against online attacks of mnemonic passwords grows way less with password length than one would expect.

The linear increase of the Shannon entropy with password length leads to a simple model for estimating the entropy of password distributions with several lengths. In detail, one can rewrite Equation 4 (Shannon entropy) as

$$H_1(\mathcal{X}) = \sum_{\ell=\ell_{\min}}^{\ell_{\max}} \Pr[L=\ell] \cdot (H_1(\mathcal{X}_\ell) - \log \Pr[L=\ell]) , \quad (5)$$

where $H_1(\mathcal{X}_\ell)$ is the entropy estimate for length ℓ. Moreover, for the probability of a password-length, $\Pr[L = \ell]$, one can use the geometric model of lengths from the Mnemonic Survey corpus (Figure 2).[18] Due to the geometric model and only a linear increase of the entropy by length, Equation 5 converges as $\ell_{\max}$ increases. We report the converged values in the following. The remaining parameter is the minimum password length $\ell_{\min}$, which one can increase to increase the password distribution strength, as it is best practice for password-based authentication in general [10]. Using the mean from the fitted geometric distribution, the average password length is $\ell_{\min} + 1.4$ for passwords that take every word and $\ell_{\min} + 0.7$ for passwords that take every second word, while the mode is $\ell_{\min}$ in both cases. Note that this consideration makes the simplifying assumption that the parameter of the geometric distribution does not depend on $\ell_{\min}$.

Table VII. Estimated entropy by generation rule and minimum password length for passwords from the Webis-Sentences-17 corpus.

Char. set	L. letters	ASCII	L. letters	ASCII	ASCII
Replacement	-	-	-	-	✓
Word	every	every	every 2nd	every 2nd	every 2nd
Char. pos.	1st	1st	1st	1st	1st
$\ell_{\min}$	Shannon entropy H_1				
8	38.0	37.9	34.8	37.2	39.0
9	41.2	41.3	38.0	40.9	42.9
10	44.4	44.8	41.2	44.6	46.7
11	47.6	48.3	44.4	48.3	50.6
12	50.8	51.8	47.6	52.0	54.5
13	54.0	55.2	50.8	55.7	58.4
14	57.2	58.7	54.0	59.4	62.3
15	60.4	62.2	57.2	63.1	66.2
16	63.6	65.7	60.4	66.8	70.1
17	66.8	69.1	63.6	70.5	74.0
18	70.1	72.6	66.8	74.2	77.9
19	73.3	76.1	70.0	77.9	81.8
20	76.5	79.6	73.2	81.6	85.7
21	79.7	83.0	76.4	85.3	89.5
22	82.9	86.5	79.6	89.0	93.4
23	86.1	90.0	82.8	92.7	97.3
24	89.3	93.5	86.0	96.4	101.2
25	92.5	97.0	89.2	100.1	105.1
26	95.7	100.4	92.4	103.8	109.0
27	98.9	103.9	95.6	107.5	112.9
28	102.1	107.4	98.8	111.2	116.8
29	105.3	110.9	102.0	114.9	120.7
30	108.5	114.3	105.2	118.6	124.6

Table VII shows the minimum-length based entropy estimates for a selection of the strongest generation rules. This table aims at replacing for mnemonic passwords the "rules of thumb" that exist for the entropy of generic passwords (e.g., [6]). Unlike these rules of thumb, which were shown to not correlate with the password distribution strength against offline attacks [39], we have shown that our entropy estimates do correlate with it (cf. Figure 5). As the Table shows, when considering that rules using only every second word lead to shorter passwords on average, these rules lose much of their advantage, and are even weaker for lowercase letter passwords.

[17] H_∞ varies between 11.3 and 14.2 bit without a clear direction.

[18] When only every second word is used, the length distribution can be adjusted accordingly. However, an adjustment is not as straight-forward when two characters per word are used, due to one-character words. As this variant gave very weak distributions, we do not consider it here.

Table VIII. Character-wise entropy (H_1) and perplexity (Ppl.) estimates for passwords by model (cf. Section IV-B). Passwords are of length 12 from the Webis-Sentences-17 corpus using the first character of every word. The uniform model represents the optimal distribution over 26/94 characters.

	Lowercase letters		ASCII	
Model	H_1	Ppl.	H_1	Ppl.
Uniform	4.70	26.0	6.55	94.0
Order 0	4.15	17.8	5.09	34.1
Order 8	3.71	13.1	3.98	15.8
Order 8, position-dependent	3.65	12.6	3.70	13.0

D. Comparison to Uniform Distribution

The uniform password distribution is the strongest among all distributions with the same number of elements, but mnemonic password distributions fall short of it for three reasons: (1) some characters occur more frequently than others, (2) characters in a password are not independent of each other, and (3) the character distributions depend on the position in the password. Table VIII illustrates exploiting these 3 effects step by step for the standard mnemonic passwords. In addition to the Shannon entropy H_1, the table also shows the perplexity Ppl. $= \log(H_1)$ which gives the number of elements in a uniform distribution with the same entropy.

According to the results shown in Table VIII, both password distributions provide in an offline scenario about the same level of security as a uniform distribution over 12 to 13 characters. The biggest effect is in both cases that the characters are not uniformly distributed. On the other hand, exploiting the differences in the character distributions by position (using position-dependent models, Section IV-B) is especially valuable for ASCII passwords, where it can nearly reduce their strength to the strength of lowercase letter passwords. Like discussed in Section V-A, ASCII passwords are only stronger than lowercase letter passwords for specific generation rules.

E. Comparison to Dictionary Passwords

While the discussion in the following paragraphs exemplifies how our strength estimates can be used to compare the strength of different password generation methods, it does not incorporate other important factors for password usage like memorability, typing convenience, or susceptibility to typing errors. Unfortunately, we are not aware of any such comparison.

A second prominent suggestion for password generation is to pick several words uniformly at random from a large dictionary [29], [33]. We use the 7776 words Diceware dictionary [33] as an example.

A computation of the strength of such dictionary-based passwords is straight-forward. The min-entropy and Shannon entropy are both equal to

$$H_\infty = H_1 = n \cdot \log 7776 \,,$$

whereas the failure probability is calculated by

$$\lambda_\beta = 1 - \frac{\beta}{7776^n} \,.$$

The maximum length of a word in the dictionary is 6 characters. On average, the created passwords have a length of $n \cdot 4.24 + (n-1)$, where $(n-1)$ is then number of space characters as illustrated on the Diceware homepage.

The comparison with Diceware highlights the relative weakness of mnemonic passwords against online attacks: already the 2-word Diceware password distribution achieves a failure probability λ_{100} of 0.999998 and is thus considerably stronger in this scenario than every rule we considered and requires on average only 9.5 characters.

However, mnemonic passwords provide a better security in the offline scenario for the same password length. For example, 3 Diceware words (average password length of 14.7) achieve 38.8 bit of Shannon entropy, which is already reached by lowercase letter mnemonic passwords of minimum length 9 (Table VII, average length of 10.4).

F. Comparison to Real-world Password Distributions

This section compares the strength estimates for mnemonic passwords with estimates for real-world password distributions from the literature.

The currently largest-scale password strength analysis of real-world passwords is the analysis of 70 million anonymized Yahoo! passwords by Bonneau [3], which results in the following estimates: Min-entropy $H_\infty \approx 6.5$, failure probability with 10 guesses $\lambda_{10} \approx 0.98178$, and work factor $\mu_{0.5} \approx 2{,}111{,}739$.[19] While no estimate for the Shannon Entropy H_1 is provided, we can apply the log-linear relationship of $\mu_{0.5}$ and H_1 that we observed for mnemonic passwords, which suggests an H_1 of ~27 Bit (cf. Figure 5). Bonneau also compares the Yahoo! estimates to estimates from the password lists leaked from the RockYou and Battlefield Heroes websites. He finds that the corresponding two password distributions are even weaker against offline attacks. Also, only the Battlefield Heroes passwords are stronger against online attacks ($H_\infty \approx 7.7$, $\lambda_{10} \approx 0.98878$).

Comparing these estimates for real-world password distribution with our estimates for mnemonic password, we see that mnemonic passwords are considerably stronger attacks both online and offline attacks. For online attacks, our estimates for the standard lowercase letters word initial rule are for H_∞ between 11.4 and 12.8, and for λ_{10} between 0.99912 and 0.99928 (Table VI)—reducing the corresponding success probability $(1 - \lambda_{10})$ compared to the Battlefield Heroes passwords by 92–94%.[20] For offline attacks, we can extend Table VII for smaller ℓ_{min}, suggesting that a higher H_1 as for real-world passwords is reached by mnemonic passwords from the standard rule with a minimum length ℓ_{min} of 5. While the length distribution of the Yahoo! passwords is unknown, the current minimum password length for new Yahoo! accounts is 8 and thus considerably larger. Therefore, we can conclude that mnemonic passwords are stronger against both online and offline attacks compared to the passwords in use today.

[19] The paper provides normalized estimates, which all use a common scale. The estimates we report are un-normalized.

[20] When known phrases are allowed as mnemonics, related results suggest a similar strength against online attacks as the Battlefield Heroes passwords have [42], which highlights the importance of developing password-blacklists to keep users from choosing such easy-to-guess phrases.

VI. Conclusion And Outlook

This paper analyzes the strength of passwords generated according to the mnemonic password advice on a huge corpus of 3 billion human-written sentences. The detailed analysis of this paper considers sentence complexity and 18 different password generation rules. To this end, the paper shows that the necessary similarity of human-chosen mnemonics and web sentences exists. Furthermore, the paper contributes one of the currently biggest corpora of human-chosen mnemonics. Additionally, this paper is the first to apply position-dependent language models to passwords, which improve on regular language models for modeling mnemonic passwords.

Our analysis addressed several questions regarding the strength of mnemonic passwords.

Of the 18 tested password generation rules, the strongest password distribution is generated by using the ASCII character set, concatenating the first character of every second word, where common word prefix replacements are used to add more special characters to the passwords. Both using only every second word and word prefix replacements have only an effect in offline attack scenarios, where adversaries are not limited by a number of guesses but by the time they want to invest.

The sentence complexity of the used mnemonics has a major effect when the adversary can perform only a few guesses, and a relatively weak effect for offline attacks.

We showed that an attacker can use knowledge on the generation process of mnemonic passwords to drastically increase his success chances, reducing the strength of mnemonic passwords against offline attacks to that of passwords from a uniform distribution over only 12 to 13 characters.

We analyzed the effect of password length on the strength estimates, and found that—as one would expect—the strength of mnemonic passwords against offline attacks grows linearly with the password length. On the other hand, if the adversary can only perform a few guesses, our results suggest that longer passwords provide no further advantage.

Using statistical modeling, this paper provides with Table VII detailed estimates of the strength of mnemonic passwords against offline attacks for different minimum password lengths and password generation rules. This table aims to replace for mnemonic passwords the inaccurate "rule of thumb" for strength calculation that was used previously. With this table, we compare mnemonic passwords to a password generation approach that performs repeated uniform sampling from a dictionary and found that mnemonic passwords are weaker against online, but stronger against offline attacks.

The analysis of the password generation rules is limited to the strength of the corresponding password distributions and ignores that the different rules are associated with different costs for the human. For example, the best generation rule requires the human to memorize a twice as long sentence. Furthermore, already having a certain generation rule in mind will likely have an influence on mnemonic choice. For instance, if the human wants to use a rule that incorporates word prefix replacements, he may limit the considered mnemonics to such where he can actually perform a replacement operation. A more detailed study on memorability and mnemonic choice would be needed to improve this discussion.

Furthermore, this analysis is restricted to English mnemonics only. The question if our results also apply to mnemonics of other languages is open for further research.

An interesting avenue for further research could be to use search algorithms to find the best password generation rule for a given sentence distribution. The 18 rules that we analyzed cover only a very small part of the parameter space for such rules. Investigations in this direction would require to lower the computational cost of evaluating a rule, which is a problem in its own right. Moreover, an analysis of the costs of generation rule parameters like suggested above could also be integrated into the cost function of the search algorithm.

Acknowledgement

We would like to thank our shepherd Lujo Bauer, our colleagues Eik List and Martin Potthast, and the anonymous reviewers for their helpful comments and suggestions. We would also like to thank Norina Marie Grosch for preparing the Webis-Mnemonics-17 corpus for publication.

References

[1] M. J. Atallah, C. J. McDonough, V. Raskin, and S. Nirenburg, "Natural Language Processing for Information Assurance and Security: An Overview and Implementations," in *Proceedings of the 2000 Workshop on New Security Paradigms*, ser. NSPW '00. New York, NY, USA: ACM, 2000, pp. 51–65.

[2] C. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012, pp. 538–552.

[4] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The Past, Present, and Future of Password-based Authentication on the Web," *Communications of the ACM*, 2015, to appear.

[5] S. Boztas, *Entropies, Guessing and Cryptography*. Royal Melbourne Institute of Technology, Dept. of Mathematics Melbourne, 1999.

[6] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus, *NIST Special Publication 800-63-2: Electronic Authentication Guideline*. Gaithersburg, MD: National Institute of Standards and Technology, 2013.

[7] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, 1999.

[8] G. V. Cormack, M. D. Smucker, and C. L. Clarke, "Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets," *Computing Research Repository*, vol. abs/1004.5168, 2010.

[9] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password Strength: An Empirical Analysis," in *Proceedings of the 29th Conference on Information Communications*, ser. INFOCOM'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 983–991.

[10] D. C. Feldmeier and P. R. Karn, "UNIX Password Security - Ten Years Later," in *Proceedings on Advances in Cryptology*, ser. CRYPTO '89. New York, NY, USA: Springer-Verlag New York, Inc., 1989, pp. 44–63.

[11] R. Flesch, "A New Readability Yardstick," *Journal of applied psychology*, vol. 32, no. 3, p. 221, 1948.

[12] W. A. Gale, "Good-Turing Smoothing Without Tears," *Journal of Quantitative Linguistics*, vol. 2, 1995.

[13] M. Ghazvininejad and K. Knight, "How to Memorize a Random 60-Bit String," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–June 2015, pp. 1569–1575.

[14] Google Inc., "Secure your passwords," Last accessed May 2016, www.google.com/safetycenter/everyone/start/password/.

[15] T. Gottron and N. Lipka, "A Comparison of Language Identification Approaches on Short, Query-Style Texts," in *Advances in Information Retrieval*, ser. Lecture Notes in Computer Science, C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, and K. van Rijsbergen, Eds. Springer Berlin Heidelberg, 2010, vol. 5993, pp. 611–614.

[16] E. Grosse and M. Upadhyay, "Authentication at scale," *IEEE Security and Privacy*, vol. 11, pp. 15–22, 2013. [Online]. Available: http://www.computer.org/cms/Computer.org/ComputingNow/pdfs/AuthenticationAtScale.pdf

[17] P. Grzybek, "History and Methodology of Word Length Studies," in *Contributions to the Science of Text and Language: Word Length Studies and Related Issues*, P. Grzybek, Ed. Dordrecht, NL: Kluwer, 2007.

[18] T. Hornby, "Password Policy Hall of Shame," Last updated February 2014, www.defuse.ca/password-policy-hall-of-shame.htm.

[19] S. Jeyaraman and U. Topkara, "Have the cake and eat it too - infusing usability into text-password based authentication systems," in *Proceedings of the 21st Annual Computer Security Applications Conference*, ser. ACSAC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 473–482.

[20] K. A. Juang, S. Ranganayakulu, and J. S. Greenstein, "Using system-generated mnemonics to improve the usability and security of password authentication," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, no. 1, 2012, pp. 506–510.

[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*. IEEE Computer Society, 2012, pp. 523–537.

[22] A. Kerckhoffs, "La Cryptographie Militaire," *JSM*, vol. 9, pp. 161–191, February 1883.

[23] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing User Studies with Mechanical Turk," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 453–456.

[24] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate Detection Using Shallow Text Features," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 441–450.

[25] N. Kumar, "Password in Practice: An Usability Survey," *Journal of Global Research in Computer Science*, vol. 2, no. 5, pp. 107–112, 2011.

[26] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in *SOUPS*, ser. ACM International Conference Proceeding Series, L. F. Cranor, Ed., vol. 149. ACM, 2006, pp. 67–78.

[27] J. Ma, W. Yang, M. Luo, and N. Li, "A Study of Probabilistic Password Models," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, ser. SP'14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 689–704.

[28] 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 the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 173–186.

[29] R. Munroe, "Password Strength," August 2011, www.xkcd.com/936/.

[30] A. Narayanan and V. Shmatikov, "Fast Dictionary Attacks on Passwords Using Time-space Tradeoff," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, ser. CCS '05. New York, NY, USA: ACM, 2005, pp. 364–372.

[31] N. Perlroth, "How to Devise Passwords that Drive Hackers Away," November 2012, www.nytimes.com/2012/11/08/technology/personaltech/how-to-devise-passwords-that-drive-hackers-away.html.

[32] J. Pliam, "On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks," in *Progress in Cryptology —INDOCRYPT 2000*, ser. Lecture Notes in Computer Science, B. Roy and E. Okamoto, Eds. Springer Berlin Heidelberg, 2000, vol. 1977, pp. 67–79.

[33] A. G. Reinhold, "The Diceware Passphrase Home Page," May 2016, www.world.std.com/~reinhold/diceware.html.

[34] P. Sparell and M. Simovits, "Linguistic Cracking of Passphrases," 2016, (Talk) RSA Conference.

[35] A. Stolcke, D. Yuret, and N. Madnani, "SRILM-FAQ," Last accessed May 2016, www.speech.sri.com/projects/srilm/manpages/srilm-faq.7.html.

[36] The Lemur Project, "The ClueWeb12 Dataset," 2012, www.lemurproject.org/clueweb12.

[37] U. Topkara, M. J. Atallah, and M. Topkara, "Passwords Decay, Words Endure: Secure and Re-usable Multiple Password Mnemonics," in *Proceedings of the 2007 ACM Symposium on Applied Computing*, ser. SAC '07. New York, NY, USA: ACM, 2007, pp. 292–299.

[38] 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*, vol. 48, no. 13. SAGE Publications, 2004, pp. 1478–1482.

[39] M. Weir, S. Aggarwal, M. P. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *ACM Conference on Computer and Communications Security*, E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds. ACM, 2010, pp. 162–175.

[40] I. H. Witten and T. Bell, "The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, July 1991.

[41] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password Memorability and Security: Empirical Results," *IEEE Security and Privacy*, vol. 2, no. 5, pp. 25–31, Sep. 2004.

[42] W. Yang, N. Li, O. Chowdhury, A. Xiong, and R. W. Proctor, "An Empirical Study of Mnemonic Sentence-based Password Generation Strategies," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1216–1229.